IBM System/36

Procedures
Control Commands
Operation Control Language

System Reference

IBM System/36

System Reference

Program Numbers 5727-SS1
5727-SS6

File Number
S36-36

Order Number
SC21-9020-5

| **Sixth Edition (June 1987)**

| This major revision makes obsolete SC21-9020-4. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition. See "About This Manual" for a summary of changes.

Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

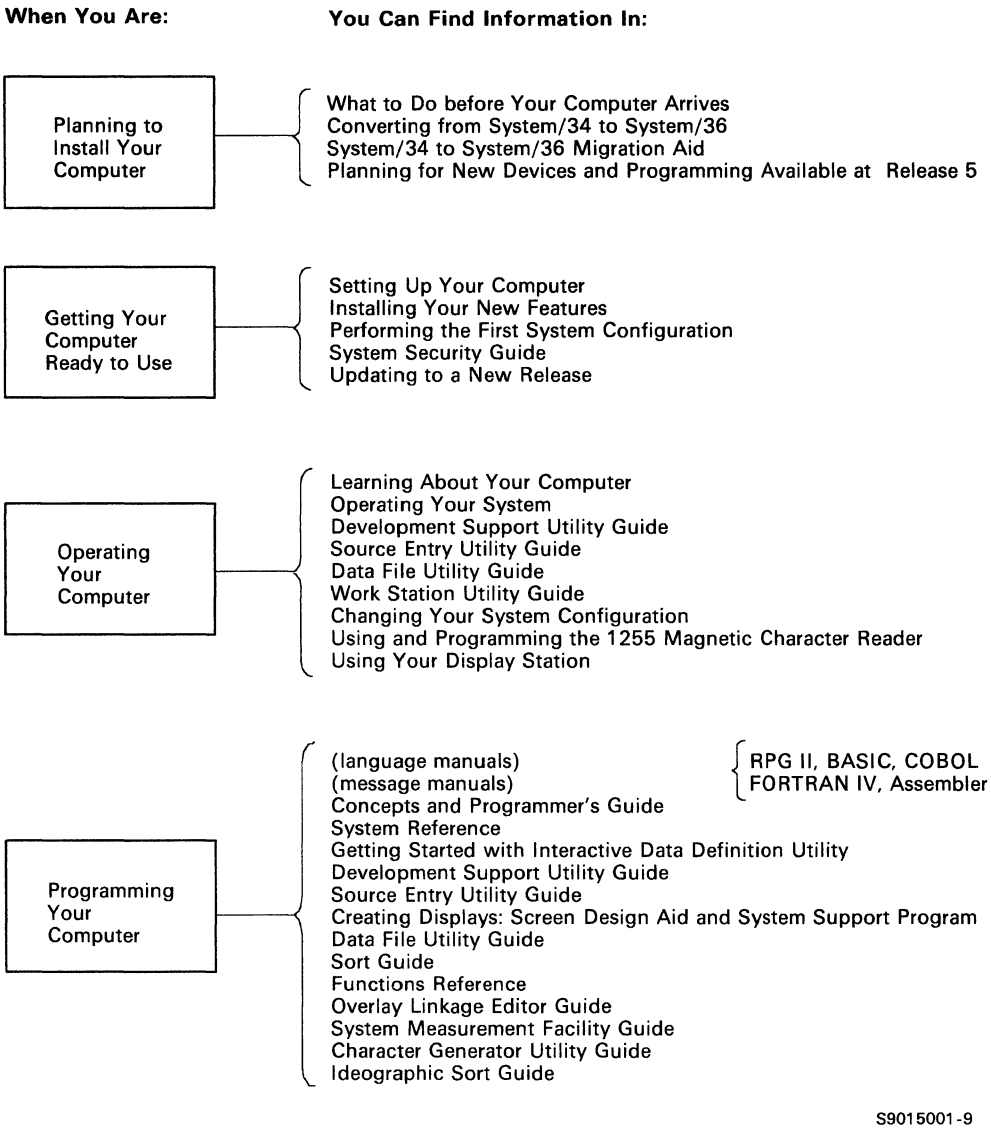
| This edition applies to Release 5, Modification Level 1, of IBM System/36 System Support Program Product (Program 5727-SS1 for the 5360 and 5362 System Units, and Program 5727-SS6 for the 5364 System Unit), and to all subsequent releases and modifications until otherwise indicated. Also, this publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

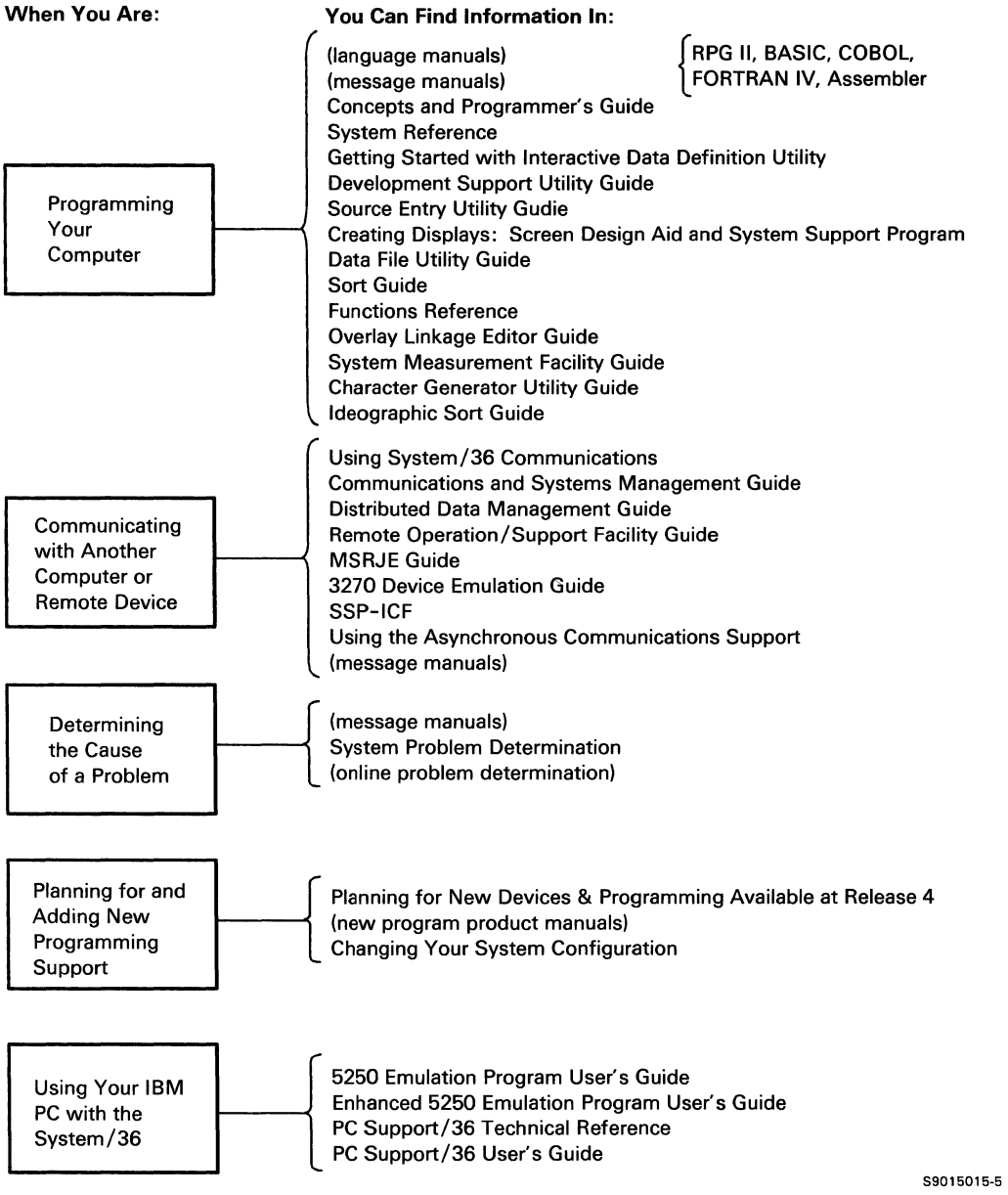
References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

The numbers at the bottom right of illustrations are publishing control numbers and are not part of the technical content of this manual.

Publications are not stocked at the address below. Requests for IBM publications should be made to your IBM representative or to your IBM-approved remarketer.

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department 245, Rochester, Minnesota, U.S.A. 55901. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.


When You Are:
You Can Find Information In:
Planning to Install Your Computer
What to Do before Your Computer Arrives
Converting from System/34 to System/36
System/34 to System/36 Migration Aid
Planning for New Devices and Programming Available at Release 5
Getting Your Computer Ready to Use
Setting Up Your Computer
Installing Your New Features
Performing the First System Configuration
System Security Guide
Updating to a New Release
Operating Your Computer
Learning About Your Computer
Operating Your System
Development Support Utility Guide
Source Entry Utility Guide
Data File Utility Guide
Work Station Utility Guide
Changing Your System Configuration
Using and Programming the 1255 Magnetic Character Reader
Using Your Display Station
Programming Your Computer
(language manuals)
RPG II, BASIC, COBOL
(message manuals)
FORTRAN IV, Assembler
Concepts and Programmer's Guide
System Reference
Getting Started with Interactive Data Definition Utility
Development Support Utility Guide
Source Entry Utility Guide
Creating Displays: Screen Design Aid and System Support Program
Data File Utility Guide
Sort Guide
Functions Reference
Overlay Linkage Editor Guide
System Measurement Facility Guide
Character Generator Utility Guide
Ideographic Sort Guide
S9015001-9


When You Are:
You Can Find Information In:
Programming Your Computer
(language manuals)
RPG II, BASIC, COBOL, FORTRAN IV, Assembler
(message manuals)
Concepts and Programmer's Guide
System Reference
Getting Started with Interactive Data Definition Utility
Development Support Utility Guide
Source Entry Utility Gudie
Creating Displays: Screen Design Aid and System Support Program
Data File Utility Guide
Sort Guide
Functions Reference
Overlay Linkage Editor Guide
System Measurement Facility Guide
Character Generator Utility Guide
Ideographic Sort Guide
Communicating with Another Computer or Remote Device
Using System/36 Communications
Communications and Systems Management Guide
Distributed Data Management Guide
Remote Operation/Support Facility Guide
MSRJE Guide
3270 Device Emulation Guide
SSP-ICF
Using the Asynchronous Communications Support
(message manuals)
Determining the Cause of a Problem
(message manuals)
System Problem Determination
(online problem determination)
Planning for and Adding New Programming Support
Planning for New Devices & Programming Available at Release 4
(new program product manuals)
Changing Your System Configuration
Using Your IBM PC with the System/36
5250 Emulation Program User's Guide
Enhanced 5250 Emulation Program User's Guide
PC Support/36 Technical Reference
PC Support/36 User's Guide
S9015015-5

Contents

About This Manual xiii
Who should use this manual xiii
How this manual is arranged xiii
What you should know xiv
If you need more information xiv
How this manual has changed xvi

Chapter 1. Introduction 1-1
Procedures 1-1
OCL Statements 1-2
SSP Utility Programs and Their Control Statements 1-2
Control Commands 1-3
Concept Information and Programming Considerations 1-3
Conventions Used for Describing Syntax Formats 1-3
Capitalized Expressions 1-3
Braces { } 1-4
Brackets [ ] 1-4
Parentheses ( ) 1-5
Underlining 1-5
Commas 1-6
Example Syntax Diagram 1-6
Directory to Using the System Support 1-7
Creating and Maintaining Disk Files 1-8
Creating and Maintaining Extended Character Files 1-10
Creating and Maintaining Libraries 1-11
Maintaining Folders and Folder Members 1-13
Processing Diskettes 1-15
Processing Tapes 1-17
Creating and Maintaining Display Formats, Menus, and Message Members 1-18
Creating and Maintaining Programs 1-19
Using the Office Products 1-21
Using the Personal Computer 1-23
Defining Data in Files 1-25
Running Programs and Procedures 1-26
Changing and Controlling Printers, Jobs, and Display Stations 1-29
Defining the System, Its Users, and Formatting the 9332 1-32
Communicating with Other Systems 1-33
Maintaining the System 1-35
Determining and Correcting Problems ... 1-36

Chapter 2. Making Your Own Procedures **2-1**
What a Procedure Is 2-1
What a Procedure Can Contain 2-1
Entering Procedures Into the System 2-2
Naming Procedures 2-2
Procedure Parameters 2-3
Procedure Parameter Defaults 2-5
Testing Entered Parameters 2-6
Parameter Coding Considerations 2-6
Continuing the Lines of a Procedure 2-7
Calling a Procedure from Another Procedure . 2-8
Procedure Attributes 2-10
Example Procedures 2-11
Example 1: Procedure SAMPLE 2-11
Example 2: Procedure LISTKEYS 2-13
Example 3: Procedure SCRNPRT 2-16
Procedure Performance Tips and Coding Techniques 2-22
Debugging Your Procedures 2-24

Chapter 3. Procedure Control Expressions ... **3-1**
Things You Can Do Using Procedure Control Expressions 3-2
Substituting Values and Information 3-2
Displaying Messages or Display Formats .. 3-5
Data File Information 3-5
Library and Folder Information 3-6
Diskette Information 3-6
Tape Information 3-7
Comparing and Evaluating Values and Branching in Procedures 3-7
Testing the Procedure or Job Environment 3-7
Ending Procedures 3-8
* (Comment) Statement 3-9
Substitution Expressions 3-10
?n? (Parameter) Expression 3-11
?n'value'? (Default Parameter) Expression 3-11
?nT'value'? (Temporary Value Parameter) Expression 3-12

?nF'value'? (Forced Value Parameter) Expression ... 3-12
?R? (Required Parameter) Expression ... 3-13
?nR? (Missing Parameter) Expression ... 3-13
?R'mic'? (Required Parameter Message) Expression ... 3-14
?nR'mic'? (Missing Parameter Message) Expression ... 3-14
?Cn? (Parameter Length) Expression ... 3-15
?C'value'? (Length) Expression ... 3-15
?CD? (Return Code) Expression ... 3-16
?CLIB? (Current Library) Expression ... 3-18
?DATE? (Program Date) Expression ... 3-18
?F'S,name'? or ?F'S,name,date'? (File Size) Expression ... 3-18
?F'A,name'? or ?F'A,name,date'? (Actual File Size) Expression ... 3-19
?L'position,length'? (Local Data Area) Expression ... 3-19
?Mmic? or ?M'mic,position,length'? (Message Member) Expression ... 3-21
?MENU? (Current Menu) Expression ... 3-22
?PRINTER? (Session Printer) Expression 3-22
?PROC? (First Level Procedure) Expression ... 3-23
?SLIB? (Session Library) Expression ... 3-23
?SYSLIST? (System List Device) Expression ... 3-23
?TIME? (System Time) Expression ... 3-24
?USER? (Operator's User ID) Expression 3-24
?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression ... 3-25
?WS? (Display Station ID) Expression ... 3-26
Nested Substitution Expressions ... 3-27
IF Conditional Expressions ... 3-28
ACTIVE (Running Procedures) Condition 3-29
BLOCKS (Available Disk Space) Condition 3-30
CONSOLE (System Console) Condition . 3-30
DATAF1 (Files, Libraries, and Folders on Disk) Condition ... 3-31
DATAI1 (Files on Diskette) Condition .. 3-32
DATAT (Files on Tape) Condition ... 3-34
DSPLY (Display Station Type) Condition 3-36
ENABLED (Enabled Communications) Condition ... 3-37
EVOKED (Evoked Procedures) Condition 3-38
INQUIRY (Inquiry Mode) Condition ... 3-39
JOBQ (Job Queue) Condition ... 3-40
LISTDONE (Phone List Completion) Condition ... 3-41
LOAD (Library Load Members) Condition 3-42
MRTMAX (Multiple Requesting Terminals) Condition ... 3-43
PROC (Library Procedure Members) Condition ... 3-44
SECURITY (Password Security) Condition 3-45
SOURCE (Library Source Members) Condition ... 3-46
SUBR (Library Subroutine Members) Condition ... 3-47
SWITCH (Switches) Condition ... 3-48
SWITCHn (Individual Switches) Condition 3-49
string1=string2 (Comparing, Equal to) Condition ... 3-50
string1>string2 (Comparing, Greater Than) Condition ... 3-52
VOLID (Diskette and Tape Volume IDs) Condition ... 3-53
Statement Portion of IF Conditional Expression ... 3-54
ELSE Expressions ... 3-55
// * (Informational Message) Statement ... 3-57
// ** (System Console Message) Statement 3-59
CANCEL Statement ... 3-60
EVALUATE Statement ... 3-60
Assigning Values ... 3-61
Adding, Subtracting, Multiplying, or Dividing ... 3-63
Evaluating Substitution Expressions ... 3-65
Setting the Return Code ... 3-66
GOTO and TAG Statements ... 3-67
PAUSE Statement ... 3-69
RESET Statement ... 3-70
RETURN Statement ... 3-71

Chapter 4. Procedures ... 4-1
#STRTUP1 Procedure ... 4-4
#STRTUP2 Procedure ... 4-6
ALERT Procedure ... 4-7
ALOCFLDR Procedure ... 4-8
ALOCLIBR Procedure ... 4-9
ALTERBSC Procedure ... 4-10
ALTERCOM Procedure ... 4-11
Tributary Station Addressing and Polling Characters ... 4-17
ALTERSDL Procedure ... 4-18
APPNINFO Procedure ... 4-19
ARCHIVE Procedure ... 4-22
ASM Procedure ... 4-27
ASMLOAD Procedure ... 4-29
ASMSAVE Procedure ... 4-30
AUTO Procedure ... 4-30
AUTOC Procedure ... 4-31
BACKUP Procedure ... 4-34
| BALPRINT Procedure ... 4-34
BASIC Procedure ... 4-39
BASICP Procedure ... 4-40

BASICR Procedure 4-41
BASICS Procedure 4-42
BASLOAD Procedure 4-44
BASSAVE Procedure 4-45
BGUATTR Procedure 4-46
BGUCHART Procedure 4-47
BGUDATA Procedure 4-50
BGUGRAPH Procedure 4-51
BGULOAD Procedure 4-54
BGUSAVE Procedure 4-55
BLDFILE Procedure 4-56
BLDINDEX Procedure 4-59
BLDLIBR Procedure 4-62
BLDMENU Procedure 4-66
BUILD Procedure 4-70
CACHE Procedure 4-72
CATALOG Procedure 4-73
Sample Disk VTOC Listings 4-77
Sample Diskette VTOC Listings 4-84
Sample Tape Label Listings 4-87
CGU Procedure 4-91
CGULOAD Procedure 4-92
CGUSAVE Procedure 4-93
CHGXLATE Procedure 4-93
CHNGEMEM Procedure 4-94
CNFIGICF Procedure 4-96
CNFIGSSP Procedure 4-97
CNFIGX25 Procedure 4-97
COBLOAD Procedure 4-98
COBOL Procedure 4-98
COBOLC Procedure 4-99
COBOLCG Procedure 4-101
COBOLG Procedure 4-102
COBOLONL Procedure 4-102
COBOLP Procedure 4-102
COBSAVE Procedure 4-103
COBSDA Procedure 4-103
COBSEU Procedure 4-104
COMPRESS Procedure 4-105
CONDENSE Procedure 4-109
COPYDATA Procedure 4-111
COPYDIAG Procedure 4-120
COPYI1 Procedure 4-121
COPYPRT Procedure 4-126
File Format for COPYPRT Files 4-128
CREATE Procedure 4-132
Message Member Statements 4-133
Considerations for the Ideographic Version of the SSP 4-135
DATE Procedure 4-137
DEFINEID Procedure 4-139
DEFINEPN Procedure 4-140
DEFINLOC Procedure 4-140
DEFINX21 Procedure 4-141
DEFINX25 Procedure 4-142
DEFSUBD Procedure 4-143
DELETE Procedure 4-144
DELNRD Procedure 4-150
DFU Procedure 4-150
DFULOAD Procedure 4-151
DFUSAVE Procedure 4-152
DICTLOAD Procedure 4-153
DICTSAVE Procedure 4-155
DISABLE Procedure 4-157
DISPLAY Procedure 4-158
DLSLOAD Procedure 4-159
DLSSAVE Procedure 4-159
DOCCNV Procedure 4-160
DOCPLOAD Procedure 4-161
DOCPSAVE Procedure 4-162
DSU Procedure 4-163
DSULOAD Procedure 4-166
DSUSAVE Procedure 4-167
EDITNRD Procedure 4-167
EM3270 Procedure 4-168
ENABLE Procedure 4-170
ENTER Procedure 4-172
EPDOWNL Procedure 4-174
EPLMRG Procedure 4-174
EP3270 Procedure 4-175
ERR Procedure 4-176
ES3270 Procedure 4-178
EXTRACT Procedure 4-180
FORMAT Procedure 4-181
FORTC Procedure 4-184
FORTCG Procedure 4-184
FORTG Procedure 4-184
FORTGO Procedure 4-185
FORTLOAD Procedure 4-186
FORTONL Procedure 4-187
FORTP Procedure 4-187
FORTRANC Procedure 4-188
FORTSAVE Procedure 4-191
FORTSDA Procedure 4-191
FORTSEU Procedure 4-192
FROMLIBR Procedure 4-193
HELP Procedure 4-199
HISTCRT Procedure 4-206
HISTCOPY Procedure 4-207
HISTORY Procedure 4-209
Sample HISTORY Display 4-214
Sample HISTORY Listing 4-218
Description of the Items on the History File Listing 4-219
ICFDEBUG Procedure 4-221
ICVERIFY Procedure 4-222
IDDUDCT Procedure 4-222
IDDUDFN Procedure 4-223

IDDUDISK Procedure 4-223
IDDULINK Procedure 4-224
IDDUPRT Procedure 4-226
IDDURBLD Procedure 4-227
IDDUXLAT Procedure 4-227
INIT Procedure 4-233
INITDIAG Procedure 4-236
INIT9332 Procedure 4-237
INQUIRY Procedure 4-238
IPL Procedure 4-240
ITF Procedure 4-241
IWLOAD Procedure 4-242
| IWPTLOAD Procedure 4-243
| IWPTSAVE Procedure 4-244
IWSAVE Procedure 4-245
JOBSTR Procedure 4-246
KEYS Procedure 4-250
KEYSORT Procedure 4-252
| LANLOAD Procedure 4-253
| LANSAVE Procedure 4-254
LIBRLIBR Procedure 4-255
LINES Procedure 4-257
LIST Procedure 4-259
LISTDATA Procedure 4-261
LISTFILE Procedure 4-267
LISTLIBR Procedure 4-272
Sample LISTLIBR Listings 4-277
Description of the Items on the Library Directory Listing 4-282
LISTNRD Procedure 4-292
LOAD3601 Procedure 4-293
LOG Procedure 4-294
| LRTRLOAD Procedure 4-295
| LRTRSAVE Procedure 4-296
MAINTX25 Procedure 4-297
MCSCONV Procedure 4-297
MOVEFLDR Procedure 4-298
MSGFILE Procedure 4-299
MSRJE Procedure 4-302
NOHALT Procedure 4-305
OFCBPRT Procedure 4-307
OFCCAL Procedure 4-309
OFCCANCL Procedure 4-310
OFCCOMM Procedure 4-310
OFCCONV Procedure 4-311
OFCDATA Procedure 4-312
OFCDFLT Procedure 4-313
OFCDIR Procedure 4-314
OFCFILE Procedure 4-315
OFCGRP Procedure 4-315
OFCINSTL Procedure 4-316
OFCLDF Procedure 4-316
OFCLOAD Procedure 4-317
OFCMAIL Procedure 4-318
OFCMAINT Procedure 4-320
OFCMSG Procedure 4-321
OFCQ Procedure 4-322
OFCSAVE Procedure 4-322
OFCSRCH Procedure 4-323
OFCSTAT Procedure 4-323
OFCUSER Procedure 4-324
OLINK Procedure 4-325
OLPDLOAD Procedure 4-328
OLPDSAVE Procedure 4-329
ORGANIZE Procedure 4-329
OVERRIDE Procedure 4-329
PASSTHRU Procedure 4-330
PASSWORD Procedure 4-331
PCEXCH Procedure 4-332
PCEXEC Procedure 4-334
PCOLOAD Procedure 4-335
PCOPROF Procedure 4-336
PCOSAVE Procedure 4-337
PCU Procedure 4-338
POST Procedure 4-345
PRINT Procedure 4-350
PRINTKEY Procedure 4-354
PROBLEM Procedure 4-356
| PROFLOAD Procedure 4-357
| PROFSAVE Procedure 4-358
PRTGRAPH Procedure 4-359
QRY Procedure 4-360
QRYDE Procedure 4-361
QRYLOAD Procedure 4-362
QRYRUN Procedure 4-363
QRYSAVE Procedure 4-367
READINFO Procedure 4-368
REBLD Procedure 4-369
RELOAD Procedure 4-369
REMOVE Procedure 4-370
RENAME Procedure 4-372
REQUESTX Procedure 4-373
RESPONSE Procedure 4-374
Automatic Response Source Statements . 4-376
Automatic Response Programming Considerations 4-378
RESTEXTN Procedure (for the Ideographic Version of the SSP) 4-380
RESTFLDR Procedure 4-383
RESTLIBR Procedure 4-386
RESTNRD Procedure 4-389
RESTORE Procedure 4-392
RETRIEVE Procedure 4-401
RJFILE Procedure 4-405
RJTABLE Procedure 4-405
ROLLKEYS Procedure 4-406
RPG Procedure 4-406
RPGC Procedure 4-407

RPGLOAD Procedure 4-410
RPGONL Procedure 4-411
RPGP Procedure 4-411
RPGR Procedure 4-412
RPGSAVE Procedure 4-413
RPGSDA Procedure 4-414
RPGSEU Procedure 4-414
RPGX Procedure 4-415
SAVE Procedure 4-416
SAVEEXTN Procedure (for the Ideographic Version of the SSP) 4-425
SAVEFLDR Procedure 4-428
SAVELIBR Procedure 4-431
SAVENRD Procedure 4-434
SDA Procedure 4-437
SDALOAD Procedure 4-439
SDASAVE Procedure 4-440
SECDEF Procedure 4-441
SECEDIT Procedure 4-443
SECLIST Procedure 4-445
SECREST Procedure 4-448
SECSAVE Procedure 4-451
SET Procedure 4-454
| SETALERT Procedure 4-457
| Alert Source Statements 4-458
SETCOMM Procedure 4-461
SEU Procedure 4-465
SEULOAD Procedure 4-467
SEUSAVE Procedure 4-468
SHRFLOAD Procedure 4-469
SHRFSAVE Procedure 4-470
SLIB Procedure 4-471
SMF Procedure 4-473
SMFDATA Procedure 4-474
SMFPRINT Procedure 4-477
SMFSTART Procedure 4-480
SMFSTOP Procedure 4-482
SORT Procedure 4-483
SPECIFY Procedure 4-483
SRTX Procedure 4-484
SRTXBLD Procedure 4-485
SRTXLOAD Procedure 4-486
SRTXSAVE Procedure 4-488
STARTM Procedure 4-489
STATEST Procedure 4-490
STOPGRP Procedure 4-491
STOPM Procedure 4-492
STRTGRP Procedure 4-493
| SWDLOAD Procedure 4-494
| SWDSAVE Procedure 4-495
SWITCH Procedure 4-496
SYSLIST Procedure 4-498
TAPECOPY Procedure 4-500
TAPEINIT Procedure 4-507
TAPESTAT Procedure 4-509
TEXTCONV Procedure 4-511
TEXTDCT Procedure 4-511
TEXTDOC Procedure 4-512
File Format for PRTFILE Files 4-516
TEXTFLDR Procedure 4-530
TEXTLOAD Procedure 4-531
TEXTOBJ Procedure 4-532
| TEXTPROF Procedure 4-532
TEXTPRTQ Procedure 4-533
TEXTREL Procedure 4-534
TEXTSAVE Procedure 4-536
TOLIBR Procedure 4-537
TRANSFER Procedure 4-542
Basic Data Exchange Files 4-548
I-Exchange Files 4-548
| TRNMGR Procedure 4-549
UPDATE Procedure 4-550
| WSFLOAD Procedure 4-552
| WSFSAVE Procedure 4-553
WSU Procedure 4-554
WSULOAD Procedure 4-557
WSUSAVE Procedure 4-558
WSUTXCR Procedure 4-559
WSUTXEX Procedure 4-561
WSUTXRV Procedure 4-563
XREST Procedure 4-563
XSAVE Procedure 4-563

Chapter 5. OCL Statements 5-1
Placement of OCL Statements 5-1
Types Of Information Contained In OCL Statements 5-2
Identifiers 5-2
OCL Parameters 5-2
Procedure Parameters 5-3
General OCL Coding Rules 5-4
Continuing OCL Statements 5-4
Comments 5-5
ABEND OCL Statement 5-7
ALLOCATE OCL Statement 5-8
ATTR OCL Statement 5-11
| CANCEL OCL Statement 5-16
| CHANGE OCL Statement 5-18
COMM OCL Statement 5-21
COMPILE OCL Statement 5-23
DATE OCL Statement 5-25
DEALLOC OCL Statement 5-27
DEBUG OCL Statement 5-28
EVOKE OCL Statement 5-30
FILE OCL Statement (for Disk Files) 5-32
FILE OCL Statement (for Diskette Files) .. 5-43
FILE OCL Statement (for Tape Files) 5-48
FORMS OCL Statement 5-55

IMAGE OCL Statement 5-59
Creating Your Own Print Belt Members . 5-61
INCLUDE OCL Statement 5-63
INFOMSG OCL Statement 5-65
JOBQ OCL Statement 5-66
LIBRARY OCL Statement 5-67
LOAD OCL Statement 5-69
LOCAL OCL Statement 5-70
LOG OCL Statement 5-72
MEMBER OCL Statement 5-73
MENU OCL Statement 5-75
MSG OCL Statement 5-76
NOHALT OCL Statement 5-79
OFF OCL Statement 5-81
POWER OCL Statement 5-82
PRINTER OCL Statement 5-83
PROMPT OCL Statement 5-97
REGION OCL Statement 5-103
RESERVE OCL Statement 5-104
RUN OCL Statement 5-105
SESSION OCL Statement 5-106
START OCL Statement 5-109
STOP OCL Statement 5-111
SWITCH OCL Statement 5-112
SYSLIST OCL Statement 5-113
VARY OCL Statement 5-116
WAIT OCL Statement 5-118
WORKSTN OCL Statement 5-120
/* (End Of Data) Statement 5-123

Chapter 6. Control Commands 6-1
ASSIGN Control Command 6-2
CANCEL Control Command 6-5
CHANGE Control Command 6-9
CONSOLE Control Command 6-15
HOLD Control Command 6-18
INFOMSG Control Command 6-20
JOBQ Control Command 6-21
MENU Control Command 6-22
MODE Control Command 6-23
MSG Control Command 6-24
OFF Control Command 6-27
POWER Control Command 6-28
PRTY Control Command 6-29
RELEASE Control Command 6-30
REPLY Control Command 6-32
RESTART Control Command 6-34
START Control Command 6-36
STATUS Control Command 6-40
STATUSF Control Command 6-45
STOP Control Command 6-48
TIME Control Command 6-52
VARY Control Command 6-53

Appendix A. SSP Utility Programs A-1
Making Your Own Procedures Using SSP Utility Programs A-1
$ARSP Utility A-3
Changing Automatic Response Values (RESPONSE Procedure) A-3
Changing Alert Indicators (SETALERT Procedure) A-3
$BICR Utility A-4
Listing Diskette Files (LISTFILE Procedure) A-4
Copying Files (TRANSFER Procedure) .. A-5
$BMENU Utility A-7
$BUILD Utility A-9
$COPY Utility A-9
Copying Disk Files (COPYDATA Procedure) A-10
Listing $COPY Files (LISTDATA/LISTFILE Procedures) .. A-14
Restoring Files (RESTORE Procedure) .. A-18
Restoring the Network Resource Directory (RESTNRD Procedure) A-23
Saving Files (SAVE Procedure) A-24
Saving the Network Resource Directory (SAVENRD Procedure) A-31
IBM System/34 Compatible Statements . A-32
$CPPE Utility A-37
$CZUT Utility A-38
$DCOPY Utility A-38
$DDST Utility A-39
$DELET Utility A-40
$DPGP Utility A-42
$DUPRD Utility A-43
$FBLD Utility A-44
Creating Files (BLDFILE Procedure) ... A-44
Creating Alternative Indexes (BLDINDEX Procedure) A-46
$FREE Utility A-47
$HELP Utility A-47
$HIST Utility A-48
$IDSET Utility A-49
$IEDS Utility A-50
$IENBL Utility A-51
$INIT Utility A-52
$LABEL Utility A-53
$MAINT Utility A-55
Create a Library (BLDLIBR Procedure) . A-56
Create Source or Procedure Members ... A-57
Change Library or Directory Size (ALOCLIBR Procedure) A-60
Change Library Member Information (CHNGEMEM Procedure) A-62

Gather Unused Library Space (CONDENSE Procedure) A-63
Copy Members from One Library to Another (LIBRLIBR Procedure) A-64
Copy Members from a Library (FROMLIBR Procedure) A-66
COPY and CEND Statements A-69
Copy Members to a Library (TOLIBR Procedure) A-72
Start a Job (JOBSTR Procedure) A-74
List Library Members and Information (LISTLIBR Procedure) A-75
List Information about Libraries (LISTFILE Procedure) A-77
Remove Members from a Library (REMOVE Procedure) A-78
Save a Library (SAVELIBR Procedure) . A-79
Restore a Library (RESTLIBR Procedure) A-80
$MGBLD Utility A-82
$MMSP Utility A-83
$MMST Utility A-83
$PACK Utility A-84
$PNLM Utility A-84
$POST Utility A-85
Copy Special E-Format Diskette Files or Basic Data Exchange Diskette Files (POST Procedure) A-85
List Basic Data Exchange Diskette Files . A-86
$PRCED Utility A-87
$PRCLT Utility A-87
$PRLST Utility A-88
$PRPWD Utility A-88
$PRUED Utility A-89
$PRUID Utility A-90
$PRURS Utility A-91
$PRUSV Utility A-93
$RENAM Utility A-94
$RREDT Utility A-95
$RRESC Utility A-96
$RRLST Utility A-97
$RRSAV Utility A-98
$RRSTR Utility A-99
$RRTED Utility A-101
$RRTLT Utility A-101
$SETCF Utility A-102
Change Communications Attributes (ALTERCOM Procedure) A-102
Set Display Station Environment (SET Procedure) A-104
Specify Print Key Information (PRINTKEY Procedure) A-105
$SETCP Utility A-106
$SFGR Utility A-108
$SINCT Utility A-110
$SINDL Utility A-110
$SINR Utility A-110
$SVCASRV Utility A-111
$TCOPY Utility A-112
Copy Data To or From Tape (TAPECOPY Procedure) A-112
$TINIT Utility A-114
$TMSERV Utility A-115
Copy a Folder Member onto Disk, Diskette, Tape, or Tape Cartridge (ARCHIVE Procedure) A-115
Copy a Folder onto Disk, Diskette, Tape, or Tape Cartridge (SAVEFLDR Procedure) A-116
List the Contents of an Archived Folder Member (LISTFILE Procedure) A-117
Reorganize a Folder (ALOCFLDR and CONDENSE Procedures) A-118
Move a Folder (MOVEFLDR Procedure) A-119
Restore a Folder Member from Disk, Diskette, Tape, or Tape Cartridge (RETRIEVE Procedure) A-120
Restore a Folder from Disk, Diskette, Tape, or Tape Cartridge (RESTFLDR Procedure) A-121
$UASC Utility A-122
$UASF Utility A-123
$XNLM Utility A-124
$XNSH Utility A-124
$XREST Utility A-125
$XSAVE Utility A-126
#GCFR Utility A-127

Appendix B. Conversions B-1
Converting Hexadecimal to Decimal B-1
Hexadecimal to Decimal Example B-2
Decimal to Hexadecimal Example B-2
Records to Blocks Conversion for Disk Files . B-3
Determining the Number of Blocks in a Sequential or Direct File B-3
Determining the Number of Blocks in an Indexed File B-3

Appendix C. Service Aid Procedures C-1
APAR Procedure C-2
DFA Procedure C-5
DUMP Procedure C-7
ERAP Procedure C-16
PATCH Procedure C-17
PTF Procedure C-18
SERVICE Procedure C-29
SERVLOG Procedure C-30
SETDUMP Procedure C-31
TRACE Procedure C-32

Appendix D. Multinational Character Set Conversion Utility D-1
Introduction D-1
5250 Hardware Support D-2
Starting the Conversion Utility D-3
Library Member Conversion D-4
Library Member Modification D-7
Conversion Tables D-10
Source Statement Length D-10
MCSCU Library Member Listing D-11
Library Statement Modification D-13
RPG Source Members D-13
Procedure Members D-14
Message Members D-15
Menu Members D-15
Display Formats D-15
Sort Members D-15
Work Station Utility D-16
Other Statement Type D-16
Copying Back into the Library D-17
Additional Library Member Considerations . D-17
Data File Conversion D-18
MCSCU Data File Listing D-21
Data File Modification D-22
Document Folder Conversion D-23
MCSCU Document Folder Listing D-25
Data Dictionary Conversion D-27
MCSCU Data Dictionary Listing D-29
Batch Interface D-31
Library Members D-31
Disk Files D-34
Document Folders D-36
Data Dictionaries D-38
Changed Characters by Language Group . D-39

Appendix E. 3262 Printer Print Belts and Translation Tables E-1
3262 Print Belts E-1
3262 Translation Tables E-7
Table #96E48 ..... E-8
Table #96E64 ..... E-10
Table #188E48 ..... E-12
Table #188E64 ..... E-14
Table #188E96 ..... E-16

Appendix F. EBCDIC and ASCII Code Tables F-1
EBCDIC F-1
ASCII F-2

Appendix G. Using the IDDUXLAT Procedure G-1
Introduction G-1
Considerations and Rules G-1
What You Should Consider Before You Run IDDUXLAT G-1
Rules that Apply to the Source Specifications G-4
Example G-8
Input to IDDUXLAT: RPG Source Specifications G-8
The IDDUXLAT Parameters This Example Uses G-11
Output from IDDUXLAT: The PRINT Option G-12

Glossary H-1

Index X-1

About This Manual

Who should use this manual . . .

This manual provides programmers with information needed to create procedures for the IBM System/36. It also provides operators and programmers with the reference information needed to identify and use procedures, control commands, operation control language (OCL) statements, and system support program (SSP) utility programs.

This section describes what this manual contains and the method used to show the statement formats.

How this manual is arranged . . .

This manual contains the following chapters and appendixes:

Chapter 1 introduces the system support and the method used to describe the syntax formats shown in this manual. This chapter also contains a directory to how the system support can be used to do tasks. This directory is indicated by a bar on the edge of the page; use this bar to help you find this directory.

Chapter 2 describes how you can write your own procedures.

Chapter 3 describes each expression and statement you can use in procedures. These are called procedure control expressions.

Chapter 4 describes each IBM-supplied procedure.

Chapter 5 describes each operation control language (OCL) statement.

Chapter 6 describes each control command.

Appendix A shows the OCL and utility control statements you can use instead of the IBM-supplied SSP procedures. Only the SSP procedures are shown; the procedures for other IBM-supplied procedures are not shown. You can use this chapter to write your own procedures based on the IBM-supplied SSP procedures. Also, if an SSP utility program has functions that are not supported by IBM-supplied SSP procedures, those additional functions are described in this appendix.

Appendix B describes the relationship of disk records, blocks, and sectors.

Appendix C describes the service aid procedures.

Appendix D describes the multinational character set conversion programs.

Appendix E lists the characters on the standard 48-, 48HN-, 64B-, 64C-, 96-, and 188-character print belts. This appendix also lists translation tables for print belt character translation.

Appendix F lists the EBCDIC and ASCII code tables. This appendix also shows tables for hexadecimal and decimal conversion.

Appendix G describes the IDDUXLAT procedure.

Glossary defines terms and abbreviations used in this manual.

Index provides page numbers for topics covered in this manual.

What you should know . . .

- *IBM System/36: Learning about Your Computer*, SC21-9018, contains introductory material about the IBM System/36. You should read this manual first if you are not familiar with the System/36.
- IBM System/36 *Concepts and Programmer's Guide*, SC21-9019, describes how the system functions. It also contains information about techniques to use when programming the System/36.

If you need more information . . .

You might need some or all of the following IBM manuals before or while you are using this manual. Except where otherwise indicated, each is a System/36 manual.

General SSP related manuals

- *Guide to Publications*, GC21-9015, contains a general guide to the System/36 publications, a glossary of all terms used in System/36 publications, and a list of topics and the manuals in which each topic can be found.
- *Using Your Display Station*, SC21-9455, describes how to use menus, commands, and procedures to operate your display station.
- *Operating Your System-5360, 5362*, SC21-9452, describes how to operate the System/36, 5360 and 5362 System Units.
- *Operating Your System-5364*, SC21-9453, describes how to operate the System/36, 5364 System Unit.
- *System Messages*, SC21-7938, describes the messages the System/36 displays or prints.
- *Creating Displays: Screen Design Aid and System Support Program*, SC21-7902, describes how to create and change display formats and menus by using either the screen design aid (SDA) or the System Support Program (SSP).
- *Changing Your System Configuration*, SC21-9052, describes how to configure your system and how to install the system programs.
- *System Security Guide*, SC21-9042, describes how to secure your system from unauthorized users and how to protect your data.
- *System Measurement Facility Guide*, SC21-9025, describes the system measurement facility (SMF) procedures, and how to interpret the information printed by these procedures.
- *System Problem Determination*, SC21-7919, describes what you can do before calling for IBM to service your system.
- *Diskette General Information Manual*, GA21-9182, describes diskettes and how to handle them. This is not a System/36 manual.

Programming language and utility manuals

- *Source Entry Utility Guide*, SC21-7901, describes how to run the source entry utility (SEU) to create and change procedures and source members.
- *Programming with RPG II*, SC21-9006.
- *Programming with BASIC*, SC21-9003.
- *Programming with COBOL*, SC21-9007.
- *Programming with FORTRAN IV*, SC21-9005.
- *Programming with Assembler*, SC21-7908.
- *Overlay Linkage Editor Guide*, SC21-9041, describes how to link-edit compiled programs.
- *Sort Guide*, SC21-7903, describes how to sort data files using the sort program.
- *Data File Utility Guide*, SC21-7900, describes how to create, change, display, and list data files using the data file utility (DFU).
- *Work Station Utility Guide*, SC21-7905, describes how to create programs using the work station utility (WSU).

- *Character Generator Utility Guide*, SC09-1055, describes how to create ideographic characters using the character generator utility (CGU).
- *Ideographic Sort Utility Guide*, SC09-1054, describes how to sort ideographic data using the ideographic sort utility.
- *Business Graphics Utilities/36 User's Guide*, SC21-7985, describes how to design graphs and charts using BGU/36.
- *Development Support Utility Guide*, SC09-1085, describes how to create, edit, print, remove and view procedure members and library source members using the development support utility (DSU).

Communications manuals

- *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference*, SC21-9533, describes the Interactive Communications Feature (SSP-ICF) subsystems.
- *Interactive Communications Feature: Upline Subsystems Reference*, SC21-9532, describes the Interactive Communications Feature (SSP-ICF) subsystems.
- *Interactive Communications Feature: Finance Subsystem Reference*, SC21-9531, describes the Interactive Communications Feature (SSP-ICF) subsystems.
- *Interactive Communications Feature: Base Subsystems Reference*, SC21-9530, describes the Interactive Communications Feature (SSP-ICF) subsystems.
- *Interactive Communications Feature: Guide and Examples*, SC21-7911, describes the Interactive Communications Feature (SSP-ICF) subsystems.
- *3270 Device Emulation Guide*, SC21-7912, describes how to use 3270 device emulation.
- *Multiple Session Remote Job Entry Guide*, SC21-7909, describes how to use remote job entry.
- *Programming with RPG II*, SC21-9006, describes how to use batch BSC with RPG.
- *Programming with Assembler*, SC21-7908, describes how to use batch BSC with Assembler.
- *Communications and Systems Management Guide*, SC21-8010, describes the remote management support (also referred to as DHCF), the change management support (also referred to as DSNX), and the problem management support (which allows System/36 to generate and send alerts).
- *Advanced Peer-to-Peer Networking (APPN) Guide*, SC21-9471, describes networking for the System/36.
- *Distributed Data Management Guide*, SC21-8011, describes how to use DDM and the network resource directory.
- *PC Support/36 User's Guide*, SC21-9088, describes the instructions necessary to load and operate PC Support/36 and perform problem determination procedures.
- *PC Support/36 Technical Reference*, SC21-9097, describes the PC Support/36 utility and the support it provides for the IBM Personal Computer.
- *Using System/36 Communications*, SC21-9082, describes the support provided by the base Communications feature.
- *Using the Asynchronous Communications Support*, SC21-9143, describes the asynchronous communications support.

If you need programming or debugging material

- *IBM 5250 Keyboard Template Assignment Sheet and Display Screen Layout Sheet*, GX21-9271, is a layout sheet to help you make display formats.
- *IBM Display Format Specifications*, GX21-9800, are coding sheets to help you make display formats.

- *IBM RPG Debugging Template*, GX21-9129, is a template to help you debug RPG problems.
- *IBM WSU/$SFGR Debugging Template*, GX21-7926, is a template to help you debug display format and WSU problems.
- *IBM Keyboard Template*, GX21-7929, shows the command keys used with the System/36 program products.
- *IBM Command Key Template*, GX21-9799, is an adhesive template to be used with 5251 display stations.
- *IBM CGU Keyboard Template (Large Keyboard)*, SC09-1027, shows the command keys used with the character generator utility.
- *IBM CGU Keyboard Template (Small Template)*, SC09-1028, shows the command keys used with the character generator utility.
- The following are blank, plastic templates on which you can place information about the command keys used by your programs:
  - *IBM 5251 Models 1 and 11 and IBM 5252 Dual Display Station Keyboard Template*, GX21-9266
  - *IBM 5251 Display Station Models 2 and 12 Keyboard Template*, GX21-9327
  - *IBM 5291 Display Station Keyboard Template*, GX21-9410
  - *IBM 5292 Color Display Station Keyboard Template*, GX21-9414
- *IBM 5292 Color Display Station Select Options*, GX21-9451, shows how to control the special features of the 5292 display.

If you do not understand a term used in this manual

See the *Glossary* at the back of this manual if you do not understand a term used in this manual. Many terms and concepts are introduced in the manual *Learning about Your Computer*. If you are unfamiliar with the System/36, you should read that manual first.

Insert tabs

Insert tabs are available to divide sections of this manual. This will help you find information quickly. Requests for insert tabs should be made to your IBM representative or the IBM branch office serving your locality. The title and order number is: *Insert Tabs for the IBM System/36 System Reference*, SX21-9801.

How this manual has changed . . .

The following procedures were added:

- BALPRINT: Allows the user to balance spooled output among a group of printers.
- EPLMRG: Allows the user to merge the personal computer machine-readable instruction files and translated tables from the system library into a virtual diskette.
- IWPTLOAD: Allows the user to copy PC Support/36 pass-through support to the PC Support/36 and system libraries from a backup diskette.
- IWPTSAVE: Allows the user to copy PC Support/36 pass-through support onto diskette from the PC Support/36 and system libraries.
- LANLOAD: Allows the user to copy LAN communications support to the LAN and system libraries from a backup diskette.
- LANSAVE: Allows the user to copy LAN communications support onto diskette from the LAN and system libraries.
- LRTRLOAD: Allows the user to copy the IBM Token-Ring Network to the PC Support/36 and system libraries from a backup diskette.
- LRTRSAVE: Allows the user to copy the IBM Token-Ring Network onto diskette from the PC Support/36 and system libraries.
- PROFLOAD: Allows the user to copy PROFS bridge support to the Personal Services/36 and system libraries from a backup diskette.

- PROFSAVE: Allows the user to copy PROFS bridge support onto diskette from the Personal Services/36 and system libraries.
- SETALERT: Allows the user to change the alert indicators for messages in a message load member.
- SWDLOAD: Allows the user to copy software distribution support to the software distribution and system libraries from a backup diskette.
- SWDSAVE: Allows the user to copy software distribution support onto diskette from the software distribution and system libraries.
- TEXTPROF: Allows the user to create or maintain DW/36 user profiles.
- TRNMGR: Allows the user to start, stop, or change error reporting in an IBM Token-Ring Network.
- WSFLOAD: Allows the user to copy the PC Support/36 work station feature to the PC Support/36 and system libraries from a backup diskette.
- WSFSAVE: Allows the user to copy the PC Support/36 work station feature onto diskette from the PC Support/36 and system libraries.

The IWDOWNL procedure was deleted.

The following OCL statements were added:

- CANCEL: Allows the user to cancel spool file entries.
- CHANGE: Allows the user to change spool file entries.
- START: Allows the user to start a printer's spool writer.
- STOP: Allows the user to stop a printer's spool writer.

Additional parameters were added to:

- The PRINT procedure
- The ATTR, FORMS, MSG, and PRINTER OCL statements
- The CANCEL, CHANGE, and MSG control commands
- The $ARSP utility program

Other changes were made throughout the manual.

Changes since the previous edition of the manual are indicated by a vertical line to the left of the change.

Note: This manual may refer to products that are announced, but are not yet available. Such information is for planning purposes only and is subject to change before general availability.

Chapter 1. Introduction

This manual describes the procedures, control commands, operation control language (OCL) statements, and procedure control expressions supplied with the system support program (SSP) and other program products.

Procedures

You use procedures to do a task on the system, such as listing the contents of a disk file or running a program.

A **procedure** is a collection of statements that cause one or more programs to be run. A **command** is an instruction that tells the system to do something. Procedures make it possible to avoid entering frequently used statements each time they are required.

To run a procedure, you enter a **procedure command** at a keyboard. A procedure command contains the name of the procedure to be run and optional information that defines the function to be done by the procedure. For example, you could enter the following to run an IBM-supplied procedure named SYSLIST:

```
SYSLIST
```

This procedure command contains only the name of the procedure to be run.

Procedure commands are usually entered with information that tells the procedure what to do. The following example shows how a procedure command could be entered to run an IBM-supplied procedure named LISTLIBR to list a library member named PAYROLL:

```
LISTLIBR PAYROLL
```

Another way to run a procedure is to create a menu. Menus allow you to enter a number that corresponds to the procedure command, rather than having to enter the procedure command. See the manual *Creating Displays* for more information about creating menus.

Many procedures are supplied as part of the SSP and the other program products. The SSP procedures allow you to create data files and libraries; save, restore, and copy data files and libraries; and list information about data files, libraries, and the system. The program product procedures allow you to create and change library members and to compile programs. See Chapter 4, "Procedures" for detailed information on the SSP and program product procedures.

You can make your own procedures for the System/36. See Chapter 2, "Making Your Own Procedures" for this information.

OCL Statements

The **operation control language** (OCL) statements provide the SSP with all the information it must have about jobs to be run. OCL statements are normally contained in procedures, although they can be entered from a keyboard. See Chapter 5, "OCL Statements" for detailed information about OCL statements.

SSP Utility Programs and Their Control Statements

SSP utility programs are supplied by IBM as part of the SSP to do certain functions, such as copying a disk file, listing a library member, or communicating with another system.

When an SSP utility program is run, OCL statements identify the program and supply additional information that is required. Besides OCL statements, **utility control statements** define the functions to be done by the SSP utility program. Normally, a procedure contains the OCL statements and utility control statements required to do a job.

The following example shows the statements needed to list a library member named PAYROLL. The $MAINT utility program is run. The OCL statements are the LOAD and RUN statements; the utility control statements are the COPY and END statements.

```
// LOAD $MAINT
// RUN
// COPY FROM-PAYLIB,TO-PRINT,NAME-PAYROLL,LIBRARY-S
// END
```

The statements in the example indicate the following:

LOAD This OCL statement loads $MAINT into main storage. $MAINT is one of the SSP utility programs.

RUN This OCL statement starts running the $MAINT program.

COPY This utility control statement passes information to the $MAINT program. The information is:

FROM-PAYLIB	The library that contains the member is named PAYLIB.
TO-PRINT	The information is to be printed.
NAME-PAYROLL	A library member named PAYROLL is to be used.
LIBRARY-S	The library member is a source member.

Using all four pieces of information, the $MAINT program is told to print a library source member named PAYROLL from the library named PAYLIB.

END This utility control statement indicates that no more utility control statements follow.

The SSP utility programs you can use for each SSP procedure are described in Appendix A, "SSP Utility Programs."

Control Commands

Control commands are used to control the system, the printers, and the display stations. The following example shows how you could stop the printing of spooled output on printer P3:

```
STOP PRT,P3
```

See Chapter 6, "Control Commands" for detailed descriptions of the control commands.

Concept Information and Programming Considerations

For introductory information about computers, see the manual *Learning about Your Computer.* For information about system concepts and programming techniques, see the *Concepts and Programmer's Guide.*

Conventions Used for Describing Syntax Formats

When syntax formats are shown in this manual, capitalized expressions, brackets, braces, parentheses, and underlining have special meanings.

Capitalized Expressions

Capitalized expressions must be entered as they are shown in the syntax formats. Numbers and special characters within a capitalized expression also must be entered as they are shown. An expression that is not capitalized must be replaced with a value that is appropriate. For example:

```
// FORMS LINES-value
```

S9020001-0

could be coded:

```
// FORMS LINES-66
```

Braces { }

Braces in a syntax format are not coded as part of the command or statement. Braces indicate that one of the values enclosed within the braces must be coded. For example:

```
// DATE    { mmddyy }
           { ddmmyy }
           { yymmdd }
```

S9020002-0

indicates that if you choose to code a date, it must be in one of the three formats shown: mmddyy, ddmmyy, or yymmdd. For example: April 14, 1985 could be entered: 041485 (mmddyy), 140485 (ddmmyy), or 850414 (yymmdd).

Brackets [ ]

Brackets in a syntax format are not coded as part of the command or statement. Brackets indicate that the expression they enclose is optional. If a list of values is enclosed in brackets, you can choose not to code a value or to code one of the items in the list. For example:

```
MOVEFLDR  folder name,[ A1           ],[ A1           ]
                      [ A2           ]  [ A2           ]
                      [ A3           ]  [ A3           ]
                      [ A4           ]  [ A4           ]
                      [ block number ]  [ block number ]
```

S9020556-0

| indicates you do not need to code the second and third parameters, but if you choose to code either of them, they must be in one of the following formats:

- The letter "A" with a number from 1 to 4
- A block number

For example, you could enter:

```
MOVEFLDR  MYFLDR,A1,A2
```

Parentheses ( )

Parentheses in a syntax format are not coded as part of the command or statement. Parentheses indicate that the value enclosed within the parentheses is an abbreviation and can be entered in place of the characters above the parentheses. For example:

```
STATUS    SESSION
(D)       (S)
```

S9020004-0

indicates that STATUS SESSION can be entered as any of the following:

```
STATUS SESSION

D S

STATUS S

D SESSION
```

Underlining

Underlining identifies default values. The system automatically uses the default value if you do not code an optional value. For example:

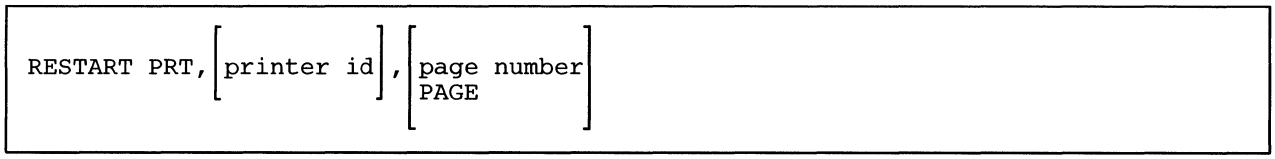
```
OFF    [ DROP ]
       [ HOLD ]
```

S9020005-0

indicates that the system assumes DROP if you do not code the parameter. (Remember, the brackets indicate the parameter is optional.)

Commas

The syntax formats often indicate that commas are required; that is, when the commas are not shown inside brackets. The commas are shown outside the brackets to remind you that if a parameter is omitted, a comma must be entered to indicate the position of the omitted parameter (when one or more parameters are coded in positions that follow the omitted parameter). For example:



S9020006-0

indicates that if the second parameter is not coded but the third parameter is, a comma should be coded to indicate the place of the missing parameter, as in:

```
RESTART PRT,,PAGE
```

For any procedure or control commands you enter, commas following the last parameter coded are optional. For example:

```
RESTART PRT,,
```

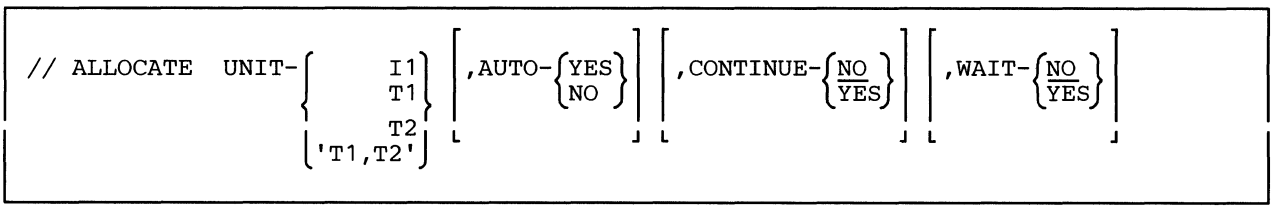
and

```
RESTART PRT
```

are treated the same.

Example Syntax Diagram

The following syntax diagram shows how these diagramming methods can be combined:



S9020007-0

To allocate unit I1 with CONTINUE as YES and WAIT as NO, you could enter:

```
// ALLOCATE UNIT-I1,AUTO-YES,CONTINUE-YES,WAIT-NO
```

Note that NO is the default for the WAIT parameter, so you could have entered the following:

```
// ALLOCATE UNIT-I1,AUTO-YES,CONTINUE-YES
```

and the results would be the same.

Directory to Using the System Support

This chapter lists tasks you may want to do using the System/36. The procedures, commands, and OCL statements that can be used to do each task are listed. The tasks are listed in the following groups:

- "Creating and Maintaining Disk Files" on page 1-8.
- "Creating and Maintaining Extended Character Files" on page 1-10.
- "Creating and Maintaining Libraries" on page 1-11.
- "Maintaining Folders and Folder Members" on page 1-13.
- "Processing Diskettes" on page 1-15.
- "Processing Tapes" on page 1-17.
- "Creating and Maintaining Display Formats, Menus, and Message Members" on page 1-18.
- "Creating and Maintaining Programs" on page 1-19.
- "Using the Office Products" on page 1-21.
- "Using the Personal Computer" on page 1-23.
- "Defining Data in Files" on page 1-25.
- "Running Programs and Procedures" on page 1-26.
- "Changing and Controlling Printers, Jobs, and Display Stations" on page 1-29.
- "Defining the System, Its Users, and Formatting the 9332" on page 1-32.
- "Communicating with Other Systems" on page 1-33.
- "Maintaining the System" on page 1-35.
- "Determining and Correcting Problems" on page 1-36.

When you are signed on the System/36, you can use the system help support to help you do a task. The system help support is made up of menus, prompt displays, and help text. The menus allow you to select a task you want to perform. When you select an item from a menu, either:

- Another menu is displayed (as you select options, each one gets more specific about the task you want to perform)
- A prompt display for a procedure or command is shown

A prompt display allows you to enter the necessary parameters and then run a procedure or command to do the task. The help text explains the menus, the menu options, the procedures and commands, and the parameters for the procedures and commands.

For more information about the system help support, see the "HELP Procedure" on page 4-199.

Creating and Maintaining Disk Files

Changing

To change the information in a disk file using the data file utility (DFU), see the "UPDATE Procedure" on page 4-550.

To use a file in a program, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

Compressing

To compress the disk space; that is, to collect all the free space on disk into an area, see the "COMPRESS Procedure" on page 4-105.

Copying

To copy a disk file and to do one or more of the following:

Create a new disk file with the same file organization
Create a new disk file with a different file organization
Change the record length of a file
Change the position and length of the keys in an indexed file
Include specific records in the new file
Omit specific records from the new file
Remove deleted records from a file

see the "COPYDATA Procedure" on page 4-111.

To add a disk file to an existing file on diskette, see the "SAVE Procedure" on page 4-416.

To create or copy a basic data exchange or I-exchange diskette file (to transfer a file to another system), see the "TRANSFER Procedure" on page 4-542.

To copy a disk file to a tape file (to transfer a file to another system) or a tape file to a disk file, see the "TAPECOPY Procedure" on page 4-500.

To copy a diskette file created by the IBM 5260 Retail System (special E-format) to a disk file, see the "POST Procedure" on page 4-345.

To copy the history file to a disk file, see the "HISTORY Procedure" on page 4-209.

To copy a spool file entry to a disk file, see the "COPYPRT Procedure" on page 4-126.

To copy work station utility (WSU) transaction file to a disk file and include or omit specific records, see the "WSUTXEX Procedure" on page 4-561.

Creating

To create an empty disk file, see the "BLDFILE Procedure" on page 4-56.

To create a disk file and enter records into the file using the data file utility (DFU), see the "ENTER Procedure" on page 4-172.

To create or use a file in a program, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

To create an alternative index for an existing disk file, see the "BLDINDEX Procedure" on page 4-59.

Organizing

See the "COPYDATA Procedure" on page 4-111.

Printing, Displaying

To print or display the contents of a disk file, see the "LISTDATA Procedure" on page 4-261 or the "LISTFILE Procedure" on page 4-267.

To print or display the names of all the files on a disk, diskette, tape, or tape cartridge, see the "CATALOG Procedure" on page 4-73.

To print or display a file saved on diskette, tape, or tape cartridge, see the "LISTDATA Procedure" on page 4-261 or the "LISTFILE Procedure" on page 4-267.

To print or display the history file, see the "HISTORY Procedure" on page 4-209.

To print or display the contents of a file using the data file utility (DFU), see the "LIST Procedure" on page 4-259.

To copy a spool file entry to a disk file, and then display that disk file, see the "COPYPRT Procedure" on page 4-126.

To print a graphics file on an intelligent printer data stream (IPDS) printer, see the "PRTGRAPH Procedure" on page 4-359.

Removing

To remove a file from disk or diskette, see the "DELETE Procedure" on page 4-144.

To copy a disk file and remove deleted records from the file, see the "COPYDATA Procedure" on page 4-111.

Renaming

To rename a disk file, see the "RENAME Procedure" on page 4-372.

Saving

To save one or more disk files on diskette, tape, or tape cartridge, see the "SAVE Procedure" on page 4-416.

To add a disk file to an existing file on diskette, see the "SAVE Procedure" on page 4-416.

Restoring

To restore one or more diskette, tape, or tape cartridge files to disk, see the "RESTORE Procedure" on page 4-392.

Securing

To secure a disk file to be read, changed, created, or removed only by certain operators, see the "SECDEF Procedure" on page 4-441 and the "SECEDIT Procedure" on page 4-443.

Sorting

To sort the contents of a disk file, see the "SORT Procedure" on page 4-483.

To sort the index keys of an indexed file, see the "KEYSORT Procedure" on page 4-252.

Creating and Maintaining Extended Character Files

Changing

To change the information in an extended character file, see the "CGU Procedure" on page 4-91.

Creating

To create the extended character file, see the "RESTEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-380.

To create ideographic characters for the extended character file, see the "CGU Procedure" on page 4-91.

Saving

To save the extended character file, see the "SAVEEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-425.

Restoring

To restore the extended character file, see the "RESTEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-380.

Sorting

To sort the extended character file, see the "SRTX Procedure" on page 4-484.

Creating and Maintaining Libraries

Changing

To change the size of a library or a library's directory, see the "ALOCLIBR Procedure" on page 4-9.

To change a library member using the source entry utility (SEU), see the "SEU Procedure" on page 4-465.

To change display formats or menus using the screen design aid utility (SDA), see the "SDA Procedure" on page 4-437.

To change the current library at your display station, see the "SLIB Procedure" on page 4-471.

To change the sign-on library for your display station, see the "SET Procedure" on page 4-454.

To change the name, subtype, or reference number of a library member, see the "CHNGEMEM Procedure" on page 4-94.

Condensing

To condense a library; that is, to collect all the free space in a library into one area, see the "CONDENSE Procedure" on page 4-109.

Copying

To copy one or more library members from one library to another library, see the "LIBRLIBR Procedure" on page 4-255.

To copy one or more library members to a disk, diskette, tape, or tape cartridge file, see the "FROMLIBR Procedure" on page 4-193.

To copy a library member to a basic data exchange diskette file to transfer the library member to another system, see the "Copy Members from a Library (FROMLIBR Procedure)" on page A-66.

To copy a disk, diskette, tape, or tape cartridge file containing one or more library members to a library, see the "TOLIBR Procedure" on page 4-537.

Creating

To create a new library, see the "BLDLIBR Procedure" on page 4-62.

To create a new library, and to copy members into the new library, see the "BLDLIBR Procedure" on page 4-62.

To create a new procedure or source member in a library using the source entry utility (SEU), see the "SEU Procedure" on page 4-465.

To create display formats or menus using the screen design aid utility (SDA), see the "SDA Procedure" on page 4-437.

Printing, Displaying

To print or display the names of the members in a library, or to list the contents of one or more library members, see the "LISTLIBR Procedure" on page 4-272.

To print or display the names of all the libraries on a disk, diskette, tape, or tape cartridge, see the "CATALOG Procedure" on page 4-73.

To print or display the names of the members in a library saved on diskette, tape, or tape cartridge, see the "LISTFILE Procedure" on page 4-267.

Removing

To remove a library from disk or diskette, see the "DELETE Procedure" on page 4-144.

To remove one or more members from a library, see the "REMOVE Procedure" on page 4-370.

To remove a library extent, see the "ALOCLIBR Procedure" on page 4-9 or the "CONDENSE Procedure" on page 4-109.

To remove a display format from a display format member, see the "FORMAT Procedure" on page 4-181 or the "SDA Procedure" on page 4-437.

Renaming

To rename a library, see the "RENAME Procedure" on page 4-372.

To rename a library member, see the "CHNGEMEM Procedure" on page 4-94.

Restoring

To restore a library from diskette, tape, or tape cartridge to disk, see the "RESTLIBR Procedure" on page 4-386.

Saving

To save a library on diskette, tape, or tape cartridge, see the "SAVELIBR Procedure" on page 4-431.

Securing

To secure a library to be read, changed, created, or removed only by certain operators, see the "SECDEF Procedure" on page 4-441 and the "SECEDIT Procedure" on page 4-443.

Maintaining Folders and Folder Members

Changing

To change the size of a folder, see the "ALOCFLDR Procedure" on page 4-8.

Condensing

To condense a folder; that is, to collect all the free space in a folder into one area, see the "CONDENSE Procedure" on page 4-109.

Converting

To convert all document folders and mail folders to a new internal format, see the "DOCCNV Procedure" on page 4-160.

Copying

To copy a folder onto disk, diskette, tape, or tape cartridge, see the "SAVEFLDR Procedure" on page 4-428.

To copy a folder member onto disk, diskette, tape, or tape cartridge, see the "ARCHIVE Procedure" on page 4-22.

To copy the shared folders facility to the PC Support/36 and system libraries from a backup diskette, see the "SHRFLOAD Procedure" on page 4-469.

Creating

To create or maintain a folder, see the "TEXTFLDR Procedure" on page 4-530.

To create a subdirectory, see the "DEFSUBD Procedure" on page 4-143.

To create or maintain a folder member, see the "TEXTDOC Procedure" on page 4-512.

Moving

To move a folder from one disk location to another, see the "MOVEFLDR Procedure" on page 4-298.

Printing, Displaying

To display a subdirectory screen, see the "DEFSUBD Procedure" on page 4-143.

To print or display the contents of a folder member, see the "LISTFILE Procedure" on page 4-267.

To print or display the names of all the folders on a disk, diskette, or tape cartridge, see the "CATALOG Procedure" on page 4-73.

Removing

To delete a subdirectory, see the "DEFSUBD Procedure" on page 4-143.

To remove a folder from disk, see the "DELETE Procedure" on page 4-144.

Renaming

To rename a folder, see the "RENAME Procedure" on page 4-372.

Reorganizing

To reorganize a folder, see the "ALOCFLDR Procedure" on page 4-8 and the "CONDENSE Procedure" on page 4-109.

Restoring

To restore a folder from disk, diskette, tape, or tape cartridge, see the "RESTFLDR Procedure" on page 4-383.

To restore a folder member from disk, diskette, tape, or tape cartridge, see the "RETRIEVE Procedure" on page 4-401.

Saving

To save a folder or all folders on disk, diskette, tape, or tape cartridge, see the "SAVEFLDR Procedure" on page 4-428.

To save a folder member on disk, diskette, tape, or tape cartridge, see the "ARCHIVE Procedure" on page 4-22.

To save the shared folders facility from PC Support/36 and system libraries on a backup diskette, see the "SHRFSAVE Procedure" on page 4-470.

Securing

To secure a folder to be read, changed, created, or removed only by certain operators, see the "SECDEF Procedure" on page 4-441 and the "SECEDIT Procedure" on page 4-443.

Processing Diskettes

Allocating

To allocate the diskette drive to a job, see the "ALLOCATE OCL Statement" on page 5-8.

To deallocate the diskette drive, see the "DEALLOC OCL Statement" on page 5-27.

Copying

To copy all or part of a diskette to another diskette or to free up contiguous space at the end of a diskette, see the "COPYI1 Procedure" on page 4-121.

To copy a diagnostic diskette, see the "COPYDIAG Procedure" on page 4-120.

To create or copy a basic data exchange or I-format diskette file, see the "TRANSFER Procedure" on page 4-542.

To copy a diskette created by the IBM 5260 Retail System (special E-format) to a disk file, see the "POST Procedure" on page 4-345.

To copy a diskette file containing one or more library members to a library, see the "TOLIBR Procedure" on page 4-537.

To copy one or more library members from a library to diskette, see the "FROMLIBR Procedure" on page 4-193.

Preparing

To prepare (also called *initialize*) a diskette before using it to save data, see the "INIT Procedure" on page 4-233.

To prepare a diagnostic diskette, see "INITDIAG Procedure" on page 4-236.

Printing, Displaying

To list the names of files, libraries, and folders contained on a diskette, or to list general information about a diskette, see the "CATALOG Procedure" on page 4-73.

To list a file saved on a diskette, see the "LISTDATA Procedure" on page 4-261or the "LISTFILE Procedure" on page 4-267.

To list a library saved on a diskette or to list a diskette exchange file, see the "LISTFILE Procedure" on page 4-267.

Removing

To remove one or more files or libraries from diskette, see the "DELETE Procedure" on page 4-144.

Restoring

To restore one or more files saved on diskette back to disk, see the "RESTORE Procedure" on page 4-392.

To restore a library from diskette to disk, see the "RESTLIBR Procedure" on page 4-386.

To restore a folder from diskette, see the "RESTFLDR Procedure" on page 4-383.

To restore a folder member from diskette, see the "RETRIEVE Procedure" on page 4-401.

Saving

To save one or more disk files on diskette, see the "SAVE Procedure" on page 4-416.

To save a library on diskette, see the "SAVELIBR Procedure" on page 4-431.

To save a folder or all folders on diskette, see the "SAVEFLDR Procedure" on page 4-428.

To save a folder member on diskette, see the "ARCHIVE Procedure" on page 4-22.

Processing Tapes

Allocating

To allocate the tape drive to a job, see the "ALLOCATE OCL Statement" on page 5-8.

To release the tape drive, see the "DEALLOC OCL Statement" on page 5-27.

Copying

To create or copy a tape exchange file, see the "TAPECOPY Procedure" on page 4-500.

To copy a tape file containing one or more library members to a library, see the "TOLIBR Procedure" on page 4-537. To copy one or more library members from a library to tape, see the "FROMLIBR Procedure" on page 4-193.

Preparing

To prepare (also called *initialize*) a tape or tape cartridge before using it to save data, see the "TAPEINIT Procedure" on page 4-507.

Printing, Displaying

To list the names of files, libraries, and folders contained on a tape or tape cartridge, or to list general information about a tape or tape cartridge, see the "CATALOG Procedure" on page 4-73.

To list a file saved on a tape or tape cartridge, see the "LISTDATA Procedure" on page 4-261 or "LISTFILE Procedure" on page 4-267.

To list a library saved on a tape or tape cartridge or to list a tape exchange file, see the "LISTFILE Procedure" on page 4-267.

Restoring

To restore one or more files saved on tape or tape cartridge back to disk, see the "RESTORE Procedure" on page 4-392.

To restore a folder from tape or tape cartridge, see the "RESTFLDR Procedure" on page 4-383.

To restore a folder member from tape or tape cartridge, see the "RETRIEVE Procedure" on page 4-401.

To restore a library from tape or tape cartridge to disk, see the "RESTLIBR Procedure" on page 4-386.

Saving

To save one or more disk files on tape or tape cartridge, see the "SAVE Procedure" on page 4-416.

To save a folder or all folders on tape or tape cartridge, see the "SAVEFLDR Procedure" on page 4-428.

To save a folder member to tape or tape cartridge, see the "ARCHIVE Procedure" on page 4-22.

To save a library on tape or tape cartridge, see the "SAVELIBR Procedure" on page 4-431.

Creating and Maintaining Display Formats, Menus, and Message Members

Display Formats

To create, change, and generate display formats using the screen design aid, see the "SDA Procedure" on page 4-437.

To only generate display formats from source statements, see the "FORMAT Procedure" on page 4-181.

To remove a display format from a display format load member, see the "SDA Procedure" on page 4-437 or the "FORMAT Procedure" on page 4-181.

Menus

To create or change a menu using the screen design aid (SDA), see the "SDA Procedure" on page 4-437.

To generate a menu from source statements, see the "BLDMENU Procedure" on page 4-66.

To display a menu, see the "MENU Control Command" on page 6-22.

Message Members

To generate a message member from source statements, see the "CREATE Procedure" on page 4-132.

To create or change the automatic response values for system or user message members, see the "RESPONSE Procedure" on page 4-374.

To create or change alert indicators for system or user message members, see the "SETALERT Procedure" on page 4-457.

To assign a message member to a procedure or a program, see the "MEMBER OCL Statement" on page 5-73.

Creating and Maintaining Programs

Assembler

To assemble an Assembler program, see the "ASM Procedure" on page 4-27.

To create or change an Assembler program, see the "SEU Procedure" on page 4-465.

BASIC

To create, change, run, and debug a BASIC program, see the "BASIC Procedure" on page 4-39.

To run a BASIC program, see the "BASICR Procedure" on page 4-41.

To run a BASIC procedure, see the "BASICP Procedure" on page 4-40.

To convert a BASIC source member to a BASIC subroutine member, see the "BASICS Procedure" on page 4-42.

BGU

To change fill patterns or color palettes to be used by BGU/36, see the "BGUATTR Procedure" on page 4-46.

To copy a graph data input file (GDIF) to a data member and store it in the user's library, see the "BGUDATA Procedure" on page 4-50.

To create, update, plot, print, or view a chart, see the "BGUCHART Procedure" on page 4-47.

To create, update, plot, print, or view a graph, see the "BGUGRAPH Procedure" on page 4-51.

RPG II

To display a menu of RPG II programming options, see the "RPGP Procedure" on page 4-411.

To create an RPG II program and then alternatively compile and correct errors in your source program online (at your display station), see the "RPGONL Procedure" on page 4-411.

To compile an RPG II source program, see the "RPGC Procedure" on page 4-407.

To compile an RPG II source program that contains auto report specifications, see the "AUTOC Procedure" on page 4-31.

To create or change a procedure or an RPG II or auto report source member, see the "RPGSEU Procedure" on page 4-414.

To create a cross reference listing of an RPG II program, see the "RPGX Procedure" on page 4-415.

To create or change display formats for an RPG program, see the "RPGSDA Procedure" on page 4-414.

To create display formats for an RPG II CONSOLE file, see the "RPGR Procedure" on page 4-412.

COBOL

To display a menu of COBOL programming options, see the "COBOLP Procedure" on page 4-102.

To create a COBOL program and then alternatively compile and correct errors in your source program online (at your display station), see the "COBOLONL Procedure" on page 4-102.

To compile a COBOL source program, see the "COBOLC Procedure" on page 4-99.

To create or change a procedure or a COBOL source member, see the "COBSEU Procedure" on page 4-104.

To create or change display formats for a COBOL program, see the "COBSDA Procedure" on page 4-103.

FORTRAN

To display a menu of FORTRAN programming options, see the "FORTP Procedure" on page 4-187.

To create a FORTRAN program and then alternatively compile and correct errors in your source program online (at your display station), see the "FORTONL Procedure" on page 4-187.

To compile a FORTRAN source program, see the "FORTRANC Procedure" on page 4-188.

To run a FORTRAN program, see the "FORTGO Procedure" on page 4-185.

To create or change a procedure or a FORTRAN source member, see the "FORTSEU Procedure" on page 4-192.

To create or change display formats for a FORTRAN program, see the "FORTSDA Procedure" on page 4-191.

Link-Editing

To link-edit one or more subroutine members into a program that can be run, see the "OLINK Procedure" on page 4-325.

WSU

To generate a work station utility (WSU) program from source specifications, see the "WSU Procedure" on page 4-554.

To create or change the source for a WSU program, see the "SEU Procedure" on page 4-465.

Using the Office Products

Personal Services/36

To change the default values used by Personal Services/36, see the "OFCDFLT Procedure" on page 4-313.

To control the activity and communications queues, see the "OFCQ Procedure" on page 4-322.

To convert the user profile data to Release 5 format, see the "OFCCONV Procedure" on page 4-311.

To copy PROFS bridge support onto diskette from the Personal Services/36 and system libraries, see the "PROFSAVE Procedure" on page 4-358.

To copy PROFS bridge support to the Personal Services/36 and system libraries, see the "PROFLOAD Procedure" on page 4-357.

To copy the document library services support from diskette into the Personal Services/36 and system libraries, see the "DLSLOAD Procedure" on page 4-159.

To copy the document library services support from the Personal Services/36 and system libraries to diskette, see the "DLSSAVE Procedure" on page 4-159.

To create or maintain a calendar, see the "OFCCAL Procedure" on page 4-309.

To enroll or change the enrollment of Personal Services/36 users, see the "OFCUSER Procedure" on page 4-324.

To file electronic document or log receipt of a hardcopy document, see the "OFCFILE Procedure" on page 4-315.

To install Personal Services/36 files, see the "OFCINSTL Procedure" on page 4-316.

To maintain library descriptions used for libraries, see the "OFCLDF Procedure" on page 4-316.

To maintain office information, see the "OFCMAINT Procedure" on page 4-320.

To maintain Personal Services/36 or System/36 communications definitions, see the "OFCCOMM Procedure" on page 4-310.

To reorganize or save office information, see the "OFCDATA Procedure" on page 4-312.

To request batch printing and deleting of calendar items, see the "OFCBPRT Procedure" on page 4-307.

To select different ways of looking at the directory, see the "OFCDIR Procedure" on page 4-314.

To send messages to a group, see the "OFCMSG Procedure" on page 4-321.

To stop the Personal Services/36 background mail tasks, see the "OFCCANCL Procedure" on page 4-310.

To view a list of the library requests submitted, see the "OFCSTAT Procedure" on page 4-323.

To work with mail, see the "OFCMAIL Procedure" on page 4-318.

To work with a user group, see the "OFCGRP Procedure" on page 4-315.

To add or update data on a file, see the "OFCSRCH Procedure" on page 4-323.

Query/36

To add or update data on a file, see the "QRYDE Procedure" on page 4-361.

To define a query or work with already existing queries, see the "QRY Procedure" on page 4-360.

To run a query and display, print, or send to disk the data produced, see the "QRYRUN Procedure" on page 4-363.

DisplayWrite/36 (DW/36)

To convert a document created by the Text Management System (TMS) to DW/36, see the "TEXTCONV Procedure" on page 4-511.

To create or maintain a folder, see the "TEXTFLDR Procedure" on page 4-530.

To create or maintain a folder member, see the "TEXTDOC Procedure" on page 4-512.

To create or maintain user profiles, see the "TEXTPROF Procedure" on page 4-532.

To display an online document in its final form, see the "READINFO Procedure" on page 4-368.

To maintain a document object in a folder, see the "TEXTOBJ Procedure" on page 4-532.

To maintain a supplemental dictionary, see the "TEXTDCT Procedure" on page 4-511.

To perform various print tasks, see the "TEXTPRTQ Procedure" on page 4-533.

To release documents that have been held for later printing, see the "TEXTREL Procedure" on page 4-534.

Print Online Support

To copy the print online support to diskette, see the "DOCPSAVE Procedure" on page 4-162.

To create a library named #TULIB and copy the print online support from diskette into that library, see the "DOCPLOAD Procedure" on page 4-161.

Using the Personal Computer

PC Support/36

To copy the IBM Token-Ring Network onto diskette from the PC Support/36 and system libraries, see the "LRTRSAVE Procedure" on page 4-296.

To copy the IBM Token-Ring Network to the PC Support/36 and system libraries, see the "LRTRLOAD Procedure" on page 4-295.

To copy the PC portion of the 3278 emulation via the IBM Personal Computer from the System/36 to the IBM Personal Computer, see the "EPDOWNL Procedure" on page 4-174.

To copy the PC Support/36 from the system to diskette, see "IWSAVE Procedure" on page 4-245.

To copy the PC Support/36 to the system, see the "IWLOAD Procedure" on page 4-242.

To copy the PC Support/36 Organizer from backup diskette to PC Support/36 and system libraries, see the "PCOLOAD Procedure" on page 4-335.

To copy PC Support/36 pass-through support onto diskette from the PC Support/36 and system libraries, see the "IWPTSAVE Procedure" on page 4-244.

To copy PC Support/36 pass-through support to the PC Support/36 and system libraries, see the "IWPTLOAD Procedure" on page 4-243.

To copy the PC Support/36 work station feature onto diskette from the PC Support/36 and system libraries, see the "WSFSAVE Procedure" on page 4-553.

To copy the PC Support/36 work station feature to the PC Support/36 and system libraries, see the "WSFLOAD Procedure" on page 4-552.

To copy software distribution support onto diskette from the software distribution and system libraries, see the "SWDSAVE Procedure" on page 4-495.

To copy software distribution support to the software distribution and system libraries, see the "SWDLOAD Procedure" on page 4-494.

To exchange data between a virtual disk or diskette and a folder, see the "PCEXCH Procedure" on page 4-332.

To issue commands on the personal computer, see the "PCEXEC Procedure" on page 4-334.

To merge the personal computer machine-readable instruction files and translated tables from the system library into a virtual diskette, see the "EPLMRG Procedure" on page 4-174.

To modify tables used to translate characters, see the "CHGXLATE Procedure" on page 4-93.

To save the PC Support/36 Organizer from PC Support/36 and system libraries on backup diskette, see the "PCOSAVE Procedure" on page 4-337.

To select a text editor, see the "PCOPROF Procedure" on page 4-336.

To send or receive messages by way of a personal computer, see the "MSG OCL Statement" on page 5-76 or the "MSG Control Command" on page 6-24.

To start the PC Utility, see the "PCU Procedure" on page 4-338.

Defining Data in Files

Interactive Data Definition Utility (IDDU)

To link or unlink a disk file, or enter or update data in a disk file, or create a disk file, see the "IDDUDISK Procedure" on page 4-223.

To create or maintain a data definition, see the "IDDUDFN Procedure" on page 4-223.

To create or maintain a data dictionary, see the "IDDUDCT Procedure" on page 4-222.

To link or unlink a file definition with a disk file, see the "IDDULINK Procedure" on page 4-224.

To print field, format, or file definitions in a data dictionary, see the "IDDUPRT Procedure" on page 4-226.

To translate the RPG source specifications contained in RPG program source members or in Text Management System (TMS) data definitions into IDDU definitions, see the "IDDUXLAT Procedure" on page 4-227.

Running Programs and Procedures

Date

To change the program, job, or job step date, see the "DATE OCL Statement" on page 5-25 or the "DATE Procedure" on page 4-137.

Data Files

To use a data file in a program, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

To reserve disk space for scratch and job data files, see the "RESERVE OCL Statement" on page 5-104.

Display Formats

To show a display format from a procedure to prompt for parameters or data, see the "PROMPT OCL Statement" on page 5-97.

Display Stations

To assign a display station to a program, see the "WORKSTN OCL Statement" on page 5-120.

Local Data Area

To change information in the local data area, see the "LOCAL OCL Statement" on page 5-70.

To substitute information from the local data area into a procedure, see "?L'position,length'? (Local Data Area) Expression" on page 3-19.

Menus

To display and use a menu, see the "MENU Control Command" on page 6-22 or the "MENU OCL Statement" on page 5-75.

Messages

To assign a message member to a program or procedure, see the "MEMBER OCL Statement" on page 5-73.

To send a message to an operator, see:

The "MSG Control Command" on page 6-24
The "// * (Informational Message) Statement" on page 3-57
The "PAUSE Statement" on page 3-69
The "ERR Procedure" on page 4-176
The "// ** (System Console Message) Statement" on page 3-59
The "MSG OCL Statement" on page 5-76

To specify whether informational messages are to be displayed, see the "INFOMSG OCL Statement" on page 5-65.

To change the automatic response level for a procedure or program, see the "NOHALT OCL Statement" on page 5-79.

To define, list, and remove messages from the message file, see the "MSGFILE Procedure" on page 4-299.

Printers

To use a printer in a program, see the "PRINTER OCL Statement" on page 5-83.

To change one or more of the following, see the "PRINT Procedure" on page 4-350, the "SET Procedure" on page 4-454, or the "FORMS OCL Statement" on page 5-55:

- The printer to be used
- The number of lines per page
- The number of characters per inch (CPI) printed horizontally
- The number of lines per inch (LPI) printed vertically
- The forms number to be used
- The orientation or size of the output on the page
- The printer drawer from which paper is used

To change the current system list device, see the "SYSLIST Procedure" on page 4-498 or the "SYSLIST OCL Statement" on page 5-113.

To change the printer used for Print key output, and to specify whether a border or heading is to be printed with the Print key output, see the "PRINTKEY Procedure" on page 4-354, the "SET Procedure" on page 4-454, or the "WORKSTN OCL Statement" on page 5-120.

Priority

To change the processing priority of a job, see the "PRTY Control Command" on page 6-29 or the "ATTR OCL Statement" on page 5-11.

Program Size

To specify how much storage a program needs to run, see the "REGION OCL Statement" on page 5-103.

Restarting

To allow jobs to run on the system after a STOP command was entered, see the "START Control Command" on page 6-36.

To release one or more held jobs on the job queue so they can be run, see the "RELEASE Control Command" on page 6-30.

Starting

To display the current status of running programs and procedures, use the STATUS USERS or STATUSF USERS control command.

To load and run a compiled program, see the "LOAD OCL Statement" on page 5-69 and the "RUN OCL Statement" on page 5-105.

To start a procedure, see:

- The "INCLUDE OCL Statement" on page 5-63
- The "JOBQ Control Command" on page 6-21
- The "JOBQ OCL Statement" on page 5-66
- The "EVOKE OCL Statement" on page 5-30

To start a job from a job stream, see the "JOBSTR Procedure" on page 4-246.

Stopping

To stop a currently running job, see the "CANCEL Control Command" on page 6-5.

To prevent any jobs from running, see the "STOP Control Command" on page 6-48.

To hold one or more jobs on the job queue, see the "HOLD Control Command" on page 6-18.

Switches

Switches are also called external indicators U1 through U8.

To change the switch settings, see the "SWITCH OCL Statement" on page 5-112.

To check the switch settings in a procedure, see "Substitution Expressions" on page 3-10.

Waiting

To wait a specific amount of time or to wait until a certain time occurs before a job begins, see the "WAIT OCL Statement" on page 5-118.

Changing and Controlling Printers, Jobs, and Display Stations

To display information about your display station environment, see the "STATUS Control Command" on page 6-40 or the "STATUSF Control Command" on page 6-45.

Displayed Data

To assign the direction of the roll keys, see the "ROLLKEYS Procedure" on page 4-406.

To display the current system list device, use the STATUS SESSION control command.

To change the current system list device, see the "SYSLIST Procedure" on page 4-498 or the "SYSLIST OCL Statement" on page 5-113.

Job Queue

To display the status of the job queue, use the STATUS JOBQ or STATUSF JOBQ control command.

To place a job on the job queue, see the "JOBQ Control Command" on page 6-21 or the "JOBQ OCL Statement" on page 5-66.

To change the number of jobs allowed to run from the job queue, see the "CHANGE Control Command" on page 6-9.

To change the position of a job on the job queue, see the "CHANGE Control Command" on page 6-9.

To prevent one or more jobs on the job queue from running, see the "HOLD Control Command" on page 6-18.

To allow one or more held jobs on the job queue to run, see the "RELEASE Control Command" on page 6-30.

To stop the job queue or a specific job queue priority, see the "STOP Control Command" on page 6-48.

To start the job queue or a specific job queue priority, see the "START Control Command" on page 6-36.

Jobs

To display the status of the programs and procedures running on the system, use the STATUS USERS or STATUSF USERS control command.

To start a job, see:

The "INCLUDE OCL Statement" on page 5-63
The "JOBQ Control Command" on page 6-21
The "JOBQ OCL Statement" on page 5-66
The "EVOKE OCL Statement" on page 5-30

To cancel a job, see the "CANCEL Control Command" on page 6-5.

To prevent jobs from running on the system, see the "STOP Control Command" on page 6-48.

To allow jobs to run on the system, see the "START Control Command" on page 6-36.

To change the processing priority of a job running on the system or a job on the job queue, see the "PRTY Control Command" on page 6-29.

Libraries

To display your current or session library, use the STATUS SESSION control command.

To change the current or session library, see the "SLIB Procedure" on page 4-471 or the "LIBRARY OCL Statement" on page 5-67.

Menus

To display a user menu, see the "MENU Control Command" on page 6-22 or the "MENU OCL Statement" on page 5-75.

To display a help menu, see the "HELP Procedure" on page 4-199.

Messages

To specify whether informational messages are to be displayed, see the "INFOMSG Control Command" on page 6-20.

To change the automatic response level for your display station, see the "NOHALT Procedure" on page 4-305.

To send a message to another display station, or to display a message sent from another display station, see the "MSG Control Command" on page 6-24.

To reply to a message displayed at the system console or a subconsole, see the "REPLY Control Command" on page 6-32.

Powering Off

To power off the system, see the "POWER Control Command" on page 6-28 or the "POWER OCL Statement" on page 5-82.

Printed Data

To display the status of printed output on the spool file, use the STATUS PRT or STATUSF PRT control command.

To display the status of the spool writers, use the STATUS WRT control command.

To change one or more of the following, see the "PRINT Procedure" on page 4-350, the "SET Procedure" on page 4-454, or the "FORMS OCL Statement" on page 5-55:

- The printer to be used
- The number of lines per page
- The number of characters per inch (CPI) printed horizontally
- The number of lines per inch (LPI) printed vertically
- The forms number to be used
- The orientation or size of the output on the page
- The printer drawer from which paper is used

To change the current system list device, see the "SYSLIST Procedure" on page 4-498.

To change your sign on or current printer, see the "SET Procedure" on page 4-454.

To change the printer used when you press the Print key and to specify whether a border or heading is to be printed, see the "PRINTKEY Procedure" on page 4-354.

To start the printing of spooled output, see the "START Control Command" on page 6-36 or the "START OCL Statement" on page 5-109.

To start, stop, or display information about printer load balancing, see the "BALPRINT Procedure" on page 4-34.

To stop the printing of spooled output, see the "STOP Control Command" on page 6-48 or the "STOP OCL Statement" on page 5-111.

To restart the printing of spooled output, see the "RESTART Control Command" on page 6-34.

To hold spooled output on the spool file to prevent it from printing, see the "HOLD Control Command" on page 6-18.

To release held spooled output for printing, see the "RELEASE Control Command" on page 6-30.

To change one or more of the following:

The position of spool file entries
The number of copies to be printed
The forms number to be used
The printer to be used
The number of separator pages to be printed
Whether spool file entries should be printed before they are closed
The priority of the spool writer

see the "CHANGE Control Command" on page 6-9 or the "CHANGE OCL Statement" on page 5-18.

To cancel one or more entries from the spool file, see the "CANCEL Control Command" on page 6-5 or the "CANCEL OCL Statement" on page 5-16.

To change the print belt images for the 3262 printer, see the "SET Procedure" on page 4-454 or the "IMAGE OCL Statement" on page 5-59.

Signing Off

To sign your display station off the system, see the "OFF Control Command" on page 6-27 or the "OFF OCL Statement" on page 5-81.

To sign another display station off the system, see the "CANCEL Control Command" on page 6-5.

System Console

To transfer the system console function to another display station, see the "CONSOLE Control Command" on page 6-15.

To change the work station IDs of one or more printers or display stations, see the "ASSIGN Control Command" on page 6-2.

System Service Display Station

To allow a display station to be used to service the system, see the "START Control Command" on page 6-36.

To stop a display station from being used to service the system, see the "STOP Control Command" on page 6-48.

Defining the System, Its Users, and Formatting the 9332

Defining

To configure the system, see the "CNFIGSSP Procedure" on page 4-97.

To change the configuration of the system, see the "CNFIGSSP Procedure" on page 4-97.

To create the user ID file, or to create the resource security file, see the "SECDEF Procedure" on page 4-441.

To enter or change data in the user ID file or the resource security file, see the "SECEDIT Procedure" on page 4-443.

To initialize and format the 9332 Disk Unit, see the "INIT9332 Procedure" on page 4-237.

Printing, Displaying

To list the configuration of the system, see the "CNFIGSSP Procedure" on page 4-97.

To list the user identification file or the resource security file, see the "SECLIST Procedure" on page 4-445.

Saving

To save on tape the user identification file or the resource security file that is on diskette, tape, or tape cartridge on tape, or to copy them to disk files, see the "SECSAVE Procedure" on page 4-451.

Restoring

To restore the user identification file or the resource security file from diskette, tape, or tape cartridge, or from disk files, see the "SECREST Procedure" on page 4-448.

Starting Up

To have procedures run automatically, immediately after starting up the system; that is, part of initial program load (IPL), see the "#STRTUP1 Procedure" on page 4-4 and the "#STRTUP2 Procedure" on page 4-6.

To initiate an IPL through program control, see the "IPL Procedure" on page 4-240.

Communicating with Other Systems

Communicating

To copy LAN communications support onto diskette from the LAN and system libraries, see the "LANSAVE Procedure" on page 4-254.

To copy LAN communications support to the LAN and system libraries, see the "LANLOAD Procedure" on page 4-253.

To pass through from your system to a remote System/36 or System/38 where you can sign on as if your display station were attached to the remote System/36 or System/38, see the "PASSTHRU Procedure" on page 4-330.

To start an SSP-ICF communications subsystem, MSRJE, or 3270 device emulation, see the "ENABLE Procedure" on page 4-170.

To stop an SSP-ICF communications subsystem, MSRJE, or 3270 device emulation, see the "DISABLE Procedure" on page 4-157.

To place a communications line or remote work station online or offline, see the "VARY Control Command" on page 6-53.

To start the automonitor function for BSC multipoint communications lines, see the "STARTM Procedure" on page 4-489.

To stop the automonitor function for BSC multipoint communications lines, see the "STOPM Procedure" on page 4-492.

To start an APPC session group, see the "STRTGRP Procedure" on page 4-493.

To stop an APPC session group, see the "STOPGRP Procedure" on page 4-491.

To start, stop, or change error reporting in an IBM Token-Ring Network, see the "TRNMGR Procedure" on page 4-549.

Defining

To set up or configure the Interactive Communications Feature (SSP-ICF) subsystems, MSRJE, or 3270 device emulation, see the "CNFIGICF Procedure" on page 4-96.

To set up a controller for the SSP-ICF Finance subsystem, see the "LOAD3601 Procedure" on page 4-293.

To create or change the list of remote IDs for a switched communications line that use the SSP-ICF BSCEL subsystem, see the "DEFINEID Procedure" on page 4-139.

To set communications configuration items see the "SETCOMM Procedure" on page 4-461.

To create or change a phone list for the autocall feature, see the "DEFINEPN Procedure" on page 4-140.

To create or change a list of numbers for the X.21 feature or a short-hold mode line configuration, see the "DEFINX21 Procedure" on page 4-141.

To change the configuration of a communications line, see:

The "ALTERCOM Procedure" on page 4-11
The "SETCOMM Procedure" on page 4-461
The "COMM OCL Statement" on page 5-21
The "SESSION OCL Statement" on page 5-106

To specify which system messages should generate alert messages, see the "ALERT Procedure" on page 4-7.

To register or cancel an available user facility on an X.21 public data network, see the "REQUESTX Procedure" on page 4-373.

Remote Job Entry

To do remote job entry, see the "MSRJE Procedure" on page 4-302.

To print the information created by a remote job entry task, see the "RJFILE Procedure" on page 4-405.

To define a remote job entry control table, see the "RJTABLE Procedure" on page 4-405. down 8

3270 Emulation

To run SNA 3270 device emulation, see the "ES3270 Procedure" on page 4-178.

To run BSC 3270 device emulation, see the "EM3270 Procedure" on page 4-168.

To sign an IBM personal computer on to SNA 3270 device emulation, see the "EP3270 Procedure" on page 4-175.

Network Resource Directory

To create or edit the network resource directory, see the "EDITNRD Procedure" on page 4-167.

To list entries from the network resource directory, see the "LISTNRD Procedure" on page 4-292.

To list the contents of the network resource directory, see the "CATALOG Procedure" on page 4-73.

To remove the network resource directory from disk, see the "DELNRD Procedure" on page 4-150.

To restore the network resource directory from diskette, tape, or tape cartridge onto disk, see the "RESTNRD Procedure" on page 4-389.

To save the network resource directory on diskette, tape, or tape cartridge, see the "SAVENRD Procedure" on page 4-434.

Maintaining the System

Disk Space

To list the contents of the disk and the areas of unused space on the disk, see the "CATALOG Procedure" on page 4-73.

To collect the one or more unused spaces between disk files and libraries, thus making room for more disk files and libraries, see the "COMPRESS Procedure" on page 4-105.

To collect all the unused space in a library into one area, thus making room for more library members, see the "CONDENSE Procedure" on page 4-109.

To create, change or delete a buffer used to keep disk data in main storage, see the "CACHE Procedure" on page 4-72.

History

To print or display history file entries, see the "HISTORY Procedure" on page 4-209.

To copy the history file to a disk file, see the "HISTORY Procedure" on page 4-209.

To erase information from the history file, see the "HISTORY Procedure" on page 4-209.

To run a procedure when the SSP automatically creates a disk file named HISTCOPY (because the history file is full), see the "HISTCOPY Procedure" on page 4-207.

System Measurement

To measure the activity of your system, see the "SMF Procedure" on page 4-473.

To measure the activity of your communications lines, see the "STARTM Procedure" on page 4-489 and the "STOPM Procedure" on page 4-492.

Determining and Correcting Problems

Correcting

To start the online problem determination procedures, see the "PROBLEM Procedure" on page 4-356. Also, see the manual *System Problem Determination* for information about what you can do before calling IBM for service.

Debugging

To cause each operation control language (OCL) statement processed in a procedure to be logged to the history file, see the "LOG OCL Statement" on page 5-72.

To cause all evaluations of the procedure control expressions to be listed as the procedure is run, see the "DEBUG OCL Statement" on page 5-28.

To display or print the history file, or to copy the history file to a disk file, see the "HISTORY Procedure" on page 4-209.

Files

To recover a file that received a permanent disk error, see the "BUILD Procedure" on page 4-70.

Loading

To load the online problem determination file from diskette to disk, see the "OLPDLOAD Procedure" on page 4-328.

Network Problem Determination

To display or print the session trace information, the directory of locations, or the network configuration information, see the "APPNINFO Procedure" on page 4-19.

Service Aids

To run the service aid procedures and programs, see the "SERVICE Procedure" on page C-29.

System Service Display Station

To allow a display station to be used to service the system, see the "START Control Command" on page 6-36.

To stop a display station from being used to service the system, see the "STOP Control Command" on page 6-48.

Determining and Correcting Problems

Correcting

To start the online problem determination procedures, see the "PROBLEM Procedure" on page 4-356. Also, see the manual *System Problem Determination* for information about what you can do before calling IBM for service.

Debugging

To cause each operation control language (OCL) statement processed in a procedure to be logged to the history file, see the "LOG OCL Statement" on page 5-72.

To cause all evaluations of the procedure control expressions to be listed as the procedure is run, see the "DEBUG OCL Statement" on page 5-28.

To display or print the history file, or to copy the history file to a disk file, see the "HISTORY Procedure" on page 4-209.

Files

To recover a file that received a permanent disk error, see the "BUILD Procedure" on page 4-70.

Loading

To load the online problem determination file from diskette to disk, see the "OLPDLOAD Procedure" on page 4-328.

Network Problem Determination

To display or print the session trace information, the directory of locations, or the network configuration information, see the "APPNINFO Procedure" on page 4-19.

Service Aids

To run the service aid procedures and programs, see the "SERVICE Procedure" on page C-29.

System Service Display Station

To allow a display station to be used to service the system, see the "START Control Command" on page 6-36.

To stop a display station from being used to service the system, see the "STOP Control Command" on page 6-48.

Chapter 2. Making Your Own Procedures

This chapter describes how you can write and use your own procedures.

What a Procedure Is

A **procedure** is a collection of statements that causes one or more programs to be run. Procedures make it possible to avoid entering several statements each time a job must be performed. The collection of statements is stored in a library member called a procedure member.

The purpose of a procedure is to do a job. The procedure can do this job by having one or more job steps. A **job step** is a unit of work done by one program. A job step usually begins with the LOAD OCL statement, and usually ends with the RUN OCL statement (some SSP utility program job steps end with an END utility control statement). The following procedure contains one job step because only one program is loaded and run:

```
// LOAD PROG1
// RUN
```

This next example has two job steps because two programs are loaded and run:

```
// LOAD PROG1
// RUN
// LOAD PROG2
// RUN
```

What a Procedure Can Contain

The statements in a procedure control the files, display stations, and printers used by a program. For example:

```
// LOAD PROG3
// FILE NAME-CUSTOMER
// RUN
```

These statements indicate the following:

LOAD The program to be run is named PROG3.

FILE A disk file named CUSTOMER is to be used by program PROG3.

RUN Indicates that the program is to be run.

Procedures can be made up of the following types of statements:

- **OCL statements**, which are used to load and run programs. OCL means **operation control language**. OCL statements also indicate how the SSP is to run the program and how the SSP is to use the input and output devices that the program may require. See Chapter 5, "OCL Statements" for more information about the OCL statements that are used in this chapter. Examples of OCL statements are LOAD, FILE, and RUN.

- **Procedure control expressions**, which control how the procedure is processed based upon certain conditions. See Chapter 3, "Procedure Control Expressions" for more information about the procedure control expressions that are used in this chapter.

- **Procedure commands**, which cause other procedures to be run. Procedures supplied by IBM as part of the SSP and as part of the other licensed programs are described in Chapter 4, "Procedures." Examples of these procedures are COPYDATA and BASIC. You can also use your own procedures.

- **Utility control statements** for SSP utility programs, which pass information to SSP utility programs. The SSP utility control statements are shown in Appendix A, "SSP Utility Programs."

Procedures cannot contain any control commands. See Chapter 6, "Control Commands" for descriptions of the control commands.

Entering Procedures Into the System

You enter procedures into a library using the source entry utility (SEU) or development support utility (DSU). SEU is described in detail in the *SEU Guide*. The SEU procedure is described in both that manual and in this manual; see the "SEU Procedure" on page 4-465 of this manual. DSU is described in detail in the *Development Support Utility Guide*. The DSU procedure is described in that manual and in this manual; see the "DSU Procedure" on page 4-163 of this manual.

You can also use the $MAINT utility program to enter procedures into a library; see "$MAINT Utility" on page A-55 for information.

Naming Procedures

A procedure name can consist of any combination of 1 to 8 characters where the first character must be alphabetic (A through Z, #, $, or @).

Care should be taken when you give a user-written procedure the same name as a system procedure. The system will automatically run the procedure that exists in the current user library if a library parameter is not used in the procedure statement. If a library parameter is used in the procedure statement, the procedure is run from the library specified by the library parameter. Refer to the "INCLUDE OCL Statement" on page 5-63 for more information on the library parameter for a procedure statement.

When the HELP procedure is used to specify parameters for a system procedure, and a user procedure with the same name exists in the current user library, the system procedure help display will be shown, but the user procedure will be run. This results in the system procedure parameters being used by the user procedure. Unless the parameters match, the user procedure will fail to run properly.

Procedure Parameters

You can define parameters for your procedures. Parameters allow information and variables to be passed to the procedure. Parameters can have up to 128 characters.

A procedure can have a maximum of 64 parameters. Parameters passed to procedures are called **positional parameters**. Whenever a parameter appears in a procedure command, it must appear in the same position in relation to other parameters in the procedure command. That is, each parameter is assigned a place, such as the first parameter or the second parameter. If a parameter is omitted, a comma must still be used to indicate the *position* of the omitted parameter.

To use parameters in your own procedures, you use substitution expressions. The following substitution expressions can be used for procedure parameters.

Expression	Meaning
?n?	This expression substitutes the value of the nth positional parameter into the statement in the procedure. For example, the value entered on the procedure command in the first parameter position would be substituted by the expression ?1?.
?n'value'?	If the nth parameter is not entered (that is, not specified on the procedure command that started the procedure), this expression defines the value of the nth parameter and substitutes the value into the statement in the procedure. Once a parameter is set to 'value', it remains at that value as if it had been entered. It is not a temporary substitution. If the nth parameter is entered, this expression substitutes the value of the nth parameter into the statement in the procedure.
?nT'value'?	If the nth parameter is not entered, this expression temporarily substitutes the value specified into the statement in the procedure. The temporary value is used only for the current substitution. If the nth parameter is entered, this expression substitutes the value of the nth parameter into the statement in the procedure.
?nF'value'?	This expression changes the value of the nth parameter and substitutes that new value into the statement in the procedure. Once a parameter is set to 'value', it remains at that value as if it had been entered. It is not a temporary substitution.
?nR?	If the nth parameter is not entered, this expression displays a message to enter the required parameter and allows the operator to enter the parameter. For parameters that are prompted for by the system, the maximum number of characters that can be entered is 60. It then gives the nth positional parameter a value and substitutes that value into the statement in the procedure. If the nth parameter has a value, no message is displayed and the value of the nth positional parameter is substituted into the statement in the procedure.

Expression	Meaning
?nR'mic'?	If the nth parameter is not entered, this expression displays a message you can specify and allows the operator to enter the parameter. For parameters that are prompted for by the system, the maximum number of characters that can be entered is 60. It then defines the nth positional parameter and substitutes the value entered into the statement in the procedure. See the "CREATE Procedure" on page 4-132 for information about how to create these messages. If the nth parameter is entered, no message is displayed and the value of the nth positional parameter is substituted into the statement in the procedure.

The resolved values will be placed in the statement starting at the position of the leftmost question mark (?).

Example

You have created a program that reads and prints the information contained in a file. You can specify the name of the file as a parameter.

You create a procedure named PROC1 to run your program (the program is named PRTFILE). You can use SEU, DSU, or the $MAINT utility to enter this procedure into the system. Parameter 1 is the name of the disk file to be used. You can create the procedure PROC1 as follows:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-?1?
// RUN
```

You would enter the PROC1 procedure into a library procedure member named PROC1. The statements in the PROC1 procedure indicate the following:

LOAD The program to be run is named PRTFILE.

FILE A disk file (UNIT-F1) is to be used by the program. The program refers to the file as INPUT. The name of the file to be used is contained in parameter 1 (indicated by LABEL-?1?).

RUN Indicates that the program is to start running.

When you enter the following command to run the PROC1 procedure:

```
PROC1 FILEA
```

the statements that would really be run are:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-FILEA
// RUN
```

Note that FILEA was substituted in place of the ?1? expression.

Procedure Parameter Defaults

You can define defaults for parameters. A **default** is a value that is automatically substituted for an omitted or undefined parameter. You indicate defaults for parameters using the ?n'value'? substitution expression.

Example

You want to change PROC1 so that the default for the file name is EMPLOYES if no parameter is entered. You could create a procedure named PROC2 (based upon PROC1) as follows:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-?1'EMPLOYES'?
// RUN
```

You would enter the PROC2 procedure into a library procedure member named PROC2 using SEU or DSU. The statements in the PROC2 procedure indicate the following:

LOAD Same as in PROC1.

FILE The program refers to the disk file as INPUT. The name of the file to be used is contained in parameter 1. If no value was entered for the first parameter when the procedure was started, a value of EMPLOYES is assumed for the LABEL parameter; that is, EMPLOYES is the default for the first parameter (indicated by LABEL-?1'EMPLOYES'?).

RUN Same as PROC1.

When you enter the following command to run the PROC2 procedure:

```
PROC2
```

the statements that would really be run are:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-EMPLOYES
// RUN
```

Note that because no value was specified for the first parameter, the value EMPLOYES was assumed.

But when you enter the following command to run the PROC2 procedure:

```
PROC2 FILEA
```

the statements that would really be run are:

```
// LOAD PRTFILE
// FILE NAME-INPUT,UNIT-F1,LABEL-FILEA
// RUN
```

Note that because a value was specified for the first parameter, that value (FILEA) was used.

Testing Entered Parameters

You can use the IF conditional expressions to check the parameters entered for a procedure. This allows you to ensure that the parameters are correct before you run your program.

The IF expression has two forms:

- IF or IFT test for true conditions
- IFF tests for false conditions

Example

The following procedure requires that the first parameter (the name of the file to be processed) be either EMPLOY or CUSTOM. If the first parameter is not one of these values, only the PAUSE and CANCEL statements are processed (which cause an error message to be displayed and the procedure to be canceled).

```
// IF ?1?=EMPLOY GOTO OK
// IF ?1?=CUSTOM GOTO OK
// PAUSE 'Parameter 1 must be EMPLOY or CUSTOM'
// CANCEL
// TAG OK
//     LOAD    PRTFILE
//     FILE    NAME-INPUT,UNIT-F1,LABEL-?1?
//     PRINTER NAME-OUTPUT,DEVICE-?2'P3'?
//     RUN
```

Parameter Coding Considerations

The blank, comma (,), apostrophe ('), question mark (?), slash (/), equal sign (=), plus sign (+), greater than sign (>), and hyphen (-) have special meanings in procedures, in OCL statements, and in utility control statements. These characters can be used in parameters for a procedure but with caution.

Continuing the Lines of a Procedure

The maximum line length of a library procedure member or of a statement you can enter from the keyboard is 120 characters. If a statement you need to use contains more than 120 characters, you can use the continuation symbol to continue the statement on one or more lines. The continuation symbol is a plus sign (+) in the position where the incomplete line ends. You can use the plus sign to continue statements in procedures and to continue lines you are entering at the keyboard. The plus sign works as a continuation character when no characters besides blanks appear to the right of the plus sign.

The continuation symbol causes the next line (starting with the first nonblank character) to be concatenated to (or appended to) the previous line. You can continue as many lines as you wish, but the total number of characters in the resulting concatenated statement cannot exceed 512.

For example, a procedure contains the following three lines:

```
// IFF ?1?=EMPLOY +
   IFF ?1?=MANAG  +
   PROCA TEST,RUN
```

The resulting statement would be:

```
// IFF ?1?=EMPLOY IFF ?1?=MANAG  PROCA TEST,RUN
```

Note that the blanks after the + sign and the blanks before each continued line are ignored. The blanks before the + sign are preserved. Also notice that no comment data was placed after the plus signs; had there been comments on these lines, the plus sign would have been ignored. Comment data can be placed on the last line only because it is not continued.

The continuation symbol can be specified anywhere within the line, for example, within a parameter, substitution expression, or any other type of data. For example, a procedure could contain the following two lines:

```
// IFF ?1?=EMPLOY  I+
   FF ?1?=MANAG   PROCA TEST,RUN
```

The resulting statement would be:

```
// IFF ?1?=EMPLOY  IFF ?1?=MANAG   PROCA TEST,RUN
```

The continuation symbol provides continuation in addition to the continuation allowed in OCL statements and utility control expressions.

You can use the continuation expression on all procedures, OCL statements, and utility control statements. However, it cannot be used on a comment (*) statement. The + sign can be specified anywhere within the line; for example, within a parameter, substitution expression, or in the middle of a word. However, to make your procedures easier to read and maintain, only continue your statements at the end of words and expressions.

If a record ends with a shift-in character just before the continuation expression, and the first non-blank character of the next record is a shift-out character, both the shift-in and shift-out characters will be removed.

Calling a Procedure from Another Procedure

One procedure can call another procedure. A procedure called by another procedure is a **nested procedure**. This is generally helpful when the same procedure is called several times in a job. The procedure could be entered and stored only once and then called as often as necessary.

Suppose, for example, that a procedure named PAYROLL contains (in addition to other statements) a TAXES procedure command, and that another procedure named TAXES contains FEDER and STATE procedure commands. The TAXES, FEDER, and STATE procedures are all called and run when the operator enters the PAYROLL procedure command.

The four procedures are as follows:

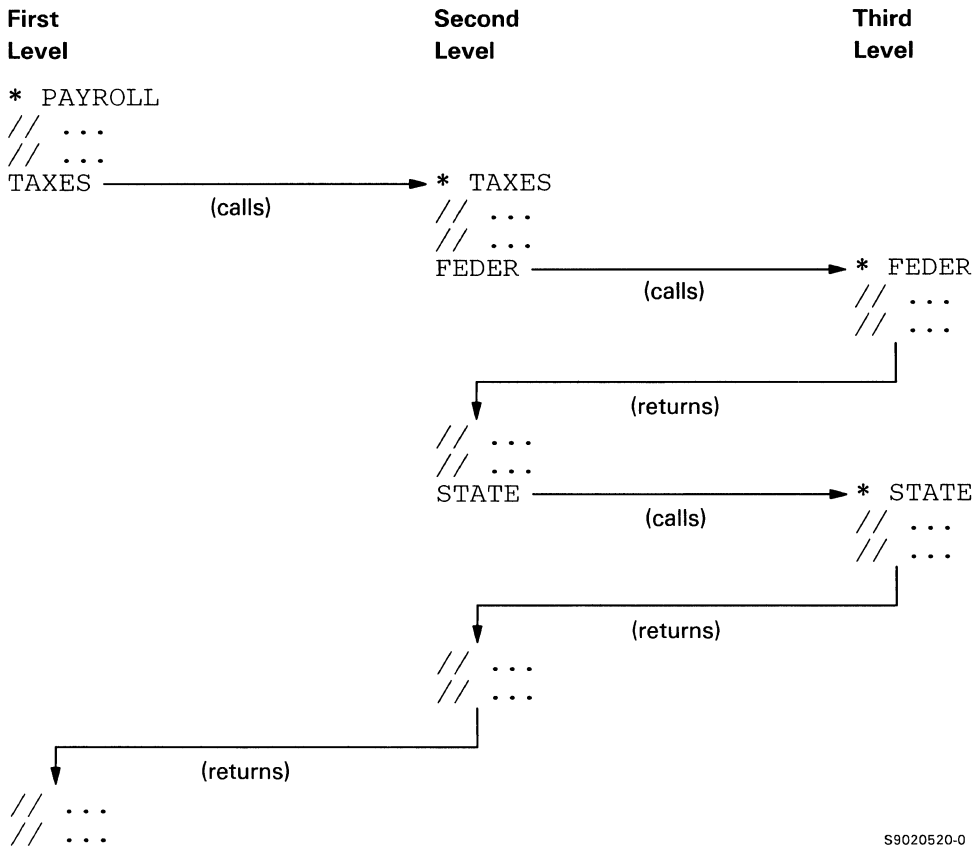
PAYROLL Procedure	**TAXES Procedure**	**FEDER Procedure**	**STATE Procedure**
* PAYROLL // ... // ... TAXES // ... // ...	* TAXES // ... // ... FEDER // ... // ... STATE // ... // ...	* FEDER // ... // ...	* STATE // ... // ...

In this example, TAXES, FEDER, and STATE are nested procedures. When a nested procedure ends, processing returns to the next statement in the calling procedure.

This example contains three levels of procedures: the first level contains PAYROLL, the second level contains TAXES, and the third level contains FEDER and STATE. One level can contain more than one procedure command, but no more than 255 levels of procedures are allowed. The procedures are called as follows:

Operator Enters

```
PAYROLL
```



Procedure Attributes

A procedure can be assigned the following attributes. These are assigned when the procedure is created or changed by SEU, or DSU, or when the procedure is created by the $MAINT utility program.

- Whether the procedure is a multiple requester terminal (MRT) procedure. For information about MRT programs and procedures, see the *Concepts and Programmer's Guide*. Normally, procedures are single requester terminal procedures.

- Whether OCL statements should be logged to the history file. OCL statements from the procedure can be prevented from being logged to the history file when the procedure is run; see the "LOG Procedure" on page 4-294. The procedure command that started a procedure is always logged to the history file. Normally, the procedure command and the OCL statements for your procedures are all logged to the history file. This is done to help you debug your procedures.

- Whether to pass parameters to the procedure or to pass data to the program. If the program data attribute is set, the data on the procedure command is passed to the first program run by the procedure. The data is passed on the first display station input or read request. The data starts with the first nonblank character following the procedure or library name and ends with the last nonblank character in the statement. For example:

  ```
  PAYROLL DATA01
  ```

  The PAYROLL procedure could have 2 data fields:

  1. Positions 1 through 4 (which contains DATA)

  2. Positions 5 and 6 (which contains 01)

  Every MRT procedure has this attribute even though the attribute may not have been selected when the procedure was created. Normally, single requester terminal procedures pass parameters to the procedure.

Example Procedures

Example 1: Procedure SAMPLE

This example shows a procedure named SAMPLE. This is a somewhat simple procedure because it has only 2 parameters, the file to be used and the creation date of the file. This procedure is not supplied as part of any program product, and must be entered by you if you want to test it. The procedure runs the $COPY utility program, and has a function similar to the LISTDATA procedure. For more information about the statements in this procedure, see the "Listing $COPY Files (LISTDATA/LISTFILE Procedures)" on page A-14.

```
SAMPLE   file name,[ mmddyy ]
                   [ ddmmyy ]
                   [ yymmdd ]
```

S9020008-0

file name specifies the name of a file to be listed. This parameter must be specified; if it is omitted, it is prompted for by the procedure.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file.

The statements in the SAMPLE procedure are as follows:

```
* SAMPLE Procedure
*  Parameter 1:  File name (required, prompted for if omitted)
*  Parameter 2:  File's creation date (optional)
// * 'SAMPLE PROCEDURE RUNNING'  1
// LOAD $COPY  2
// FILE NAME-COPYIN,LABEL-?1R?,  3
//   IFF ?2?= DATE-?2?,  4
//   UNIT-F1  5
// RUN  6
// COPYFILE OUTPUT-CHAR  7
// END  8
```

Description of the Lines in the SAMPLE Procedure

The lines contained in the SAMPLE procedure are described as follows. For more information about the OCL statements in this procedure, see Chapter 5, "OCL Statements." For more information about the COPYFILE utility control statement in this procedure, see "Listing $COPY Files (LISTDATA/LISTFILE Procedures)" on page A-14.

Line	Description
1	Displays a message indicating the procedure is running.
2	Loads the $COPY utility program into main storage.
3	Defines the name of the file to be listed. If the first parameter was not entered, it is prompted for using the ?1R? expression.
4	If a date was specified, the DATE parameter is included.

5 Specifies that the file is a disk file. Note that because the UNIT parameter does not end with a comma, it is the last parameter of the FILE statement. This allows the DATE parameter to be optional; that is, if no date parameter was specified, the DATE parameter is not generated.

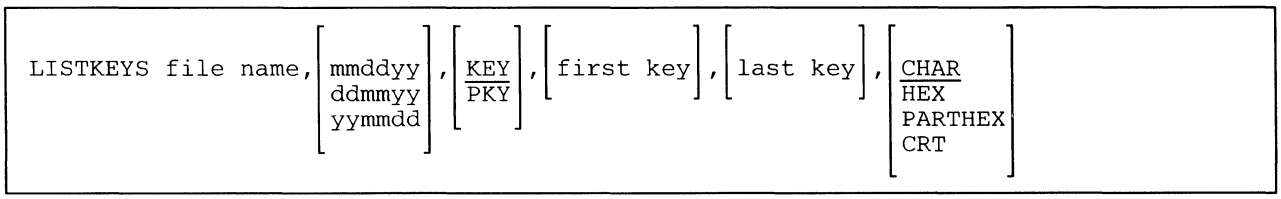
6 Starts running the $COPY program, which then reads the COPYFILE and END utility control statements.

7 Causes the printable characters in the file to be listed.

8 Indicates the end of the utility control statements to the $COPY program.

Example 2: Procedure LISTKEYS

This example shows a procedure named LISTKEYS. This is a somewhat more difficult procedure because it contains 6 parameters, 2 of which have defaults. This procedure is not supplied as part of any program product, and must be entered by you if you want to run it. The procedure runs the $COPY utility program and has a function similar to the LISTDATA procedure. For more information about the statements in this procedure, see the "Listing $COPY Files (LISTDATA/LISTFILE Procedures)" on page A-14.



S9020009-0

file name specifies the name of an indexed file to be listed. This parameter must be specified; if it is omitted, it is prompted for by the procedure.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file.

KEY specifies the indexed file has normal (unpacked) keys. If no parameter is specified, KEY is assumed.

PKY specifies the indexed file has packed keys.

first key specifies the first key in the file to be listed. If no parameter is specified, the file is listed starting with the first key in the file.

last key specifies the last key in the file to be listed. If no parameter is specified, the file is listed ending with the last key in the file. A last key can only be specified if a first key is specified.

CHAR specifies the printable characters in the file are to be listed. Any characters that are not printable are listed as blanks. If no parameter is specified, CHAR is assumed.

HEX specifies that both the printable characters and their hexadecimal representations are to be listed. Any unprintable characters are listed as blanks, but their hexadecimal representations are to be listed.

PARTHEX specifies the printable characters in the file are to be listed. Also, if any characters are not printable, those characters' hexadecimal representations are to be listed.

CRT specifies that the records are to be displayed.

The statements in the LISTKEYS procedure are as follows:

```
* LISTKEYS Procedure
// MEMBER USER1-MESSAGES,LIBRARY-MYLIB  1
// IF JOBQ-NO IF EVOKED-NO * 0001  2
// IF ?1?= EVALUATE ?1R'0002'?  3
// IF ?1?= RESET LISTKEYS ,?2?,?3?,?4?,?5?,?6?  4
// EVALUATE ?3'KEY'? ?6'CHAR'?  5
// IFF ?3?=KEY IFF ?3?=PKY ERR 0003,3  6
// IFF ?6?=CHAR IFF ?6?=HEX +
   IFF ?6?=PARTHEX IFF ?6?=CRT ERR 0004,3  7
// IF ?4?= IFF ?5?= ERR 0005,3  8
// LOAD $COPY  9
// FILE NAME-COPYIN,LABEL-?1?,  10
//   IFF ?2?= DATE-?2?,  11
//   UNIT-F1  12
// RUN  13
// COPYFILE OUTPUT-?6?,REORG-YES  14
// IFF ?4?= IF ?5?=  SELECT ?3?,FROM-?4?  15
// IFF ?4?= IFF ?5?= SELECT ?3?,FROM-?4?,TO-?5?  16
// END  17
// RETURN  18
*
* Parameters:
*  1  File name (required, prompted for if omitted)
*  2  File's creation date (optional)
*  3  KEY or PKY.  Indicates normal or packed keys
*     (default: KEY)
*  4  First key to list (default: first key in file)
*  5  Last key to list                          (default: last key in file)
*  6  CHAR, HEX, PARTHEX, or CRT.  Indicates how the
*     file is to be listed.  (default: CHAR)
```

The LISTKEYS procedure uses five messages from a message member named MESSAGES, which is in a library named MYLIB. The messages in MESSAGES are as follows:

```
MESSAGES,1
0001 The LISTKEYS procedure is running
0002 Enter the name of the indexed file to list:
0003 The third parameter must be KEY, PKY, or blank
0004 The sixth parameter must be CHAR, HEX, PARTHEX, CRT, or blank
0005 A last key is only allowed if a first key is entered
```

For more information about message members, see the "CREATE Procedure" on page 4-132.

Description of the Lines in the LISTKEYS Procedure

The lines in the LISTKEYS procedure are described as follows. For more information about the OCL statements in this procedure, see Chapter 5, "OCL Statements." For more information about the COPYFILE and SELECT utility control statements in this procedure, see "Listing $COPY Files (LISTDATA/LISTFILE Procedures)" on page A-14.

Line	Description
1	Indicates the first-level user message member (named MESSAGES).
2	Displays a message if the procedure is entered from the keyboard. The message has a number of 0001.
3	If the file name is omitted, a displayed message asks for the file name. The message has a number of 0002.
4	If the operator did not enter a name, the procedure is restarted. Any parameters the operator entered are included.
5	If no third parameter is specified, KEY is assumed; if no sixth parameter is specified, CHAR is assumed.
6	If the third parameter is not KEY or PKY, an error message is displayed. The message has a number of 0003 and an error option of 3. You would take the 3 option and cancel the procedure.
7	If the sixth parameter is not CHAR, HEX, PARTHEX, or CRT, an error message is displayed. Note how the + sign was used to continue this statement onto two lines. The message has a number of 0004 and an error option of 3. You would take the 3 option and cancel the procedure.
8	If a fourth parameter is not specified but a fifth parameter is specified, an error message is displayed. The message has a number of 0005 and an error option of 3. You would take the 3 option and cancel the procedure.
9	Loads the $COPY utility program into main storage.
10	Defines the name of the file to be listed.
11	If a date was specified, the DATE parameter is included.
12	Specifies that the file is a disk file. Note that because the UNIT parameter does not end with a comma, it is the last parameter of the FILE statement. This allows the DATE parameter to be optional; that is, if no date parameter was specified, the DATE parameter is not generated.
13	Starts running the $COPY program, which then reads the COPYFILE, SELECT, and END utility control statements.
14	Specifies how the file is to be listed (OUTPUT-?6?). REORG-YES specifies the records are to be listed sequentially by index key.

15 If parameter 4 is entered and parameter 5 is not entered, the file is listed starting with the specified key. If parameters 4 and 5 are not entered, the entire file is listed.

16 If both parameters 4 and 5 are entered, the file is listed starting with the specified key and ending with the specified key.

17 Indicates the end of the utility control statements to the $COPY program.

18 Ends the procedure. The comment statements following the RETURN are not read by the system.

Example 3: Procedure SCRNPRT

This example shows a procedure named SCRNPRT. This procedure calls a program that reads a file and prints information from the file. The procedure has 5 parameters, and shows a prompt display if the operator does not enter any parameters. The procedure also checks each parameter to ensure that parameter's value is correct. If an incorrect parameter is found, the prompt display is reshown with a message indicating the parameter in error; also the cursor is positioned to that parameter and the parameter is highlighted.

This procedure is not supplied as part of any program product, and must be entered by you if you want to run it.

```
SCRNPRT  file name,[printer id],[copies],[10],[6]
                   [P1      ]   [1     ]  [15]  [8]
                   [P2      ]
```

S9020010-0

If no first parameter is entered, a prompt display is shown.

file name specifies the name of a file to be listed.

printer id specifies the printer ID to be used. Either P1 or P2 can be specified; P1 is the default.

copies specify the number of copies to print. Any number from 1 through 5 can be entered; 1 is the default.

10 or 15 specifies characters per inch value to use. 10 is the default.

6 or 8 specifies lines per inch value to use. 6 is the default.

Figure 2-1 on page 2-17 shows a listing of the SCRNPRT procedure. Figure 2-2 shows a sample display that is shown by the procedure. Figure 2-3 on page 2-21 shows the S and D-specifications used to create the display.

```
1  * SCRNPRT
   * Prints the Print key display images contained in a disk file
   *
   * Parameter 1:     Name of file containing display images
   * Parameter 2:     Printer ID to use (P1 or P2)           Default: P1
   * Parameter 3:     Number of copies to print (1-5)        Default: 1
   * Parameter 4:     CPI value (10 or 15)                   Default: 10
   * Parameter 5:     LPI value (6 or 8)                     Default: 6
   *
   * Parameter 6:     Message for prompt display.
   *
   * Parameters 31 through 35 are used for error indicators
   *
   * Parameter 31: File name error
   * Parameter 32: Printer ID error
   * Parameter 33: Number of copies error
   * Parameter 34: CPI error
   * Parameter 35: LPI error
   *
   * Set parameter defaults
2  // EVALUATE ?2'P1'?  ?3'1'?  ?4'10'?  ?5'6'?
   *
3  // IFF ?1?= GOTO CHKFILE
   *
4  // IF ?L'1,8'?=SCRNSPOL EVALUATE P1=?L'10,8'?   Default for parm 1
   // IF ?L'1,7'?=SCRNPRT EVALUATE P1=?L'10,8'?    Default for parm 1
   *
5  // TAG PROMPT
   *
   // PROMPT MEMBER-SCRN,FORMAT-SCRNPRT,LENGTH-'8,2,1,2,1,75'
   // IF ?CD?=2007 RETURN                  Cmd7 - End Procedure
   *
   * Reset error indicators
6  // EVALUATE P31=  P32=  P33=  P34=  P35=
   *
7  // IFF ?1?= GOTO CHKFILE
   // EVALUATE P6='You must enter a file name.'  P31='X'
   // GOTO PROMPT
   *
8  // TAG CHKFILE
   *
   // IF DATAF1-?1? GOTO FILEOK
   // EVALUATE P6='File ?1? does not exist, enter a different name.' P31='X'
   // GOTO PROMPT
   *
9  // TAG FILEOK
   // LOCAL OFFSET-10,BLANK-8,DATA-'?1?'
   *
10 // IFF ?2?=P1 IFF ?2?=P2 IFF ?2?=PX GOTO P2ERR
   // GOTO P3CHK
   // TAG P2ERR
   // EVALUATE P6='Printer ID must be P1 or P2; re-enter value.'  P32='X'
   // GOTO PROMPT
   *
11 // TAG P3CHK
   *
   // IFF ?3?=1 IFF ?3?=2 IFF ?3?=3 IFF ?3?=4 IFF ?3?=5 GOTO P3ERR
   // GOTO P4CHK
   // TAG P3ERR
   // EVALUATE P6='Number of copies must be 1 through 5.'  P33='X'
   // GOTO PROMPT
   *
12 // TAG P4CHK
   *
   // IFF ?4?=10 IFF ?4?=15 GOTO P4ERR
   // GOTO P5CHK
   // TAG P4ERR
   // EVALUATE P6='Character per inch value must be 10 or 15.' P34='X'
```

Figure 2-1 (Part 1 of 2). Listing of SCRNPRT Procedure

```
   // GOTO PROMPT
   *
13 // TAG P5CHK
   *
   // IFF ?5?=6  IFF ?5?=8  GOTO P5ERR
   // GOTO RUN
   // TAG P5ERR
   // EVALUATE P6='Line per inch value must be 6 or 8.'  P35='X'
   // GOTO PROMPT
   *
14 // TAG RUN
   *
   // LOCAL BLANK-*ALL,DATA-'SCRNPRT'          Procedure name
   // LOCAL OFFSET-10,DATA-'?1?'               File name
   // LOCAL OFFSET-20,DATA-'?2?'               Printer ID
   // LOCAL OFFSET-30,DATA-'?3?'               Number of copies
   // LOCAL OFFSET-40,DATA-'?4?'               CPI value
   // LOCAL OFFSET-50,DATA-'?5?'               LPI value
   // IF ?5?=8 LOCAL OFFSET-60,DATA-'88'       Lines per page setting if LPI=8
   // IF ?5?=6 LOCAL OFFSET-60,DATA-'66'       Lines per page setting if LPI=6
   *
15 // INFOMSG NO
   *
16 BASICR SCRNPRT,TEMLIB,32,,SCRNPRT≠
   *
17 // IF DATAF1-SCRN.?WS? DELETE SCRN.?WS?,F1
   // INFOMSG YES
   // * 'Displays printed.'
```

Figure 2-1 (Part 2 of 2). Listing of SCRNPRT Procedure

Description of the SCRNPRT Procedure

The lines called out in the SCRNPRT procedure in Figure 2-1 on page 2-17 are described as follows. For more information about the OCL statements in this procedure, see Chapter 5, "OCL Statements."

Line Group	Description
1	Comment information. Describes the procedure in general and describes each parameter.
2	Sets the defaults for parameters 2 through 5.
3	If parameter 1 (the file name) is entered, the GOTO statement bypasses the initial showing of the prompt display.
4	If the operator did not enter a name, the file name stored in the local data area is assumed. The name is stored in positions 10 through 17 by either the SCRNPRT procedure (or by another procedure called SCRNSPOL).
5	Shows the prompt display. The display format load member is named SCRN. The format is named SCRNPRT (to match the procedure name). If the operator presses command key 7, the procedure ends.
6	Clears the parameters used for the display format error indicators.
7	If the file name is not entered on the prompt display, parameter 6 is set to a message and the prompt display is reshown. Parameter 31 is given a value, which sets on indicator 31 and causes the cursor to be positioned at the file name field, and shows the field in reverse image.
8	If the file name entered does not exist, parameter 6 is set to a message and the prompt display is reshown. Parameter 31 is given a value, which sets on indicator 31 and causes the cursor to be positioned at the file name field, and shows the field in reverse image.
9	If the file exists, the name is stored in the local data area.
10	If the printer ID entered is not P1 or P2, parameter 6 is set to a message and the prompt display is reshown. Parameter 32 is given a value, which sets on indicator 32 and causes the cursor to be positioned at the printer ID field, and shows the field in reverse image.
11	If the number of copies entered is not 1 through 5, parameter 6 is set to a message and the prompt display is reshown. Parameter 33 is given a value, which sets on indicator 33 and causes the cursor to be positioned at the number of copies field, and shows the field in reverse image.
12	If the characters per inch value entered is not 10 or 15, parameter 6 is set to a message and the prompt display is reshown. Parameter 34 is given a value, which sets on indicator 34 and causes the cursor to be positioned at the characters per inch field, and shows the field in reverse image.
13	If the lines per inch value entered is not 6 or 8, parameter 6 is set to a message and the prompt display is reshown. Parameter 35 is given a value, which sets on indicator 35 and causes the cursor to be positioned at the lines per inch field, and shows the field in reverse image.

14 Stores the parameter values in the local data area. The program reads these values to print the information properly. To ensure the paper lines up, the lines per page value is set based upon the lines per inch value.

15 Causes informational messages to not be displayed.

16 Runs a BASIC program that reads the file and prints information stored in the file.

17 Deletes a temporary work file created by the BASIC program and displays a message indicating the procedure is done.

```
                          SCRNPRT PROGRAM
This program lists the Print key display images contained in a disk file.

Name of the disk file that contains the display images . . . . . _______
ID of printer to be used . . . . . . . . . . . . . . . . . . P1,P2  P1
Number of copies to print  . . . . . . . . . . . . . . . . . 1-5   1
Character per inch value (15 valid for P1 only)  . . . . . 10,15  10
Line per inch value (8 valid for P1 only)  . . . . . . . . . 6,8   6



Cmd7 - End program
```

Figure 2-2. Sample SCRNPRT Display Prompt

The S- and D-specifications for the display format shown by the procedure are shown below. For more information about display formats, see the manual *Creating Displays*.

IBM International Business Machines Corporation

System/36 Display Format Specifications

First Edition

Use this coding sheet only to define display formats for WSU and $SFGR This coding sheet could contain typographical errors

GX21-9800-0 U/M 050*
Printed in U S A
*No of sheets per pad may vary slightly

Display Control Specification

S	Sequence Number	Specification Type	Display Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Null Fill	Reserved	Right-to-Left Display	WSU Only: Enter Mode Sequence: Start	End	Entry Required	Repeat	Reserved	Priority	Preprocess	Review Mode Record Identifying Indicators: 1	2	3	Insert Mode Record Identifying Indicators: 1	2	3	Reserved	Key Mask	Reserved
	1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16	17 18	19 20	21	22	23 24	25 26	27	28	29 30	31 32	33 34	35 36	37 38	39	40	41	42	43	44	45	46	47	48 49	50 51	52 53	54 55	56 57	58 59	60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
		S	SCRNPRT									Y																					G		

The H specifications are optional If you choose to use the H specifications, they must be placed between the S specification and the D specifications for a display format Use the other side of this form to code the H specifications

Field Definition Specification

D	Sequence Number	Specification Type	WSU Field Name / Field Name	WSU Arrays: Elements per Row or Column	Fill Direction	Line Skip Factor	Horizontal Positioning Space Factor	Starting Location: Line Number	Horizontal Position	Field Length	Output Data	Edit Code (WSU Only)	Input Allowed	Data Type	Mandatory Fill	Mandatory Entry	Self-Check	Adjust/Fill	Position Cursor	Enable Dup	Controlled Field Exit	Auto Record Advance	Protect Field	High Intensity	Blink Field	Nondisplay	Reverse Image	Underline	Column Separators	Reserved	Lowercase	Reserved	Constant Type	Constant Data	Continuation
	1 2 3 4 5	6	7 8 9 10 11 12 13 14	15	16	17	18	19 20	21 22	15 16 17 18	23 24	25 26	27	28	29	30	31	32 33	34	35	36	37 38	39 40	41 42	43 44	45 46	47 48	49 50	51	52 53 54 55	56	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
		D						29	226	Y									Y													C	SCRNPRT PROGRAM	X
		D																																
		D						79	4	2Y																						C	This program lists t	X
		D	he Print key display images contained in a disk file.																																
		D						68	7	2Y																						C	Name of the disk fil	X
		D	e that contains the display images																																
		D	FILENAME					8	771Ø1	Y							31									31		Y								
		D						67	9	3Y																						C	ID of printer to be u	X
		D	sed																							P1, P2								
		D	PRTID					2	971Ø2	Y							32									32		Y								
		D						6711	3Y																						C	Number of copies to p	X
		D	rint																							. 1-5								
		D	COPIES					1	1171Ø3	Y							33									33		Y								
		D						6613	3Y																						C	Character per inch va	X
		D	lue (15 valid for P1 only)																							1Ø, 15								
		D	CPI					2	1371Ø4	Y							34									34		Y								
		D						6615	3Y																						C	Line per inch value (X	
		D	8 valid for P1 only)																							. 6, 8								
		D	LPI					1	1571Ø5	Y							35									35		Y								
		D	MSG					7518	506															06											
		D						1923	2Y																						C	Cmd7 - End program	
		D																																
		D																																

Figure 2-3. S- and D-Specifications for SCRNPRT Display Format

Procedure Performance Tips and Coding Techniques

After you know how to make procedures, you can use this section to improve the performance of your procedures.

- Use GOTO and TAG statements rather than several redundant IF expressions. Use one IF expression and a GOTO expression to reduce the time needed to evaluate several IF expressions. The statements skipped by the GOTO and TAG expressions are not processed. For example, rather than doing this:

```
// IF ?1?/Y LOAD $MAINT
// IF ?1?/Y RUN
// IF ?1?/Y COPY FROM-#LIBRARY,NAME-TEST,LIBRARY-P,TO-PRINT
// IF ?1?/Y END
.
.
.
```

  Do this, which avoids duplicating the tests for parameter 1 by using GOTO and TAG statements:

```
// IFF ?1?/Y GOTO A
//   LOAD $MAINT
//   RUN
//   COPY FROM-#LIBRARY,NAME-TEST,LIBRARY-P,TO-PRINT
//   END
// TAG A
.
.
.
```

- Use ELSE statements if you have more than one IF expression and only one of the expressions can be true. All ELSE statements are skipped after a true IF expression. For example, rather than doing this, which processes all three statements even though only one of the statements will be true:

```
// IF ?2?/T SWITCH 1XXXXXXX
// IF ?2?/J SWITCH X1XXXXXX
// IF ?2?/S SWITCH XX1XXXXX
```

  Do this, which stops processing after the first true condition:

```
// IF ?2?/T            SWITCH 1XXXXXXX
//   ELSE IF ?2?/J     SWITCH X1XXXXXX
//     ELSE IF ?2?/S   SWITCH XX1XXXXX
```

- Combine IF expressions where possible. The remainder of a statement is not processed after a false condition. For example, rather than doing this (which wastes space in the library):

```
// IF ?2?/T GOTO NEXT
// IF ?2?/J GOTO NEXT
// IF ?2?/S GOTO NEXT
// GOTO ERROR
// TAG NEXT
.
.
.
// RETURN
// TAG ERROR
// PAUSE 'ERROR IN PARAMETER 2'
// CANCEL
```

Do this, which checks the value of parameter 2 and if it is not equal to T, J, or S, the ERROR is processed:

```
// IFF ?2?/T IFF ?2?/J IFF ?2?/S GOTO ERROR
.
.
.
// RETURN
// TAG ERROR
// PAUSE 'ERROR IN PARAMETER 2'
// CANCEL
```

- Avoid using the informational message (// *) statement to display prompting messages (such as: ENTER MEMBER NAME or ENTER LIBRARY NAME). Use the PROMPT OCL statement and a display format instead. The advantages are:

  - More information can be displayed.
  - Fewer disk operations are required.
  - For remote display stations, fewer data transmissions are made. The // * statement must save the current display contents, show the message, and reshow the display after the procedure ends. The PROMPT statement just shows the display format without having to save the current display contents.

  See the "PROMPT OCL Statement" on page 5-97 for more information.

- After you have tested your procedures, stop the logging of the OCL statements to the history file. You may only need to have the OCL statements logged when you are creating and testing your procedure. You can stop the logging by either of two ways:

  - The source entry utility (SEU) has an end of job option that allows you to specify whether the statements should be logged. See the *SEU Guide*.
  - The LOG command or OCL statement can specify whether the statements should be logged. See the "LOG Procedure" on page 4-294 or the "LOG OCL Statement" on page 5-72.

- Avoid the use of comments within a procedure. Reading comments takes up system time.

- Use your own libraries for your applications; that is, run procedures and programs from a library other than the system library (#LIBRARY). The system library has a very large directory, and therefore more time is needed to search for a library member in the system library than for the same member in one of your libraries.

  Also, the SSP always searches the current library first, and if the member is not found, then it searches the system library. See the "SLIB Procedure" on page 4-471 and the "LIBRARY OCL Statement" on page 5-67 for more information about setting the current library.

- Use substitution expressions to concatenate values. For example:

```
// IFF ?1?/0 IFF ?1?/1 GOTO ERROR
// SWITCH XXX?1?XX?1?X
```

  If the first parameter is 1, the SWITCH statement will be:

```
// SWITCH XXX1XX1X
```

  If the first parameter is 0, the SWITCH statement will be:

```
// SWITCH XXX0XX0X
```

- Concatenate values to create unique names. For example the ?WS? expression, which substitutes the current display station ID, can be used to create a file name that will be unique for each display station:

```
// FILE NAME-OUTPUT,LABEL-FILEA?WS?
```

  This allows more than one operator to use the procedure containing this statement because each display station would have its own unique work file. The program refers to the output file as OUTPUT, and if an operator at display station W1 ran the procedure, the actual name of the file would be FILEAW1.

- Use IF conditional expressions to avoid making the system operator respond to an informational message when a procedure is sent to the job queue or when the procedure is started by the EVOKE OCL statement or an SSP-ICF evoke operation. For example:

```
// IF JOBQ-NO IF EVOKED-NO * 'Procedure running'
```

  This example would display the message only when the procedure was being run from the display station; that is, not from the job queue and not evoked.

- Change the value of a parameter, which allows an operator to use fewer keystrokes. For example:

```
// * 'ENTER 1 TO PROCESS MONTHLY; 2 TO PROCESS WEEKLY'
// IF       ?1R?=1 EVALUATE P1='MONTHLY'
//  ELSE IF ?1?=2  EVALUATE P1='WEEKLY'
//         ELSE  CANCEL
INVENTRY ?1?
```

  If the operator enters 1, the procedure INVENTRY MONTHLY is run; if the operator enters 2, the procedure INVENTRY WEEKLY is run. If neither 1 or 2 is entered, the procedure is canceled.

Debugging Your Procedures

You can use the DEBUG and LOG OCL statements, and the HISTORY procedure to debug your procedures. See the "DEBUG OCL Statement" on page 5-28, the "LOG OCL Statement" on page 5-72, and the "HISTORY Procedure" on page 4-209 for more information.

Chapter 3. Procedure Control Expressions

This chapter describes the expressions and statements you can use to control the processing of your procedures.

These expressions and statements can be placed anywhere among the statements in a procedure; that is, anywhere among the OCL statements and utility control statements. The procedure control expressions include the following:

- Comment statements
- Substitution expressions
- IF conditional expressions
- ELSE expressions
- The informational message (// *) statement
- The system console message (// **) statement
- The CANCEL statement
- The EVALUATE statement
- The GOTO and TAG statements
- The PAUSE statement
- The RESET statement
- The RETURN statement

Things You Can Do Using Procedure Control Expressions

This section lists several tasks you may want to do using procedure control expressions. The tasks are listed in the following groups:

- "Substituting Values and Information"
- "Displaying Messages or Display Formats" on page 3-5
- "Data File Information" on page 3-5
- "Library and Folder Information" on page 3-6
- "Diskette Information" on page 3-6
- "Tape Information" on page 3-7
- "Comparing and Evaluating Values and Branching in Procedures" on page 3-7
- "Testing the Procedure or Job Environment" on page 3-7
- "Ending Procedures" on page 3-8

Substituting Values and Information

Command Keys

To check whether a command or function key was pressed, see the "?CD? (Return Code) Expression" on page 3-16.

Date and Time

To substitute the current date, see the "?DATE? (Program Date) Expression" on page 3-18.

To change the date, see the "DATE OCL Statement" on page 5-25.

To substitute the current time, see the "?TIME? (System Time) Expression" on page 3-24.

Library Names

To substitute the current library name, see the "?CLIB? (Current Library) Expression" on page 3-18.

To substitute the session library name, see the "?SLIB? (Session Library) Expression" on page 3-23.

To change the session or the current library, see the "SLIB Procedure" on page 4-471 or the "LIBRARY OCL Statement" on page 5-67.

Local Data Area

To substitute data from the local data area, see the "?L'position,length'? (Local Data Area) Expression" on page 3-19.

To place information into the local data area, see the "LOCAL OCL Statement" on page 5-70.

Message Text

To substitute data from a message member, see the "?Mmic? or ?M'mic,position,length'? (Message Member) Expression" on page 3-21.

To specify which message member is to be used, see the "MEMBER OCL Statement" on page 5-73.

Parameters

To substitute the value of a parameter that was entered on a procedure statement, see the following:

- The "?n? (Parameter) Expression" on page 3-11
- The "?n'value'? (Default Parameter) Expression" on page 3-11
- The "?nT'value'? (Temporary Value Parameter) Expression" on page 3-12
- The "?nF'value'? (Forced Value Parameter) Expression" on page 3-12
- The "?R? (Required Parameter) Expression" on page 3-13
- The "?nR? (Missing Parameter) Expression" on page 3-13
- The "?nR'mic'? (Missing Parameter Message) Expression" on page 3-14
- The "EVALUATE Statement" on page 3-60

To substitute the length of a parameter, see the "?Cn? (Parameter Length) Expression" on page 3-15.

Printers

To substitute the printer ID of the session printer, see the "?PRINTER? (Session Printer) Expression" on page 3-22.

To substitute the system list device (either a printer ID or whether the output is to be printed or displayed), see the "?SYSLIST? (System List Device) Expression" on page 3-23.

To specify a printer, see the "PRINT Procedure" on page 4-350 or the "PRINTER OCL Statement" on page 5-83.

Procedure or Menu

To substitute the outermost (first-level) procedure name that is being run, see the "?PROC? (First Level Procedure) Expression" on page 3-23.

To substitute the current user menu name, see the "?MENU? (Current Menu) Expression" on page 3-22.

To cause a menu to be displayed, see the "MENU OCL Statement" on page 5-75.

Return Codes

To check whether a compilation was successful, see the "?CD? (Return Code) Expression" on page 3-16.

User ID

To substitute the operator's user ID who is running the procedure, see the "?USER? (Operator's User ID) Expression" on page 3-24.

Work Station ID

To substitute the work station ID that is running the procedure, see the "?WS? (Display Station ID) Expression" on page 3-26.

Displaying Messages or Display Formats

Display Formats

To cause a display format to be shown at the display station that is running the procedure, see the "PROMPT OCL Statement" on page 5-97.

To determine which command key was pressed on a PROMPT display, see the "?CD? (Return Code) Expression" on page 3-16.

Messages

To display an informational message at the display station that is running the procedure, see the "// * (Informational Message) Statement" on page 3-57.

To display a message at the system console, see the "// ** (System Console Message) Statement" on page 3-59.

To display a message because a required parameter was not specified, see the "?nR? (Missing Parameter) Expression" on page 3-13, the "?R? (Required Parameter) Expression" on page 3-13, or the "?nR'mic'? (Missing Parameter Message) Expression" on page 3-14.

To display a message, and to pause for the operator to respond to the message, see the "ERR Procedure" on page 4-176, or the "PAUSE Statement" on page 3-69.

To specify which message member is to be used, see the "MEMBER OCL Statement" on page 5-73. For additional information on dual language message members, see the "$MGBLD Utility" on page A-82.

Data File Information

File Existence

To determine whether a data file exists on the system, see the "DATAF1 (Files, Libraries, and Folders on Disk) Condition" on page 3-31.

To determine whether a data file exists on diskette, see the "DATAI1 (Files on Diskette) Condition" on page 3-32.

To determine whether a data file exists on tape or tape cartridge, see the "DATAT (Files on Tape) Condition" on page 3-34.

File Size

To determine the allocated size of a data file, see the "?F'S,name'? or ?F'S,name,date'? (File Size) Expression" on page 3-18.

To determine the actual number of records contained in a data file, see the "?F'A,name'? or ?F'A,name,date'? (Actual File Size) Expression" on page 3-19.

Space Available

To determine the number of disk blocks available, see the "BLOCKS (Available Disk Space) Condition" on page 3-30.

Library and Folder Information

Library Names

To substitute the name of the current library, see the "?CLIB? (Current Library) Expression" on page 3-18.

To substitute the name of the session library, see the "?SLIB? (Session Library) Expression" on page 3-23.

To specify a library, see the "SLIB Procedure" on page 4-471 or the "LIBRARY OCL Statement" on page 5-67.

Library Existence

To determine whether a library exists on the system, see the "DATAF1 (Files, Libraries, and Folders on Disk) Condition" on page 3-31.

To determine whether a library exists on a diskette, see the "DATAI1 (Files on Diskette) Condition" on page 3-32.

To determine whether a library exists on tape or tape cartridge, see the "DATAT (Files on Tape) Condition" on page 3-34.

Member Existence

To determine whether a source member exists in a library, see the "SOURCE (Library Source Members) Condition" on page 3-46.

To determine whether a procedure member exists in a library, see the "PROC (Library Procedure Members) Condition" on page 3-44.

To determine whether a subroutine member exists in a library, see the "SUBR (Library Subroutine Members) Condition" on page 3-47.

To determine whether a load member exists in a library, see the "LOAD (Library Load Members) Condition" on page 3-42.

Space Available

To determine the number of disk blocks available, see the "BLOCKS (Available Disk Space) Condition" on page 3-30.

Diskette Information

Volume ID

To substitute the volume ID of a diskette, see the "?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression" on page 3-25.

To test for the volume ID of a diskette, see the "VOLID (Diskette and Tape Volume IDs) Condition" on page 3-53.

Existence

To determine whether a file or library exists on a diskette, see the "DATAI1 (Files on Diskette) Condition" on page 3-32.

Tape Information

Volume ID

To substitute the volume ID of a tape or tape cartridge, see the "?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression" on page 3-25.

To test for a volume ID of a tape or tape cartridge, see the "VOLID (Diskette and Tape Volume IDs) Condition" on page 3-53.

Existence

To determine whether a file or library exists on a tape or tape cartridge, see the "DATAT (Files on Tape) Condition" on page 3-34.

Comparing and Evaluating Values and Branching in Procedures

Comparing

To test the values of parameters or to test for one or more conditions, see the "IF Conditional Expressions" on page 3-28.

To compare one expression with another, see "string1=string2 (Comparing, Equal to) Condition" on page 3-50 or "string1>string2 (Comparing, Greater Than) Condition" on page 3-52.

Evaluating

To assign a value to a parameter, to add, subtract, multiply, or divide values, to set the ?CD? (return code) expression, or to evaluate substitution expressions, see the "EVALUATE Statement" on page 3-60.

Branching

To branch to another statement in a procedure, see the "GOTO and TAG Statements" on page 3-67.

Testing the Procedure or Job Environment

Display Station Type

To determine whether a procedure is being run from the system console, see "CONSOLE (System Console) Condition" on page 3-30. To determine whether a procedure is being run from a display station that can display ideographic characters during an ideographic session, or that can display 27 lines and 132 characters per line, see "DSPLY (Display Station Type) Condition" on page 3-36.

Job Queue

To determine whether a procedure is being run from the job queue, see "JOBQ (Job Queue) Condition" on page 3-40.

Evoked Procedure

To determine whether a procedure was started by an EVOKE OCL statement or by the Interactive Communications Feature (SSP-ICF), see "EVOKED (Evoked Procedures) Condition" on page 3-38.

Inquiry Mode

To determine whether a procedure is being run during Inquiry mode, see "INQUIRY (Inquiry Mode) Condition" on page 3-39.

MRT Requestors

To determine whether the maximum number of requester terminals are attached to a multiple requester terminal (MRT) procedure, see "MRTMAX (Multiple Requesting Terminals) Condition" on page 3-43.

Subsystems Enabled

To determine whether an Interactive Communications Feature (SSP-ICF) subsystem is enabled, see the "ENABLED (Enabled Communications) Condition" on page 3-37.

Security Level

To determine whether password security is active on the system, see the "SECURITY (Password Security) Condition" on page 3-45.

To determine the security classification of an operator running the procedure, see the "SECURITY (Password Security) Condition" on page 3-45.

Running Procedures

To determine whether one or more specified procedures are currently running on the system, see the "ACTIVE (Running Procedures) Condition" on page 3-29.

Switches

The switches are also called indicators U1 through U8. To determine the switch settings, see the "SWITCH (Switches) Condition" on page 3-48.

Ending Procedures

End

To cancel a procedure, see the "CANCEL Statement" on page 3-60.

To end a nested procedure and return to the calling procedure, see the "RETURN Statement" on page 3-71.

To end a procedure and call the same or another procedure, see the "RESET Statement" on page 3-70.

* (Comment) Statement

Comment statements usually explain the purposes of the statements in a procedure. Comments are not displayed when the procedure runs. Comments are listed only when the procedure is printed or displayed.

```
* comment
```

S9020011-0

comment can be any combination of words and characters. The * must be entered in column 1. Any characters following the * are not processed when the procedure is run.

Example 1

This example shows a comment statement as the first statement in a procedure.

```
* TESTA procedure
// LOAD PROG1
// FILE NAME-INPUT
// RUN
```

Example 2

If you have many comment statements, put a RETURN statement at the end of the procedure and put your comments after the RETURN. This way the system processes the RETURN statement and your comments are not processed (thus saving the amount of time the system would otherwise have used to read the comments). For example:

```
// ...
// ...     (statements in the procedure)
// ...
// RETURN
*
*
*          (comments)
*
*
```

Substitution Expressions

Substitution expressions allow you to substitute information into the statements processed when the procedure is run. Examples of information that can be substituted are:

- Positional parameters on the statement that called the procedure.
- Information supplied by the operator in response to a display prompt or a message issued from within the procedure. If a procedure that issues messages is placed on the job queue, the messages appear on the system console, and the system operator must know the correct responses.
- Specified positions in the local data area. (For an explanation of how to place data into the local data area, see the "LOCAL OCL Statement" on page 5-70.)

Substitution expressions can be used while entering statements from the keyboard or in the command statement in a menu. However, a substitution expression for a *positional* parameter (one of the following expressions: ?n?, ?n'value'?, ?nT'value'?, ?nF'value'?, ?nR?, ?nR'mic?) entered from the keyboard or contained in a menu will result in a null substitution, that is, no value is substituted. A substitution expression for a *nonpositional* parameter results in the proper substitution. For example, if you were entering the following statement from the keyboard:

```
// FILE NAME-?2'FILEA'?,DATE-?DATE?
```

If the current date is 021480, the following statement is generated:

```
// FILE NAME-,DATE-021480
```

Comment statements, which are indicated by an asterisk (*) in position 1 of the statement, are not completely processed by the SSP; therefore substitution is not performed on comment statements. Substitution occurs whenever a valid expression is encountered, even if the expression is in the comment portion of an OCL statement.

Substitution expressions always begin and end with a question mark. A substitution expression begins any time a question mark is immediately followed by a number or by one of the following letters: C, D, F, L, M, P, R, S, T, U, V, or W. This means that, for example, if you specify ?C in a procedure, it will always be processed as a substitution expression. An error message may be displayed if you entered the ?C as part of anything else. A question mark followed by any other letter is not treated as a substitution expression.

The following are descriptions of the substitution expressions that can be used in a procedure. Examples of how to use the substitution expressions are included.

?n? (Parameter) Expression

This expression substitutes the value of the nth positional parameter. If the nth parameter does not have a value, no value is substituted.

n is a number from 1 through 64 that specifies the parameter to be substituted.

For example, a procedure contains the following statement:

```
// * '?3? WAS DELETED'
```

If the third parameter does not have a value (that is, it was not specified on the procedure statement and was not assigned a value by a previous statement within this procedure), the following statement is generated:

```
// * ' WAS DELETED'
```

If the value of the third parameter is FILEX, the following statement is generated:

```
// * 'FILEX WAS DELETED'
```

?n'value'? (Default Parameter) Expression

This expression substitutes the value of the nth positional parameter; or, if the nth parameter does not have a value, the expression permanently assigns a default value to the parameter and then substitutes that value.

n is a number from 1 through 64 that specifies the parameter to be substituted.

value specifies the value to be assigned to the parameter if the parameter currently has no value. Any following references to the nth parameter within the procedure use the assigned value.

For example, a procedure contains the following statement:

```
// FILE NAME-?2'FILEA'?
```

If the second parameter does not have a value, the following statement is generated:

```
// FILE NAME-FILEA
```

This example shows how references to the second parameter that follow the default expression also use the value FILEA.

```
// FILE NAME-?2'FILEA'?
*
*
// FILE NAME-?2?
```

If the second parameter does not have a value, the following statements are generated:

```
// FILE NAME-FILEA
*
*
// FILE NAME-FILEA
```

?nT'value'? (Temporary Value Parameter) Expression

This expression substitutes the value of the nth positional parameter; or, if the nth parameter does not have a value, the expression temporarily assigns a value to the parameter and then substitutes the value.

n is a number from 1 through 64 that specifies the parameter to be substituted.

T indicates that the value is temporarily assigned to the parameter.

value specifies the temporary value. A temporary value is used only for the current substitution expression. For other references to the nth parameter within the procedure, the parameter does not have a value.

For example, a procedure contains the following statements:

```
// FILE NAME-?2T'FILEA'?
// * 'THE SECOND PARAMETER IS ?2T'NOT DEFINED: FILEA ASSUMED'?'
// * 'THE SECOND PARAMETER IS NOW ?2?'
```

If the second parameter does not have a value, the following statements are generated:

```
// FILE NAME-FILEA
// * 'THE SECOND PARAMETER IS NOT DEFINED: FILEA ASSUMED'
// * 'THE SECOND PARAMETER IS NOW '
```

Note that the ?2? expression in the last statement substituted the original value of parameter 2, which was blank.

If the second parameter was FILEC, the following statements are generated:

```
// FILE NAME-FILEC
// * 'THE SECOND PARAMETER IS FILEC'
// * 'THE SECOND PARAMETER IS NOW FILEC'
```

?nF'value'? (Forced Value Parameter) Expression

This expression immediately forces a new value to be assigned to the nth positional parameter, even if the nth positional parameter already has a value.

n is a number from 1 through 64 that specifies the parameter to be substituted.

F indicates the value is to be assigned to the parameter regardless of the parameter's current value.

value specifies the value that is to be substituted into the statement containing the expression.

For example, a procedure contains the following statements:

```
// LOAD PROG1
// FILE NAME-INPUT,LABEL-?3F'FILEA'?
// RUN
// LOAD PROG2
// FILE NAME-INPUT,LABEL-?3?
// RUN
```

The following statements are generated, regardless of parameter 3's value before program PROG1 was loaded.

```
// LOAD PROG1
// FILE NAME-INPUT,LABEL-FILEA
// RUN
// LOAD PROG2
// FILE NAME-INPUT,LABEL-FILEA
// RUN
```

To change a parameter so that it no longer has a value, you can specify the following:

```
?nF''?
```

?R? (Required Parameter) Expression

This expression displays the message:

```
Enter required parameter
```

and waits for the operator to enter (at the keyboard) the value to be substituted. R indicates that an operator reply is required. Up to 60 characters can be entered in response to the message. For example, a procedure contains the following statement:

```
// FILE NAME-?R?
```

When the statement is encountered, the message `Enter required parameter` is displayed. The operator then types FILEA at the keyboard, presses the Enter key, and the following statement is generated:

```
//  FILE NAME-FILEA
```

See "Procedure Parameters" on page 5-3 for more information on procedure parameters.

?nR? (Missing Parameter) Expression

This expression substitutes the value of the nth positional parameter; or, if the nth parameter does not have a value, displays a message:

```
Enter missing parameter
```

and waits for the operator to enter (at the keyboard) the value to be substituted. References to the nth parameter that follow this expression use the value entered by the operator. Up to 60 characters can be entered in response to the message.

n is a number from 1 through 64 that specifies the parameter to be substituted.

R indicates that the parameter is required.

For example, a procedure contains the following statement:

```
// FILE NAME-INPUT,LABEL-?1R?
```

If the first parameter does not have a value, the message `Enter missing parameter` is displayed. The operator then enters FILEA from the keyboard, and the following statement is generated:

```
// FILE NAME-INPUT,LABEL-FILEA
```

See "Procedure Parameters" on page 5-3 for more information on procedure parameters.

?R'mic'? (Required Parameter Message) Expression

This expression displays a message from the current user first-level message member and waits for the operator to enter the value to be substituted. Up to 60 characters can be entered in response to the message. See the "MEMBER OCL Statement" on page 5-73 for information about assigning a message member to a procedure. See the "CREATE Procedure" on page 4-132 for information about creating a message member.

R indicates that the parameter is required.

mic identifies the message identification code of the message to be displayed.

For example, a procedure contains the following statements:

```
// MEMBER USER1-MESSAGES
// LOAD PROGA
// FILE NAME-?R'0015'?
// RUN
```

When this statement is processed, the SSP displays message 0015 from the current first-level message member named MESSAGES:

```
Enter the name of the file:
```

The operator then enters the word PAYROLL at the keyboard and the following is generated for the FILE OCL statement:

```
// FILE NAME-PAYROLL
```

See "Procedure Parameters" on page 5-3 for more information on procedure parameters.

?nR'mic'? (Missing Parameter Message) Expression

This expression substitutes the value of the nth positional parameter; or, if the nth parameter does not have a value, displays a message from the current first-level message member and waits for the operator to enter the value to be substituted. Up to 60 characters can be entered in response to the message. References to the nth parameter that follow this expression use the value entered by the operator.

See the "MEMBER OCL Statement" on page 5-73 for information about assigning a message member to a procedure. See the "CREATE Procedure" on page 4-132 for information about creating a message member.

n is a number from 1 through 64 that specifies the parameter to be substituted.

R indicates that the parameter is required.

mic identifies the message identification code of the message to be displayed if the nth parameter does not have a value.

For example, a procedure contains the following statements:

```
// MEMBER USER1-MESSAGES
// LOAD PROGA
// FILE NAME-?1R'0015'?
// RUN
// * 'FILE ?1? WAS USED'
```

If the first parameter is not entered, message 0015 from the current first-level message member named MESSAGES is displayed:

```
Enter the name of the file:
```

The operator then enters the word PAYROLL and the following is generated for the FILE OCL statement:

```
// FILE NAME-PAYROLL
```

See "Procedure Parameters" on page 5-3 for more information on procedure parameters.

?Cn? (Parameter Length) Expression

This expression substitutes the length of the nth positional parameter. The result of the substitution is a 3-digit number with leading zeros.

C indicates that character length information is being substituted.

n is a number from 1 through 64 that specifies the parameter to be substituted.

For example, a procedure contains the following statement:

```
// IF ?C2?>8 * '?2? HAS MORE THAN 8 CHARACTERS'
```

If parameter 2 has a value of FILEABCDE, the following statement would be generated:

```
// IF 009>8 * 'FILEABCDE HAS MORE THAN 8 CHARACTERS'
```

?C'value'? (Length) Expression

This expression substitutes the length of the specified value. The result of the substitution is a 3-digit number with leading zeros.

C indicates that character length information is being substituted.

value specifies the value whose length is to be substituted. If the value is less than zero, the minus sign is included in the length. For example, -23 has a length of 003.

For example, a procedure contains the following statement:

```
// * '?USER? HAS ?C'?USER?'? CHARACTERS'
```

If the operator has a user ID of SUSAN, the following statement would be generated:

```
// * 'SUSAN HAS 005 CHARACTERS'
```

?CD? (Return Code) Expression

This expression allows you to check conditions in a procedure. This expression substitutes a 4-digit return code set by the SSP, by a program product, or by a procedure using the EVALUATE statement. The possible return codes are:

Return Code	Meaning
0000	The previous job step ended normally, or this step is the first step in the job. This value is also returned when the Enter key is pressed on a PROMPT OCL statement.
1002	Warning errors were found in the COBOL compilation.
1004	Conditional errors were found in the COBOL compilation.
1008	Serious errors were found in the previous job step (ASM, COBOLC, FORMAT, FORTRANC, OLINK, RPGC, or WSU procedures).
1010	In response to the ERR procedure, an operator took the 0 option.
1011	In response to the ERR procedure, an operator took the 1 option.
1012	In response to the ERR procedure, an operator took the 2 option.
1312	The previous job step was an MRT program that ended without releasing the display station.
1991	The HELP procedure was canceled by the operator pressing command key 7.
2001-2024	Command keys 1 through 24 returned from the PROMPT OCL statement display. 2001 means command key 1, 2002 means command key 2, ..., 2024 means command key 24.
2030	The file specified on the FILE OCL statement is not available. It is currently in use by a suspended program, a program with the never-ending program attribute, a job waiting because of a WAIT OCL statement, or a job that has acquired the file using a FILE OCL statement outside of a LOAD and RUN OCL statement pair.
2031	The file specified on the FILE OCL statement is currently being used. You can try again to allocate the file.
2032	The diskette drive specified by the ALLOCATE OCL statement is not available. It is currently in use by a suspended program, a program with the never-ending program attribute, a job waiting because of a WAIT OCL statement, or a job that used the ALLOCATE OCL statement to allocate the drive.
2033	The diskette drive specified by the ALLOCATE OCL statement is currently being used. You can try again to allocate the drive.
2034	Errors were found by the CREATE procedure ($MGBLD utility program).
2035	Tape drive 1 specified by the ALLOCATE OCL statement is not available. It is currently in use by a suspended program, or a program with the never-ending program attribute, or a job waiting because of a WAIT OCL statement, or a job that used the ALLOCATE OCL statement to allocate the drive.
2036	Tape drive 1 specified by the ALLOCATE OCL statement is currently being used. You can try again later to allocate the drive.

Figure 3-1 (Part 1 of 2). ?CD? Return Code Definitions

Return Code	Meaning
2037	Tape drive 2 specified by the ALLOCATE OCL statement is not available. It is currently in use by a suspended program, or a program with the never-ending program attribute, or a job waiting because of a WAIT OCL statement, or a job that used the ALLOCATE OCL statement to allocate the drive.
2038	Tape drive 2 specified by the ALLOCATE OCL statement is currently being used. You can try again later to allocate the drive.
2040	The printer specified in the PRINTER OCL statement is already being continued by the CONTINUE-YES parameter.
2041	The tape cartridge drive specified by the ALLOCATE OCL statement is currently being used by a suspended program, a program with the never-ending program attribute, or a job waiting because of a WAIT OCL statement, or a job that used the ALLOCATE OCL statement to allocate the drive.
2042	The tape cartridge drive specified by the ALLOCATE OCL statement is currently being used. You can try again later to allocate the drive.
2043	The tape cartridge drive or diskette drive specified by the ALLOCATE OCL statement is not available. A common non-sharable, waitable system resource is being used by the diskette drive or tape cartridge drive.
2044	The tape cartridge drive or diskette drive specified by the ALLOCATE OCL statement is not available. A common non-sharable system resource is being used by the diskette drive or tape cartridge drive or it is currently in use by a suspended program, a program with the never-ending program attribute, a job waiting because of a WAIT OCL statement, or a job that used the ALLOCATE OCL statement to allocate the drive.
2045	The maximum number of MRTs that can be attached has been exceeded. You can try your request again later.
2090	Roll Up (Roll ↑) key from PROMPT OCL statement.
2091	Roll Down (Roll ↓) key from PROMPT OCL statement.
2092	Help key from PROMPT OCL statement.
2093	Record Backspace key from PROMPT OCL statement. (The Home key was pressed while the cursor was in the home position.)
2143	The HELP procedure was canceled by the operator pressing command key 3.
3721	The operator canceled the previous job step by selecting the 2 option in response to a message, or the previous step was an MRT program and the operator released the display station by interrupting the MRT program (by pressing the Attn key) and then selecting the 2 option.
8158	The SSP-ICF session abnormally ended.

Figure 3-1 (Part 2 of 2). ?CD? Return Code Definitions

In the following example, if the return code equals 1008, a message is sent to the display station that started the procedure.

```
// IF ?CD?=1008  MSG ?WS?,'AN ERROR OCCURRED DURING THE COMPILE'
```

The SSP resets the return code to 0000 whenever it processes a RUN OCL statement.

?CLIB? (Current Library) Expression

This expression substitutes the name of the current library. See the "LIBRARY OCL Statement" on page 5-67 for more information about the current library. For example, a procedure contains the following statements:

```
// LIBRARY NAME-MYLIB
// JOBQ ?CLIB?,PROC
```

The following JOBQ statement is generated:

```
// JOBQ MYLIB,PROC
```

The current library is the library that is currently active. The current library is established when you process the LIBRARY OCL statement within a procedure.

?DATE? (Program Date) Expression

This expression substitutes the current program date. For example, a procedure contains the following statement:

```
// FILE NAME-FL?DATE?
```

If the current program date is 021483, the following statement is generated. Note that the format is the current session date format:

```
// FILE NAME-FL021483
```

?F'S,name'? or ?F'S,name,date'? (File Size) Expression

This expression substitutes the number of blocks or records reserved (or allocated) for a resident (T) disk file. The value substituted is given in the units (blocks or records) specified when the file was created. The result of the substitution is an 8-digit number with leading zeros. If the file label does not exist in the VTOC, 00000000 is substituted. If 2 or more files exist on disk with the same name but no date is specified, the size of the file with the most recent creation date is substituted.

F indicates that file size information is substituted.

S indicates that the allocated size is substituted.

name is the name of the file.

date is the creation date of the file.

For example, a procedure contains the following statement:

```
// FILE NAME-FILEB,BLOCKS-?F'S,FILEA'?
```

If FILEA was created with a size of 50 blocks, the following statement will be generated for FILEB:

```
// FILE NAME-FILEB,BLOCKS-00000050
```

This type of substitution, when used with a FILE OCL statement, could result in an error if the file does not exist.

?F'A,name'? or ?F'A,name,date'? (Actual File Size) Expression

This expression substitutes the actual number of data records in a resident (T) disk file. The result of the substitution is an 8-digit number with leading zeros. If a perm (T) file is used as an S file and the job ends normally, the file is scratched. Until it is scratched, it will give a positive record number. If 2 or more files exist on disk with the same name and no date is specified, the size of the file with the most recent creation date is substituted.

F indicates that file size information is substituted.

A indicates that the actual number of records is substituted.

name is the name of the file.

date is the creation date of the file.

For example, a procedure contains the following statement:

```
// FILE NAME-FILEB,RECORDS-?F'A,FILEA'?
```

If FILEA actually contains 150 data records, the following statement is generated for FILEB, which creates FILEB with a size of 150 records:

```
// FILE NAME-FILEB,RECORDS-00000150
```

These types of substitutions, when used with a FILE OCL statement, could result in an error if the file does not exist or if the file is empty.

If the file is a remote file, the number of records substituted does not reflect any records added by a user currently using the file.

?L'position,length'? (Local Data Area) Expression

This expression substitutes a value from the 512-byte display station local data area. For information on changing the display station local data area, see the "LOCAL OCL Statement" on page 5-70. The data is substituted from either the user or system local data area, depending on the last AREA parameter of the LOCAL OCL statement. If no previous AREA parameter was specified, the user local data area is assumed. The system local data area is used by IBM supplied procedures, and any data you place in the system local data area can be lost. If you call an IBM supplied procedure that uses the system local data area, the data in your user area is not affected. Also, the local data area being used when you call an IBM procedure is recognized as the local data area to use when the called procedure ends.

L indicates that a value is substituted from the local data area.

position specifies the starting position (a number from 1 through 512, where 1 specifies the first position) of the local data area to be substituted.

length specifies the number of positions in the local data area to be substituted. Beginning blanks and embedded blanks (blanks inside the characters being substituted, for example, `'ABC DEF'`) are allowed in the substituted value, but trailing blanks are not substituted.

For example, a procedure contains the following statement:

```
// FILE NAME-INPUT,LABEL-?L'12,8'?,UNIT-F1
```

If eight positions, from 12 through 19, of the local data area contain `'FILEA   '`, the following statement is generated:

```
// FILE NAME-INPUT,LABEL-FILEA,UNIT-F1
```

If the substituted data is then used in an expression or statement that has a length restriction, the data must obey that restriction.

CAUTION

If the local data area contains IGC data, make sure that an equal number of shift-in and shift-out characters are substituted; otherwise, the results will be unpredictable.

?Mmic? or ?M'mic,position,length'? (Message Member) Expression

This expression substitutes a value from a statement in the first-level (USER1) message member. For information about assigning a message member, see the "MEMBER OCL Statement" on page 5-73.

M indicates that a value is substituted from a statement in the message member.

mic specifies the message identification code of the message containing the value to be substituted.

position specifies the starting position (a decimal number from 1 through 75, where 1 specifies the first position) of the message text to be substituted.

length specifies the decimal number of positions of the value to be substituted.

If position and length are not specified, the entire message text is substituted. Beginning blanks and embedded blanks (blanks inside the characters being substituted, for example, `'ABC DEF'`) are allowed in the substituted value, but trailing blanks are truncated. For example, a procedure contains the following statement:

```
// * '?M'0014,10,8'? WAS USED'
```

If eight positions, from 10 through 17, of message 0014 contain `'FILEA   '`, the following statement is generated:

```
// * 'FILEA WAS USED'
```

If the substituted data is then used in an expression or statement that has a length restriction, the data must obey that restriction.

CAUTION

If the message contains IGC data, make sure that an equal number of shift-in and shift-out characters are substituted; otherwise, the results will be unpredictable.

For information about retrieving ideographic messages from the message member, see "Considerations for the Ideographic Version of the SSP" on page 4-135.

?MENU? (Current Menu) Expression

This expression substitutes the 1- through 6-character menu name of the currently active user menu. If no menu is currently active, no value is substituted.

For example, a procedure contains the following statement:

```
// IF ?MENU?=SAMPLE   MENU MINE
```

If the menu named SAMPLE is currently active, the test will be true and the MENU OCL statement will be processed to display the menu MINE. If the menu was not named SAMPLE, the MENU statement will not be processed.

?PRINTER? (Session Printer) Expression

This expression substitutes the 2-character value that indicates the session printer. The SET or PRINT procedure is used to set the session printer; see "SET Procedure" on page 4-454 or "PRINT Procedure" on page 4-350.

For example, a procedure contains the following statement:

```
// SYSLIST ?PRINTER?
```

If the printer ID of the session printer is P2, the following statement is generated:

```
// SYSLIST P2
```

If the session printer is defined to be SYSTEM, the actual printer ID of the system printer is substituted. If the session printer is SYSTEM and no system printer is defined, the word SYSTEM is substituted.

?PROC? (First Level Procedure) Expression

This expression substitutes the 1- to 8-character name of the first-level procedure that is running. The first-level procedure is the first procedure called in a series of nested procedures. For example, a procedure contains the following statement:

```
// * '?PROC? IS RUNNING'
```

If the name of the first level procedure is PROCABC, the following statement is generated:

```
// * 'PROCABC IS RUNNING'
```

?SLIB? (Session Library) Expression

This expression substitutes the name of the active library for the session. For example, the session library is MYLIB, and a procedure contains the following statement:

```
// JOBQ ?SLIB?,PROCEDUR
```

The following statement is generated:

```
// JOBQ MYLIB,PROCEDUR
```

The session library is the library that is active at the keyboard. The session library is established when you:

- Sign on to the system
- Run the SLIB procedure
- Enter the LIBRARY OCL statement at the keyboard
- Enter the MENU command with a library name

?SYSLIST? (System List Device) Expression

This expression substitutes the 2- to 3-character value that indicates where the system list output is being displayed or printed. The SYSLIST or PRINT procedure is used to set the system list device, see "SYSLIST Procedure" on page 4-498 or "PRINT Procedure" on page 4-350. The values substituted are:

OFF if the current system list device is set to off

CRT if the system list output is set to CRT

printer id (a 2-character printer ID) if the system list output is set to a printer

For example, a procedure contains the following statement:

```
// IF ?SYSLIST?=CRT  SYSLIST ?PRINTER?
```

If the system list device is CRT, and if the session printer is P2, the following statement is generated:

```
// IF CRT=CRT  SYSLIST P2
```

If the system list device is defined to be SYSTEM and no system printer is defined, CRT is substituted.

?TIME? (System Time) Expression

This expression substitutes the current system time in the format: **HHMMSS**, where HH is the hours, MM is the minutes, and SS is the seconds. The system time is based upon the time the system operator enters during initial program load (IPL). For example, a procedure contains the following statement:

```
// * 'THE TIME IS ?TIME?'
```

If the current system time is 9:45:24, the following statement is generated:

```
// * 'THE TIME IS 094524'
```

?USER? (Operator's User ID) Expression

This expression substitutes the 1- to 8-character user ID assigned to the operator who started the job. For example, a procedure contains the following statement:

```
// LOCAL DATA-'?USER?'
```

If the user ID of the operator is MIKE, the following statement is generated:

```
// LOCAL DATA-'MIKE'
```

For MRT procedures, blanks are substituted for the ?USER? expression.

Note: You should avoid using ?USER? to begin a file name if the first character of the user ID to be substituted is numeric because the first character of a file name must not be numeric.

?VOLID? or ?VOLID'location'? (Diskette or Tape Volume ID) Expression

This expression substitutes the 1- to 6-character volume ID of the diskette or tape in the specified location. Use this expression with care; otherwise, you may change or erase the wrong diskette or tape.

location specifies the diskette slot, magazine locations, or tape unit to be searched for the diskettes or tapes. On a system without a diskette magazine drive, specifying any diskette location results in S1. The location can be any of the following:

S1 specifies the volume ID of the diskette in slot 1. If no location is entered, this location is assumed.

S2 specifies the volume ID of the diskette in slot 2.

S3 specifies the volume ID of the diskette in slot 3.

M1.nn specifies the volume ID of the diskette in magazine 1, position nn. nn can be any number from 01 through 10.

M2.nn specifies the volume ID of the diskette in magazine 2, position nn. nn can be any number from 01 through 10.

T1 specifies the volume ID of the tape on tape drive 1.

T2 specifies the volume ID of the tape on tape drive 2.

TC specifies the volume ID of the tape in the tape cartridge drive.

For example, a procedure contains the following statement:

```
// * 'THE VOLUME ID OF THE DISKETTE IS ?VOLID'M1.02'?'
```

If the volume ID of the diskette in magazine 1, position 2 is VOL001, the following statement is generated:

```
// * 'THE VOLUME ID OF THE DISKETTE IS VOL001'
```

Use this expression with caution. It is possible you may bypass the normal volume ID checking that is meant to ensure that the proper diskette or tape is being used.

?WS? (Display Station ID) Expression

This expression substitutes the 2-character work station ID of the display station that called the procedure. For example, a procedure contains the following statement:

```
// FILE NAME-INPUT,LABEL-FILEA?WS?
```

If the procedure is called from display station W3, the following statement is generated:

```
// FILE NAME-INPUT,LABEL-FILEAW3
```

If the procedure is run from the job queue, the SSP substitutes the ID of the display station from which the job was placed on the queue.

If the procedure is run using the EVOKE OCL statement, the SSP substitutes the ID of the display station from the job that issued the evoke. If the procedure is run using the evoke operation of the Interactive Communications Feature (SSP-ICF), the SSP substitutes the SSP-ICF session ID (00 through 99).

Nested Substitution Expressions

You can nest substitution expressions (that is, have substitution expressions that call other substitution expressions). For example, a procedure contains the following statement:

```
?M'0014,1,29'?
```

This expression specifies that the first 29 characters of message number 0014 should be substituted. In this example, the first 29 characters of message 0014 are:

```
// FILE NAME-FILEA?WS?
```

Therefore, the statement that results from the first substitution expression contains another substitution expression. The SSP then processes the latter expression. If the work station ID of the display station that started the job is W1, the resulting statement is:

```
// FILE NAME-FILEAW1
```

By using nested expressions, you can use a substitution expression as a default value within another expression. For example:

```
?1'?2?'?
```

instructs the SSP to substitute the value of the first parameter. If the first parameter does not have a value, the SSP substitutes the value of the second parameter.

When using nested substitution expressions, take care to avoid expressions that might cause an infinite loop. For example, a procedure contains the following statement:

```
// FILE NAME-?1R?
```

If the operator did not specify the first parameter, a message is displayed telling the operator to enter the first parameter value. If the operator enters ?1?, the SSP will **continually** substitute ?1? into the statement. The operator will have to interrupt and then cancel the procedure.

You can have an unlimited number of nested substitution expressions in any one statement.

IF Conditional Expressions

Conditional expressions within a procedure allow you to conditionally process certain OCL and utility control statements when the procedure is run.

The IF expression can be used only within a procedure. An IF expression tests for a specified condition; if the condition is met, the specified statement is processed.

IF expressions can have any of the following formats:

```
// IF   condition    statement
Or:
// IFT  condition    statement
Or:
// IFF  condition    statement
```

S9020012-0

IF and IFT test whether the condition specified by **condition** is true. If the specified condition is true, the statement following the condition is processed.

IFF tests whether the condition specified by **condition** is false (if not true). If the specified condition is false, the statement following the condition is processed.

The **condition** portion of the IF expression must be a continuous string of nonblank characters. A blank indicates the end of the condition. Blanks between apostrophes are allowed, for example 'AB D'. The conditions you can specify are shown beginning with "ACTIVE (Running Procedures) Condition" on page 3-29.

An IF or IFT test for multiple conditional expressions is true when one of these multiple conditions is met. For example:

```
// IF ACTIVE-'PROC1,PROC2' PROC3
```

Procedure PROC3 is run if PROC1, PROC2, or both PROC1 and PROC2 are running on the system.

An IFF test for multiple conditional expressions is false when all of the multiple conditions are met. For example:

```
// IFF ACTIVE-'PROC1,PROC2' PROC3
```

Procedure PROC3 is run if both PROC1 and PROC2 are not running on the system.

The **statement** portion of the IF expression is described in "Statement Portion of IF Conditional Expression" on page 3-54.

You can also use an ELSE expression to do a function when the IF expression is not satisfied. See "ELSE Expressions" on page 3-55.

When you are using certain conditional expressions, the specified condition may be true **at the time** the condition was tested. However, it may not be true when you try to process something at a later time.

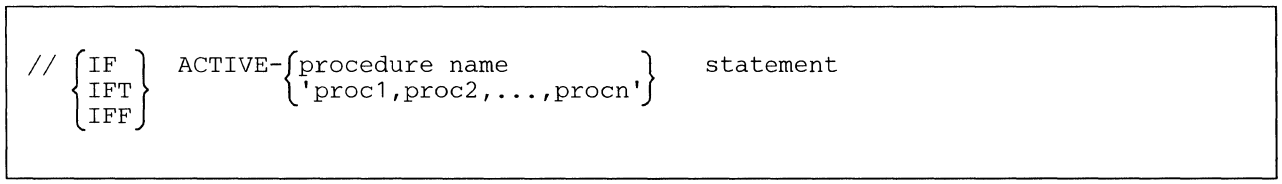
Processing based upon the results of an IF expression should be done with care because the condition being tested for may have changed.

ACTIVE (Running Procedures) Condition

The ACTIVE conditional expression determines whether one or more procedures are running on the system.

If the condition is true, at least one of the specified procedures was active **at the time** the ACTIVE test was evaluated. Processing based upon the ACTIVE test should be done with caution because the condition being tested for may have changed. If you try to test for the procedure that contained the ACTIVE test, the test will always be false. The ACTIVE test cannot be used to determine if you are the last user of an MRT.

If an NRT is to be tested, the procedure name must be the procedure that contains the // RUN statement of the NRT.



S9020013-0

procedure name specifies the name of a procedure. The condition is true if the specified procedure is currently running on the system.

'proc1,proc2,...,procn' specifies two or more procedure names. For IF and IFT tests, the condition is true if any of the specified procedures are currently running on the system. For IFF tests, the condition is false if all the specified procedures are not running on the system.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if a procedure called PROC1 is active, the system should cancel the running of the procedure that contains the IF test.

```
// IF ACTIVE-PROC1 CANCEL
```

Example 2

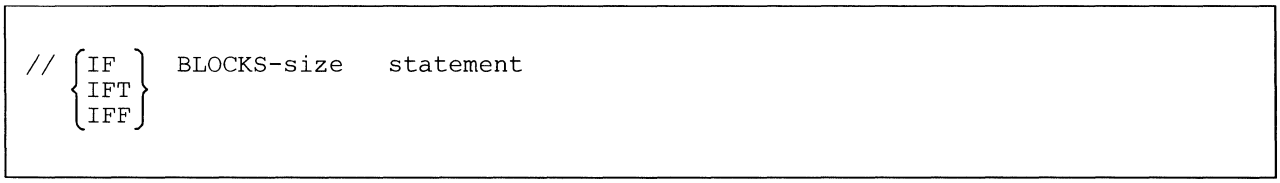
This example specifies the procedure PAYROLL can be run only if both procedure PROC1 and procedure PROC2 are not active.

```
// IFF ACTIVE-'PROC1,PROC2' PAYROLL
```

BLOCKS (Available Disk Space) Condition

The BLOCKS conditional expression determines whether the specified amount of disk space is available.

If the condition is true, the specified number of blocks was available **at the time** the BLOCKS expression was evaluated. If other programs are running, another program might use that available space before your program tries to use it.



S9020014-0

size specifies the number of blocks, and can be a 1- to 8-digit number. This condition is true if the specified number of contiguous blocks (that is, blocks that are all together in one place) is available on the disk.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

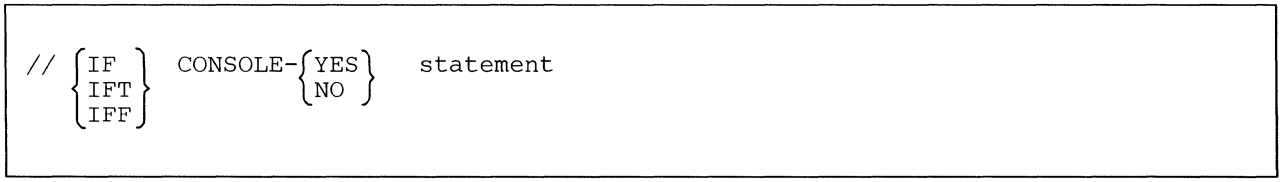
Example

This example specifies that if 150 contiguous disk blocks are not available, the job should be canceled.

```
// IFF BLOCKS-150 CANCEL
```

CONSOLE (System Console) Condition

The CONSOLE conditional expression determines whether the procedure is being run from the system console.



S9020015-0

YES is true if the procedure is currently running from the system console.

NO is true if the procedure is not running from the system console.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

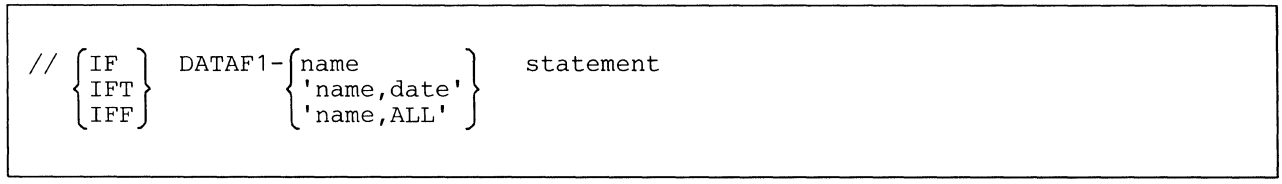
Example

This example specifies that if the procedure is not being run on the system console, the system should cancel the running of the procedure.

```
// IF CONSOLE-NO  CANCEL
```

DATAF1 (Files, Libraries, and Folders on Disk) Condition

The DATAF1 conditional expression determines whether the specified file,library, or folder exists on disk. This condition is true if a resident (T) file, library, or folder exists on the disk with the name and creation date (optional) specified. If ALL is specified, the condition is true if a file, library, or folder exists on the disk that starts with the characters in the specified name. This test will **not** detect a scratch (S) or job (J) file with the specified name.



S9020016-0

name specifies the name of the file, library, or folder to be searched for.

date specifies the creation date of the file. The date is optional. If the date is not specified, the file having the most recent creation date is assumed. If the date is specified, it must be in the session date format. If the name of a library or folder is specified, the date should not be specified. If the name of a library or folder and the date are specified, a false condition is returned. The date may be entered in one of the following formats: mmddyy, ddmmyy, or yymmdd.

ALL specifies that the disk is searched for a file, library, or folder that starts with the characters in the specified name. If no files, libraries, or folders that start with the given characters exist, a false condition is returned. If the ALL parameter is specified, only local files, libraries, and folders will be detected. If the ALL parameter is not specified, the network resource directory is searched for the file, library, or folder.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example

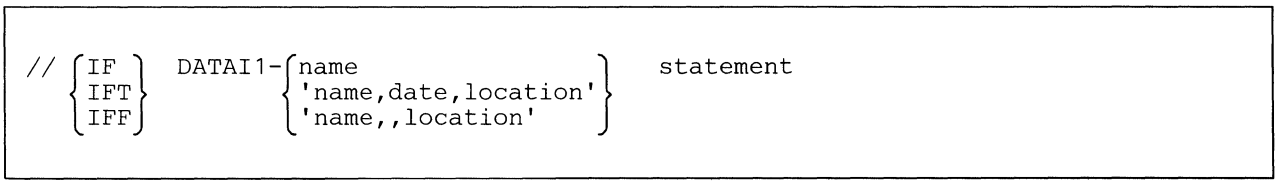
This example specifies that if a disk file with the name specified by the second positional parameter does not exist, the BLDFILE procedure should be run to create the file.

```
// IFF DATAF1-?2? BLDFILE ?2?,S,BLOCKS,10,100
```

DATAI1 (Files on Diskette) Condition

The DATAI1 conditional expression determines whether the specified file exists on diskette. The condition is true if a file exists on the diskette with the name and creation date (optional) specified. If your system has a diskette magazine drive, the location parameter specifies the location of the diskette or diskettes to be searched.

For magazine drives: if more than one slot is being checked, the test is stopped when a match is found; therefore, not all slots may have been tested. When the condition is true for a magazine drive, the diskette remains in the diskette reader until the magazine drive door is opened, a procedure is run which uses the diskette drive, or the job step or job ends.



S9020475-0

name specifies the name of the file to be searched for.

date specifies the creation date of the file. The date is optional. If the date is not specified, the test is true when the first file is found. The date may be entered in any of the following formats: mmddyy, ddmmyy, or yymmdd.

location specifies the location of the diskette to be checked if the system has a diskette magazine drive. It can be any of the following:

S1, S2, or S3 identifies an individual diskette slot. If a location is not specified, S1 is assumed.

ALLS, specifies that all slots (S1, S2, and S3) should be searched.

M1.nn identifies a location within magazine 1. **nn** can be any number from 01 through 10. For example, M1.04 indicates location 4 within magazine 1.

M2.nn identifies a location within magazine 2. **nn** can be any number from 01 through 10.

ALL1 specifies that all diskettes in magazine 1 should be searched.

ALL2 specifies that all diskettes in magazine 2 should be searched.

ALL which specifies that all diskettes in both magazines should be searched.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if a file named FILEA exists on the diskette in slot 1, the DELETE procedure is to be run to delete that file.

```
// IF DATAI1-'FILEA,,S1' DELETE FILEA,I1
```

Example 2

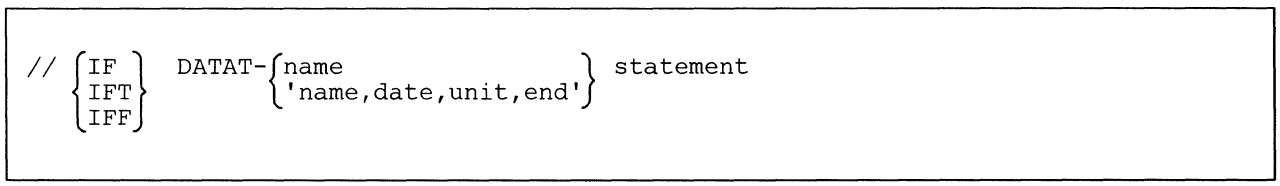
This example specifies that if a file named FILEB does not exist on the diskette in magazine 1, slot 5, the SAVE procedure is to be run to copy the file to the diskette.

```
// IFF DATAI1-'FILEB,,M1.05' SAVE FILEB,,,VOL001,M1.05
```

DATAT (Files on Tape) Condition

The DATAT conditional expression determines whether the specified file exists on tape. The condition is true if a file exists on the specified tape drive with the name and creation date (optional) specified.

If a tape is not mounted on the specified unit or if the specified unit is offline, an error message will be issued.



S9020017-0

name specifies the name of the file to be searched for.

date specifies the creation date of the file. The date is optional. If the date is not specified, the test is true when the first file with the specified name is found. The date may be entered in any one of the following formats: mmddyy, ddmmyy, or yymmdd.

unit specifies the tape unit to be checked if the system has more than one tape drive. It can be any of the following:

T1 checks the tape mounted on tape drive 1. If unit is not specified, T1 is assumed.

T2 checks the tape mounted on drive 2.

TC checks the tape in the tape cartridge drive.

end specifies the position of the tape after the tape has been searched. It can be any of the following:

REWIND specifies that the tape is to be rewound to the beginning of the tape after it is searched. If end is not specified, REWIND is assumed.

LEAVE specifies that the tape is to remain in the position where it found the file.

UNLOAD specifies that the tape is to be rewound and unloaded for removal after it is searched. If UNLOAD and TC are specified, the tape will be rewound after processing.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if a file named FILEA does not exist on tape drive 1, the SAVE procedure is to be run to copy the file to the end of the tape.

```
// IFF DATAT-'FILEA,,T1,LEAVE' SAVE FILEA,,,VOL001,T1,,,,,,,LEAVE
```

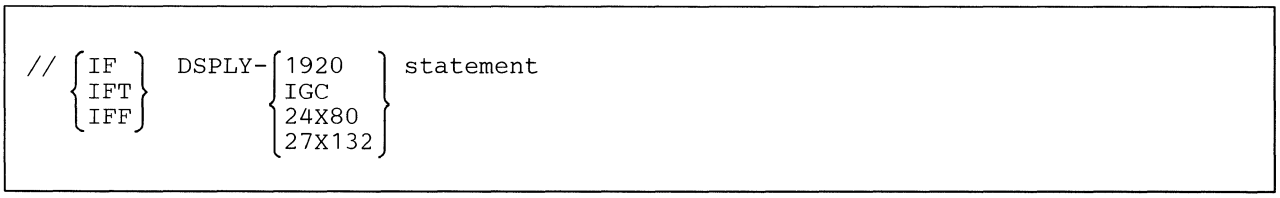
Example 2

This example specifies that if a file named FILEB does exist on tape drive 2, the RESTORE procedure is to be run to copy the file from tape to disk.

```
// IF  DATAT-'FILEB,,T2,REWIND' RESTORE FILEB,,,,,,T2
```

DSPLY (Display Station Type) Condition

The DSPLY conditional expression determines the type of display station that is being used.



S9020018-0

1920 is supported for compatibility with the IBM System/34. This test will always be true.

IGC is valid for the ideographic version of the SSP and is true if the procedure containing the expression is run during an ideographic session at a display station that can display ideographic characters.

24X80 is the same as specifying DSPLY-1920. It means that the procedure is run at a display station that can display 24 lines and 80 characters per line. This test will always be true.

27X132 is true if the procedure is run at a display station that can display 27 lines and 132 characters per line. The 3180 Model 2 display station can switch back and forth between this 27-line by 132-character display and a 24-line by 80-character display.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example

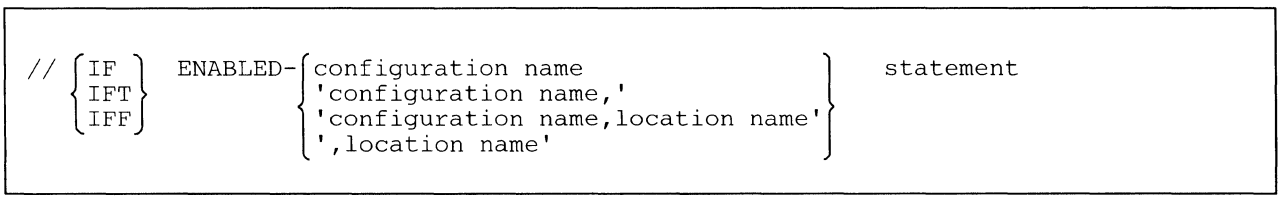
This example specifies that during an ideographic session, if the display station from which this expression is run is an ideographic display station, then the PROMPT OCL statement should display a format called FMT1 in a load member called FORMATS.

```
// IF DSPLY-IGC PROMPT MEMBER-FORMATS,FORMAT-FMT1
```

ENABLED (Enabled Communications) Condition

The ENABLED conditional expression determines whether the specified Interactive Communications Feature (SSP-ICF) configuration name and location name are enabled.

Note: If ',location name' is specified and the name corresponds to an APPN location, then the condition result is guaranteed to be correct only if the location was explicitly enabled.



S9020019-0

configuration name or 'configuration name,' is true if the specified SSP-ICF configuration name is enabled.

'configuration name,location name' is true if the specified SSP-ICF configuration name and its location name are enabled.

',location name' is true if the specified SSP-ICF location name is enabled.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if the SSP-ICF configuration name COMNAME is not enabled, the message `Config name COMNAME not enabled` should be displayed.

```
// IFF ENABLED-COMNAME * 'Config name COMNAME not enabled'
```

or:

```
// IFF ENABLED-'COMNAME,' * 'Config name COMNAME not enabled'
```

Example 2

This example specifies that if the SSP-ICF configuration name COMNAME and the location name LOC1 are not enabled, the message `Communications not enabled` should be displayed.

```
// IFF ENABLED-'COMNAME,LOC1' * 'Communications not enabled'
```

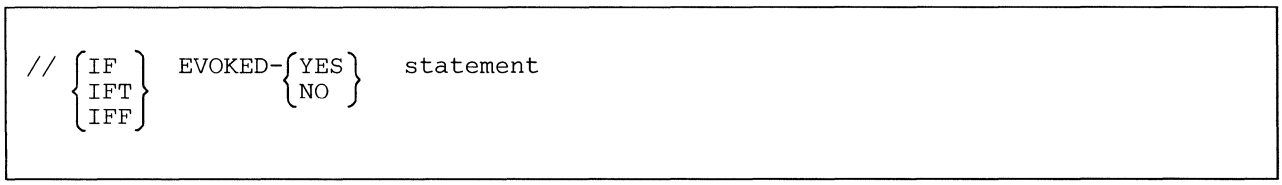
Example 3

This example specifies that if the SSP-ICF location name LOC1 is enabled, the message `Communications enabled` should be displayed.

```
// IF ENABLED-',LOC1' * 'Communications enabled'
```

EVOKED (Evoked Procedures) Condition

The EVOKED conditional expression determines whether a procedure was started by an Interactive Communications Feature (SSP-ICF) evoke operation or by the EVOKE OCL statement.



S9020020-0

YES is true if the procedure was evoked through SSP-ICF or by the EVOKE OCL statement.

NO is true if the procedure was not evoked through SSP-ICF or by the EVOKE OCL statement.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if the procedure containing this statement was evoked by the EVOKE OCL statement, the message `Procedure running` will be sent to the display station that did the evoke operation.

```
// IF EVOKED-YES MSG ?WS?,Procedure running
```

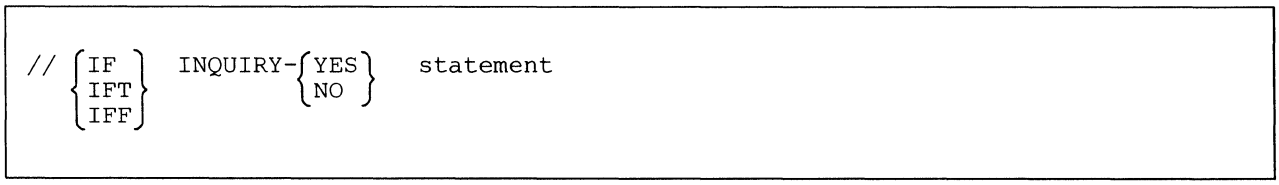
Example 2

This example specifies that if the procedure containing this statement was not evoked by the EVOKE OCL statement and is not running from the job queue, the message `Procedure running` will be displayed at the display station.

```
// IF JOBQ-NO IF EVOKED-NO * 'Procedure running'
```

INQUIRY (Inquiry Mode) Condition

The INQUIRY conditional expression determines whether a procedure is being run under inquiry mode. That is, the operator used the Attn (attention) key to interrupt a program to run this procedure.



S9020021-0

YES is true if the procedure containing the expression is running in inquiry mode.

NO is true if the procedure is not running under inquiry mode.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

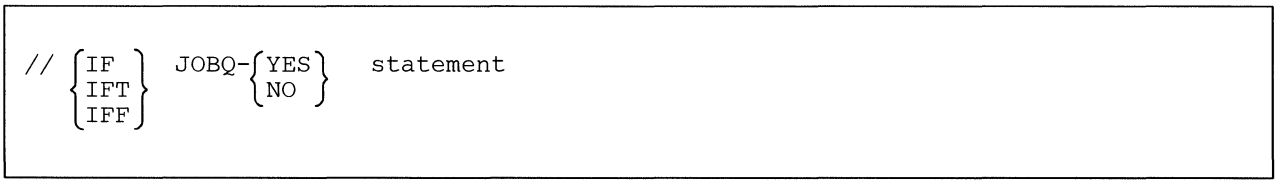
Example

This example specifies that if the procedure containing this statement is running in inquiry mode, the procedure is to be canceled.

```
// IF INQUIRY-YES CANCEL
```

JOBQ (Job Queue) Condition

The JOBQ conditional expression determines whether a procedure is being run from the job queue.



S9020022-0

YES is true if the procedure is run from the job queue.

NO is true if the procedure is not run from the job queue.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if the procedure containing the statement is run from the job queue, the message `Procedure running` will be sent to the display station that placed the job on the job queue.

```
// IF JOBQ-YES MSG ?WS?,'Procedure running'
```

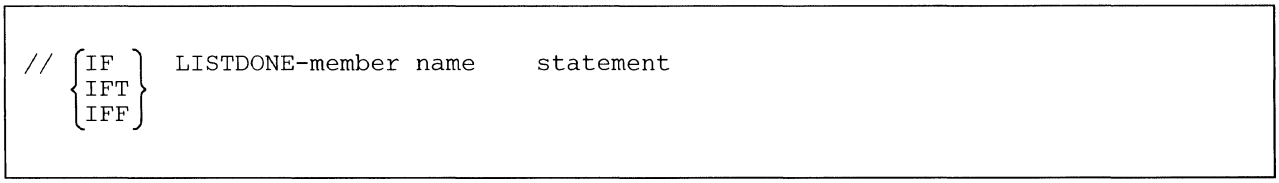
Example 2

This example specifies that if the procedure containing this statement is not run from the job queue, the message `Procedure running` will be displayed.

```
// IF JOBQ-NO * 'Procedure running'
```

LISTDONE (Phone List Completion) Condition

The LISTDONE conditional expression determines whether all the numbers in a phone list have been called. This condition is true if every number in the specified phone list has been called successfully or, for those numbers not called successfully, the retry count has been exhausted.



S9020023-0

member name specifies the name of a phone list that was created by the DEFINEPN or DEFINX21 procedure.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

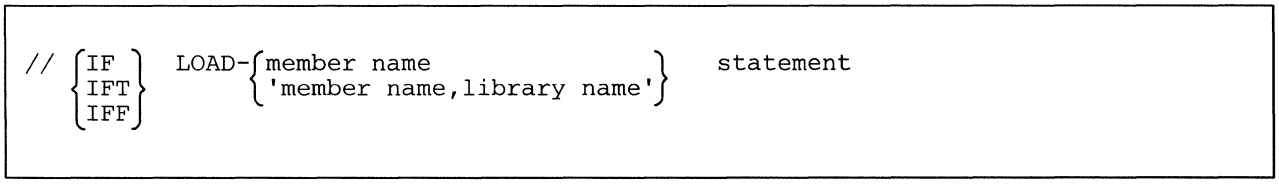
Example

In this example, the program CALLPROG will repeat itself, each time calling the next number in the list. When the last number has been called, or if all error retry counts have been exhausted, the IF expression will be true and the procedure will end.

```
// TAG START
// LOAD CALLPROG
// COMM LINE-1,PHONE-LIST1
// RUN
// IFF LISTDONE-LIST1 GOTO START
// * 'Program CALLPROG completed'
```

LOAD (Library Load Members) Condition

The LOAD conditional expression determines whether a library load member exists in a specified library. The condition is true if the specified load member exists in the specified library.



S9020024-0

member name specifies the name of the library load member to be searched for.

library name specifies the name of the library to be searched for the load member. If no library name is specified, the system library (#LIBRARY) is assumed.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

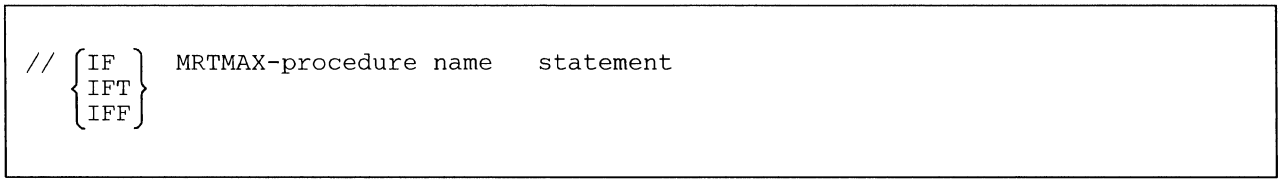
Example

This example specifies that if a load member named PROG1 exists in the current library (indicated by ?CLIB?), the REMOVE procedure should be run to remove that member.

```
// IF LOAD-'PROG1,?CLIB?' REMOVE PROG1,LOAD,?CLIB?
```

MRTMAX (Multiple Requesting Terminals) Condition

The MRTMAX conditional expression determines whether a multiple requester terminal (MRT) procedure has the maximum number of requester terminals. The condition is true if the specified MRT procedure has the maximum number of requesters attached.



S9020025-0

procedure name specifies the name of a MRT procedure.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

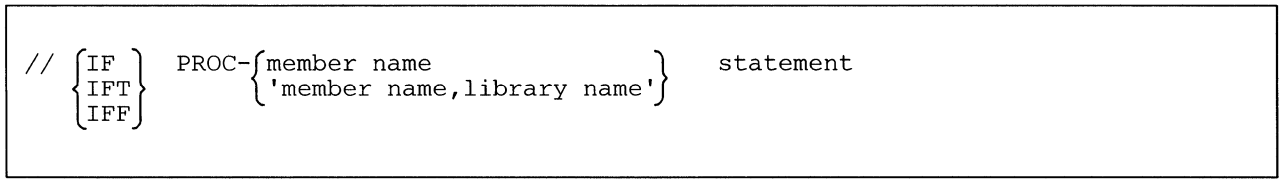
Example

This example specifies that if the ORDERS procedure has the maximum number of users attached, two messages are to be displayed, and a PAUSE statement and a CANCEL statement are to be processed.

```
// IFF MRTMAX-ORDERS GOTO START
// * 'Too many people are using the ordering procedure'
// * 'Canceling procedure; Try again later'
// PAUSE
// CANCEL
// TAG START
ORDERS
```

PROC (Library Procedure Members) Condition

The PROC conditional expression determines whether a library procedure member exists in a specified library. The condition is true if the specified procedure member exists in the specified library.



S9020026-0

member name specifies the name of the library procedure member to be searched for.

library name specifies the name of the library to be searched for the procedure member. If no library name is specified, the system library (#LIBRARY) is assumed.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example

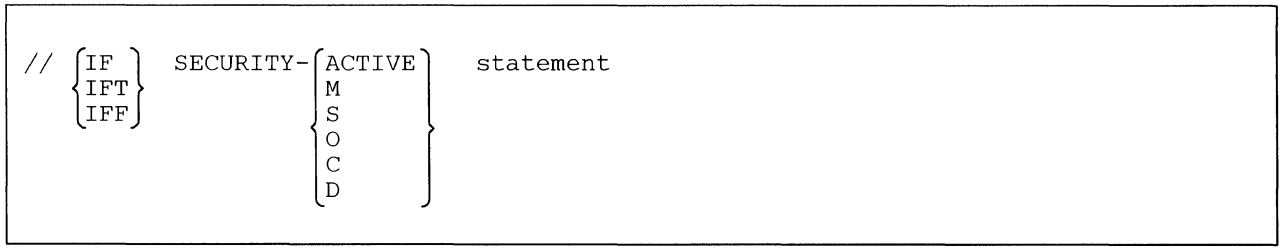
This example specifies that if a procedure member named TEST exists in the current library (indicated by ?CLIB?), the REMOVE procedure should be run to remove that member.

```
// IF PROC-'TEST,?CLIB?' REMOVE TEST,PROC,?CLIB?
```

SECURITY (Password Security) Condition

The SECURITY conditional expression determines the following:

- Whether password security is active on the system.
- The operator's security classification.



S9020027-0

ACTIVE is true if password security is active on the system.

M, S, O, C, and D specify security levels. The condition is true if the operator's security classification equals or exceeds the specified security classification. If password security is not active, all these tests will be true.

M Master security officer

S Security officer

O System operator

C Subconsole operator

D Display station operator

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if password security is not active, the message should be sent to the system console.

```
// IFF SECURITY-ACTIVE ** 'PROCEDURE ?PROC? BEING RUN BY ?USER?'
```

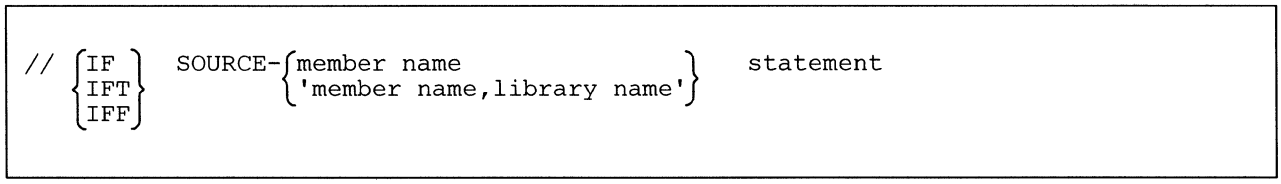
Example 2

This example specifies that this procedure can only be run if the operator has a security level of system operator, security officer, or master security officer. If the operator has a security level of subconsole operator or display station operator, the procedure is canceled. If password security is not active, the procedure must be run from the system console.

```
// IFF SECURITY-O CANCEL
//     ELSE IFF SECURITY-ACTIVE IF CONSOLE-NO CANCEL
// LOAD PROGRAM1
// RUN
```

SOURCE (Library Source Members) Condition

The SOURCE conditional expression determines whether a library source member exists in a specified library. The condition is true if the specified source member exists in the specified library.



S9020028-0

member name specifies the name of the library source member to be searched for.

library name specifies the name of the library to be searched for the source member. If no library name is specified, the system library (#LIBRARY) is assumed.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

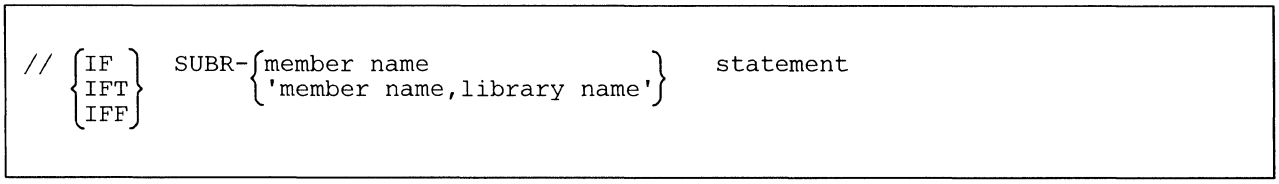
Example

This example specifies that if a source member named FORMATS does not exist in the current library (indicated by ?CLIB?), the SDA procedure should be run to create that member.

```
// IFF SOURCE-'FORMATS,?CLIB?' SDA FORMATS,?CLIB?
```

SUBR (Library Subroutine Members) Condition

The SUBR conditional expression determines whether a library subroutine member exists in a specified library. The condition is true if the specified subroutine member exists in the specified library.



S9020029-0

member name specifies the name of the library subroutine member to be searched for.

library name specifies the name of the library to be searched for the subroutine member. If no library name is specified, the system library (#LIBRARY) is assumed.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

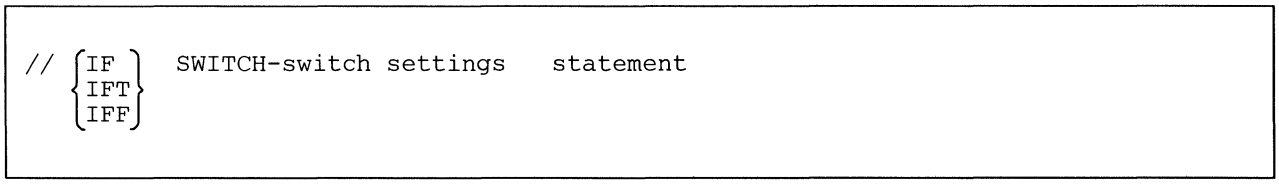
Example

This example specifies that if a subroutine member named PROG1 exists in the current library (indicated by ?CLIB?), the REMOVE procedure should be run to remove that member.

```
// IF SUBR-'PROG1,?CLIB?' REMOVE PROG1,SUBR,?CLIB?
```

SWITCH (Switches) Condition

The SWITCH conditional expression determines the settings of the user program status indicator (UPSI) switches. The condition is true if all eight of the switches for the display station are in the specified state.



S9020030-0

switch settings consist of 8 characters, one for each of the eight switches (1 through 8). For each of the eight positions in the switch settings value, one of the following characters must be used:

0 Zero indicates that the corresponding indicator must be off for the condition to be true.

1 One indicates that the corresponding indicator must be on for the condition to be true.

X Indicates that the corresponding indicator is not checked.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

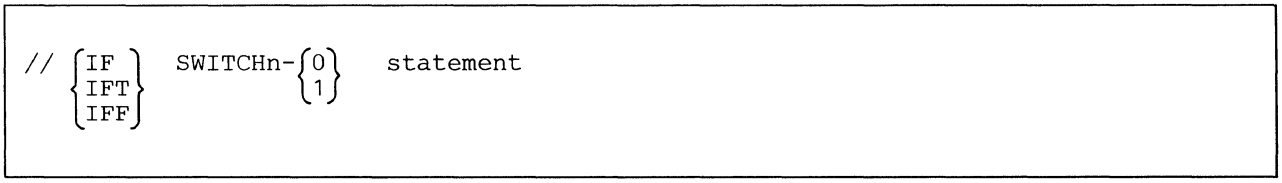
Example

This example specifies that if switches 1 and 2 are on and if switch 3 is off (the other switches are not checked), the procedure called MASTER should be run.

```
// IF SWITCH-110XXXXX MASTER
```

SWITCHn (Individual Switches) Condition

The SWITCHn conditional expression determines the setting of the nth switch.



S9020031-0

n is a number from 1 through 8 indicating which switch is to be tested.

0 Zero. The condition is true if the nth switch is off.

1 One. The condition is true if the nth switch is on.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

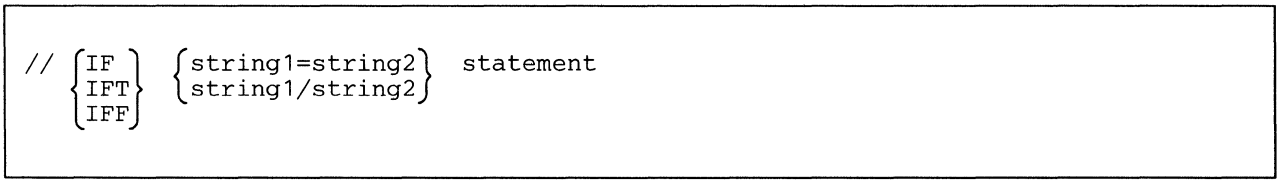
Example

This example specifies that if indicator 2 is off, it should be turned on.

```
// IF SWITCH2-0 SWITCH X1XXXXXX
```

string1=string2 (Comparing, Equal to) Condition

The = conditional expression determines whether one expression equals another expression. This condition is true if string1 is equal to string2.



S9020032-0

string1 and string2 represent two values, such as parameters, character data, or substitution expressions. Each character string can be up to 128 characters long. If the string contains blanks, equal signs (=), slashes (/), greater than signs (>), or hyphens (-), the string must be enclosed in apostrophes ('). If the string contains apostrophes, such as o'clock, you should enter two apostrophes; for example: 'o''clock'.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

If either of the two values being compared has any alphabetic characters or is enclosed by apostrophes ('), the two strings are padded on the right with blanks to a length of 128, and the comparison is done on a character-by-character basis, starting with the leftmost character.

If both of the values being compared have only numeric characters (and are not enclosed by apostrophes), the two strings are padded on the left with decimal zeros to a length of 128, and the comparison is done on a character-by-character basis, starting with the leftmost character. Numeric digits preceded by a + sign or - sign are considered numeric strings.

For example,

```
// IF ABCD='ABC ' PROCA
// IF 13=1        PROCB
```

would be compared as:

```
ABCD:     ABCD
ABC:      ABC
13:       000...0013
1:        000...0001
```

The ... indicates more zeros.

Numeric data can be either positive or negative, for example, +25 or -3. The + sign is not needed: for example, 25 is assumed to be +25. The value -0 is treated as 0. Character data preceded by a sign must be enclosed in quotes. For example: '+ABCDE'. If the first character of a string is a + or - sign, **all** of the remaining characters in the string must be numeric. If a sign appears in a string in any position but the first position, it is considered to be data. Other examples of how strings can be specified are shown below:

```
string1=string2

'string1'=string2

'string1'='string2'
```

Example 1

This example specifies that if the third positional parameter is PAYROLL, the procedure named PAYROLL should be run.

```
// IF ?3?=PAYROLL PAYROLL
```

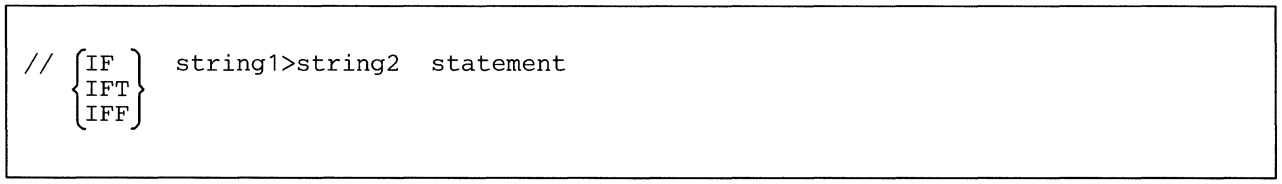
Example 2

This example specifies that if the first positional parameter does not have a value, the PROMPT OCL statement should be processed.

```
// IF ?1?/ PROMPT MEMBER-SCREENS,FORMAT-F1
```

string1>string2 (Comparing, Greater Than) Condition

The > conditional expression determines whether one expression is greater than another expression. This condition is true if string1 is greater than string2.



S9020033-0

string1 and string2 represent two values, such as parameters, character data, or substitution expressions. Each character string can be up to 128 characters long. If the string contains blanks, equal signs (=), slashes (/), greater than signs (>), or hyphens (-), the string must be enclosed in apostrophes ('). For information about how the strings are compared and how they can be entered, see "string1=string2 (Comparing, Equal to) Condition" on page 3-50.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example

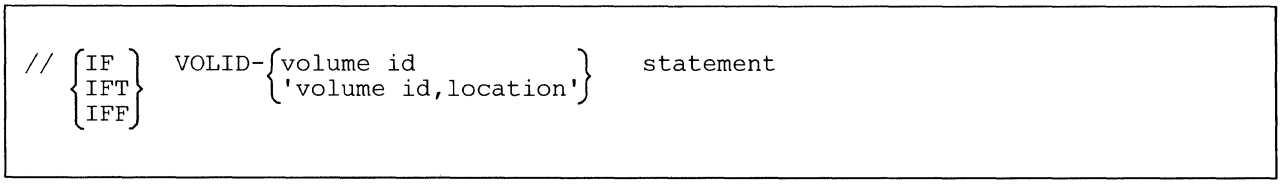
If the value of the fourth positional parameter is greater than 100, the procedure called MASTER should be run.

```
// IF ?4?>100 MASTER
```

VOLID (Diskette and Tape Volume IDs) Condition

The VOLID conditional expression determines whether the correct diskette or tape is being used. This condition is true if the volume ID of the specified diskette or tape is the same as the specified volume ID. For magazine drives: if more than one slot is being checked, the test is stopped when a match is found; therefore, not all slots may have been tested.

If the diskette drive or specified location is empty, or the tape drive is empty, a message will be displayed. The operator can then insert a diskette or tape into the specified location or cancel the job.



S9020034-0

volume id specifies the volume ID to be searched for.

location specifies the location of the diskette or tape to be checked if the system has a diskette magazine drive or tape drive. On a system without a diskette magazine drive, specifying any diskette location results in S1. It can be any of the following:

S1, S2, or S3 identifies an individual diskette slot. If a location is not specified, S1 is assumed.

ALLS, specifies that all slots (S1, S2, and S3) should be searched.

M1.nn identifies a location within magazine 1. **nn** can be any number from 01 through 10. For example, M1.04 indicates slot 4 within magazine 1.

M2.nn identifies a location within magazine 2. **nn** can be any number from 01 through 10.

ALL1 specifies that all diskettes in magazine 1 should be searched.

ALL2 specifies that all diskettes in magazine 2 should be searched.

ALL specifies that all diskettes in both magazines should be searched.

T1 specifies that the tape in tape drive 1 should be searched.

T2 specifies that the tape in tape drive 2 should be searched.

TC specifies that the tape in the tape cartridge drive should be searched.

statement specifies a statement to be processed. See the "Statement Portion of IF Conditional Expression" on page 3-54 for more information.

Example 1

This example specifies that if the volume ID of the inserted diskette is not VOL001, the message `Insert correct diskette in slot 1` should be displayed.

```
// IFF VOLID-VOL001 PAUSE 'Insert correct diskette in slot 1'
```

Example 2

This example specifies that if the volume ID of the diskette in location M2.03 is TEST, the information on the diskette should be removed.

```
// IF VOLID-'TEST,M2.03'  INIT TEST,,DELETE,M2.03
```

Example 3

This example specifies that if the volume ID of the tape mounted on tape drive 1 is TEST, the information on the tape should be removed.

```
// IF VOLID-'TEST,T1'  TAPEINIT T1,STDLABEL,TEST,,,ERASE
```

Statement Portion of IF Conditional Expression

The statement portion in an IF expression can be any of the following:

- An OCL statement without the // characters that normally precede an OCL statement.
- A statement that calls another procedure.
- A utility control statement without the // characters that normally introduce a utility control statement.
- Another IF, IFT, or IFF expression without the // characters that normally precede the expression. For example:

  ```
  // IF PROC-PAYROLL IF SWITCH3-1 PAYROLL
  ```

  specifies that if a procedure member named PAYROLL is in the system library and if indicator 3 is on, the PAYROLL procedure should be run.
- A // *, // **, CANCEL, EVALUATE, GOTO, PAUSE, RESET, or RETURN statement without the // characters that normally precede the statement.

ELSE Expressions

The ELSE expression is an optional expression that can be used only with the IF expression. Use the ELSE expression when you want to process a statement because an IF expression was not satisfied. The ELSE expression is processed only when the one or more IF expressions preceding the ELSE expression are not satisfied. The ELSE expression can be used with all forms of the IF expression (IF, IFT, and IFF).

The ELSE expression has the following format:

```
// ELSE  statement
```

S9020035-0

statement can be any statement that is valid for the IF conditional expression; see "Statement Portion of IF Conditional Expression" on page 3-54.

Example of IF and ELSE expressions:

```
// IF ?1?= RETURN
// ELSE DELETE ?1?
```

In this example:

- If the first parameter of the procedure is not entered, the RETURN statement is processed.
- If the first parameter on the procedure is entered (parameter 1 is not blank), the DELETE procedure should be run to delete the file specified by the first parameter.

There can be only one ELSE expression after the IF expression and it must be the first expression in that line. If the ELSE expression does not immediately follow the IF expression, the ELSE expression is ignored. An IF expression can follow an ELSE expression in a conditional statement. For example:

```
// IF ?2?= PAYROLL DAILY
// ELSE IF ?2?=YEAREND PAYROLL YEAREND
//      ELSE PAYROLL WEEKLY
```

In this example, if the second parameter is blank, the PAYROLL procedure with the parameter DAILY is called. If the second parameter is YEAREND, it is passed to PAYROLL and the PAYROLL procedure is run. If the second parameter is neither blank nor YEAREND, the parameter WEEKLY is passed to PAYROLL and PAYROLL is run.

You can also have conditional expressions in which more than one IF expression (in the same line) comes before an ELSE expression, as in the following:

```
// IF SWITCH1-1 IF SWITCH2-1 PROCA
// ELSE PROCB
```

In this case, procedure PROCA will only be run when both switch 1 and switch 2 are on. If either switch 1 or switch 2 is off (that is, if either of the tests are false), procedure PROCB is run.

Continuation of a statement following an ELSE expression onto two or more lines will make the continuation expression incorrect. For example:

```
// IF SWITCH1-1 PROCA
//    ELSE FILE NAME-WORK,
//         UNIT-F1
```

will result in an error when switch 1 is on because the system will process the two statements:

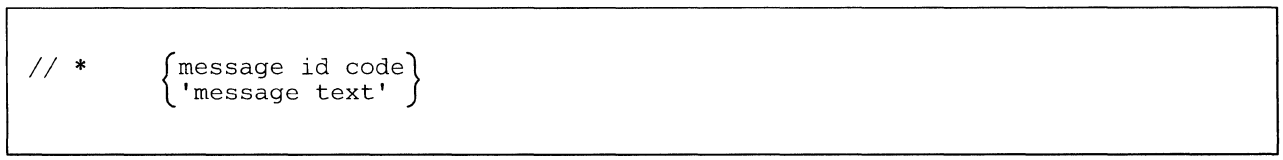
```
// PROCA
//         UNIT-F1
```

which will cause an error.

// * (Informational Message) Statement

The // * statement causes an informational message to be displayed at the display station that submitted the job unless the // * statement is in a job run from the job queue or a job that released its requester display station. In those cases, the // * statement causes a message to be displayed at the system console.

This type of message is not displayed if INFOMSG NO was previously entered. See the "INFOMSG Control Command" on page 6-20 for more information.



S9020036-0

message id code specifies the 4-digit message identification code of a message in the current first-level message member. The text of the specified message number is displayed. See the "MEMBER OCL Statement" on page 5-73 for information on assigning a first-level message member. For information about creating message members, see the "CREATE Procedure" on page 4-132.

'message text' specifies the message text to be displayed. The message text must be enclosed in apostrophes ('). Any character can be used in the message text. If the message contains embedded apostrophes (such as the apostrophe in o'clock), then enter each embedded apostrophe as two apostrophes (for example: o''clock).

The message text can be up to 505 characters. The entire message is displayed, 75 characters at a time. The message is listed 75 characters at a time in the history file.

The message text can contain ideographic characters. If an attempt is made to send ideographic characters to a display station that cannot display ideographic characters, the IGC characters will be replaced by periods.

Example 1

In the following example, the message indicates that the PAYROLL procedure is running.

```
// * 'PAYROLL Procedure Running'
// LOAD PAYROLL
// RUN
```

Example 2

In the following example, the message indicates that the PAYROLL procedure is running; however, the message is displayed only when the procedure is being run from the keyboard.

```
// IF JOBQ-NO IF EVOKED-NO * 'PAYROLL Procedure Running'
// LOAD PAYROLL
// RUN
```

Example 3

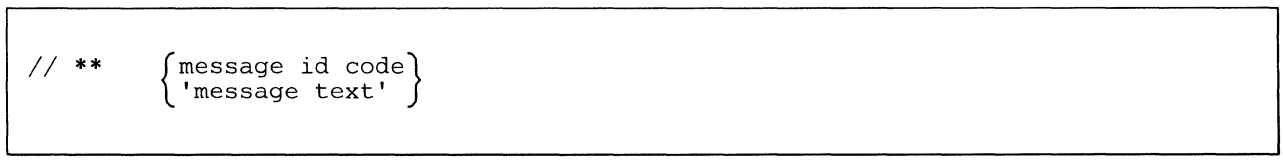
In the following example, the message indicated by number 0005 is displayed from the first-level message member named MESSAGES.

```
// MEMBER USER1-MESSAGES,LIBRARY-MYLIB
// * 0005
// LOAD PAYROLL
// RUN
```

// ** (System Console Message) Statement

The // ** statement causes a message to be displayed on the console display at the system console. The job stops processing until the system operator responds to the message. The message is displayed with a 0 option response.

If the procedure containing the // ** statement is run from the display station assigned as the system console, the message is displayed on the command display at the system console.



S9020037-0

message id code specifies the 4-digit message identification code of a message in the current first-level message member. The text of the specified message is displayed. See the "MEMBER OCL Statement" on page 5-73 for information on assigning a first-level message member. For information about creating message members, see the "CREATE Procedure" on page 4-132.

'message text' specifies the message text to be displayed. The message text must be enclosed in apostrophes ('). Any character can be used in the message text. If the message contains embedded apostrophes (such as the apostrophe in o'clock), then enter each embedded apostrophe as two apostrophes (for example: o''clock).

The message text can be up to 504 characters. The entire message is displayed, 75 characters at a time, with each message having a 0 option response. The message is listed 75 characters at a time in the history file.

The message text can contain ideographic characters. If an attempt is made to send ideographic characters to a display station that cannot display ideographic characters, periods will be displayed for the ideographic characters.

Example

To cause the message `Insert the next diskette` to appear on the system console display, your procedure can contain:

```
SAVE FILE1,,,VOL001
// ** 'Insert the next diskette'
SAVE FILE2,,,VOL002
```

CANCEL Statement

The CANCEL statement cancels a procedure and returns control to the keyboard for the next statement. If a CANCEL statement is used to cancel an MRT procedure, an error message is issued to all active requesters of that procedure.

```
// CANCEL
```

S9020038-0

Example

This example shows the CANCEL statement being used to cancel a procedure when an IF expression is not satisfied:

```
// IFF DATAF1-FILEA CANCEL
```

EVALUATE Statement

The EVALUATE statement allows you to do the following:

- Assign values to a parameter
- Perform addition, subtraction, multiplication, and division of values
- Evaluate substitution expressions
- Set the return code

More than one item can be evaluated at a time.

The EVALUATE statement can contain any type of data. It can contain arithmetic expressions, substitution expressions, or comments.

You can use the EVALUATE statement as the result of a series of conditional tests. This allows your conditions to be easier to follow. For example:

```
// IF DATAF1-FILEA   IF SWITCH1-1   EVALUATE (EVERYTHING IS FINE)
//    ELSE  CANCEL
PROCA
```

If the tests are true, the EVALUATE and the PROCA statements would be processed; otherwise, the CANCEL statement would be processed. Note the comment EVERYTHING IS FINE on the EVALUATE statement. Anything that is not recognized as an expression is considered to be a comment.

Assigning Values

You can assign more than one parameter on the EVALUATE statement, but each parameter being assigned must be separated by one or more blanks. The expressions are evaluated from left to right. The results of one expression can be used in a following expression.

```
// EVALUATE  Pn[,length]=expression
```

S9020039-0

P indicates a parameter is assigned.

n indicates the number of the procedure parameter (1 through 64) into which the expression is saved. **n** can be either a number or a substitution expression with a numeric value.

length is an optional value that specifies the length (number of characters) of the resulting expression. The length can be a number from 1 through 15. If the length is not specified, the length of the expression is used; any leading zeros are not used in the final result. The length can be specified only for numeric values. If a length is specified and the number of significant digits in the result is greater than the specified length, the result is truncated on the left. For example:

```
P3,2=12345
```

causes parameter 3 to be set to '45'.

If a length is specified and the number of significant digits in the result is smaller than the specified length, the result is padded on the left with zeros. For example:

```
P3,5=123
```

causes parameter 3 to be set to '00123'.

If no length is specified, the length of the expression is used. For example:

```
P3=876253
```

causes parameter 3 to be set to '876253'.

For negative numbers, a place should be reserved for the minus sign (-). For example:

```
// EVALUATE P5,2=-12-1
```

parameter 5 will contain 13 instead of -13 because no room was allowed for the minus sign.

expression indicates an expression. The expression can be an integer constant, a substitution expression having the value of an integer, a substitution expression having a value from a statement in a message member, or a character string. The character string can contain IGC characters. If the character string contains any embedded blanks, it must be enclosed in apostrophes ('); for example: P5='ab c'. To set a parameter off, enter 2 consecutive apostrophes (''); for example: P5=''.

Example 1

This example assigns the value 123 to parameter 1, the value 456 to parameter 2, and sets parameter 3 off. Both parameters 1 and 2 have a length of 3; parameter 3 has a length of 0.

```
// EVALUATE P1=123 P2=00456 P3=''
```

Example 2

This example assigns the value 00123 to parameter 1. Parameter 1 has a length of 5.

```
// EVALUATE P1,5=123
```

Example 3

This example assigns the value 'EXAMPLE DATA' to parameter 1. Parameter 1 has a length of 12.

```
// EVALUATE P1='EXAMPLE DATA'
```

Example 4

If the operator has a user ID of MAT, this example assigns the value MAT to parameter 1. Parameter 1 has a length of 3.

```
// EVALUATE P1=?USER?
```

Example 5

If parameter 3 has a value of 123, parameter 9 has a value of 30, and parameter 10 has a value of 5, this example assigns the value 00123 to parameter 30. Parameter 30 has a length of 5.

```
// EVALUATE P?9?,?10?=?3?
```

would be evaluated as:

```
// EVALUATE P30,5=123
```

Adding, Subtracting, Multiplying, or Dividing

You can assign more than one parameter on the EVALUATE statement, but each parameter being assigned must be separated by one or more blanks. The parameters being assigned are evaluated from left to right.

```
// EVALUATE  Pn[,length]=expression
```

S9020040-0

P indicates a parameter is assigned.

n indicates the number of the procedure parameter (1 through 64) into which the expression is saved. **n** can be either a number or a substitution expression with a numeric value.

length is an optional value that specifies the length (number of characters) of the resulting expression. The length can be a number from 1 through 15. If the length is not specified, the length of the expression is used; any leading zeros are not used in the final result. For more information about the length parameter, see "Assigning Values" on page 3-61.

expression indicates one or a combination of the standard arithmetic operations of addition, subtraction, multiplication, or division. These expressions are made up of numbers (integer constants or a substitution expression having the value of an integer), signs (+, -, * for multiplication, and / for division), and, optionally, left and right parentheses to group operations. Characteristics of such arithmetic expressions, and the rules for constructing them, are as follows:

The minus (-) sign can indicate a negative number as well as the subtraction operation. For example:

```
// EVALUATE P1=-4+-5
```

is valid and assigns -9 to parameter 1.

Blanks are not allowed within one assignment expression, for example:

```
// EVALUATE P1 = 1 + 2
```

is not valid; but:

```
// EVALUATE P1=1+2
```

is valid.

Multiplication and division operations have a higher priority than addition and subtraction operations. In other words, multiplication and division are performed before addition and subtraction in an expression.

A left parenthesis indicates the start of an algebraic operation, and a right parenthesis indicates the end of an algebraic operation. Operations grouped by parentheses are performed before operations not grouped by parentheses, regardless of their relative priorities. Up to 30 operations can be grouped within the entire expression.

When division is performed, remainders are dropped. No rounding is performed. To obtain the remainder, multiply the quotient by the original divisor, and then subtract that result from the dividend. For example:

```
// EVALUATE P1=17/5 P2=17-(?1?*5)
```

assigns the quotient of the division operation to parameter 1 and the remainder to parameter 2.

If you attempt to divide by zero, an error message is issued.

The result and each operand of the expression are limited to 15 digits. If you attempt to evaluate an expression with a result greater than 15 digits, the results will be unpredictable.

Example 1

This example assigns the value 246 to parameter 1 and the value -1 to parameter 2. Parameter 1 has a length of 3 and parameter 2 has a length of 2.

```
// EVALUATE P1=123+123  P2=0001+2-4
```

Example 2

This example assigns the value 00246 to parameter 1. Parameter 1 has a length of 5.

```
// EVALUATE P1,5=123+123
```

Example 3

If parameter 3 has a value of 123, parameter 9 has a value of 30, and parameter 10 has a value of 5, this example assigns the value 00125 to parameter 30. Parameter 30 has a length of 5.

```
// EVALUATE P?9?,?10?=?3?+2
```

would be evaluated as:

```
// EVALUATE P30,5=123+2
```

Example 4

If parameter 1 has a value of 3 and parameter 2 has a value of 7, this example assigns the value 25 to parameter 3. Because multiplication and division operations have a higher priority than addition and subtraction operations, the value in parameter 1 is multiplied by the value in parameter 2 first. Then 4 is added to the result.

```
// EVALUATE P3=4+?1?*?2?
```

Example 5

If parameter 1 has a value of 5, this example assigns the value 30 to parameter 2. Because it is grouped by parentheses, the addition operation is performed before the multiplication operation.

```
// EVALUATE P2=3*(?1?+5)
```

Example 6

This example shows how to create a file based upon the size of two work files. Files CUSTMST and WORK are two resident files that contain several records.

```
// EVALUATE P64=?F'A,CUSTMST'?+?F'A,WORK'?
BLDFILE NEWFILE,S,RECORDS,?64?,100
```

For example, if file CUSTMST contained 200 records and file WORK contained 150 records, the statements would be evaluated as:

```
// EVALUATE P64=200+150
BLDFILE NEWFILE,S,RECORDS,350,100
```

Evaluating Substitution Expressions

```
// EVALUATE  expression
```

S9020041-0

expression can be one or more substitution expressions. Each expression must be separated by one or more blanks. The expressions are evaluated from left to right. The results of one expression can be used in a following expression.

Example 1

This example assigns the value ABC to parameter 1 only if parameter 1 does not have a value.

```
// IF ?1?/ EVALUATE ?1'ABC'?
```

Example 2

This example assigns the value MYLIB to parameter 3 if parameter 3 does not have a value and assigns the value PROC to parameter 2 if parameter 2 does not have a value.

```
// EVALUATE ?3'MYLIB'? ?2'PROC'?
```

Setting the Return Code

You can save the return code from one job, run a second job, and then set the return code value back to the value it was for the first job.

```
// EVALUATE  CD=nnnn
```

S9020042-0

CD indicates that the return code is to be restored.

nnnn indicates the 4-digit return code value to be restored.

Example

In this example, the first EVALUATE statement saves the return code value for program 1, and the second EVALUATE statement restores the return code value for program 3.

```
// LOAD PROG1
// RUN
* Save return code for PROG3
// EVALUATE P64,4=?CD?
// ALLOCATE UNIT-I1,WAIT-NO
* This will change return code if diskette drive is unavailable
// LOAD PROG2
// RUN
* Restore return code from PROG1
// EVALUATE CD=?64?
// LOAD PROG3
// RUN
```

GOTO and TAG Statements

The GOTO and TAG statements allow you to branch around groups of statements within a procedure. By using GOTO and TAG statements instead of several IF expressions, you can reduce the number of expressions that the SSP must process during procedure generation.

```
// GOTO label
```

```
// TAG label
```

S9020043-0

A GOTO statement causes the SSP to branch to the statement following a TAG statement with the same label. (More than one TAG statement in a procedure can have the same label. The system branches to the next TAG statement with the required label.) The SSP searches for the target TAG statement within the procedure. The search begins with the statement following the GOTO statement. If the SSP does not find the target statement before it reaches the end of the procedure, the SSP goes to the first statement in the procedure and resumes the search.

The following shows an example of GOTO and TAG statements:

```
* Example of GOTO and TAG statements
// GOTO A
  .
  .      These statements are skipped
  .
// TAG A     Processing resumes after this statement
  .
  .
  .
```

The word TAG cannot be substituted, for example:

```
// EVALUATE P1=TAG
// GOTO A
  .
  .
// ?1? A
```

is not valid.

Example

A procedure named PROC runs four procedures: STEP1, STEP2, STEP3, and STEP4. Normally, to start PROC, the operator enters:

```
PROC
```

If the operator cancels PROC at some point, the procedure can be restarted at the canceled step by entering:

```
PROC RESTART,step name
```

step name is STEP1, STEP2, STEP3, or STEP4.

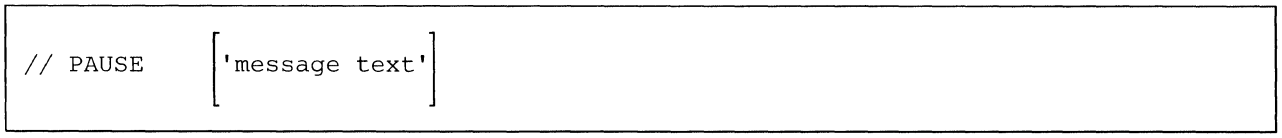
The following are the statements in PROC:

```
// IF ?1?= IF ?2?=  GOTO STEP1        (STEP1 IS THE DEFAULT)
// IFF ?1?=RESTART  * 'INCORRECT ENTRY'
// IFF ?1?=RESTART  CANCEL
// IF  ?2?=STEP1    GOTO STEP1
// IF  ?2?=STEP2    GOTO STEP2
// IF  ?2?=STEP3    GOTO STEP3
// IF  ?2?=STEP4    GOTO STEP4
// * 'INCORRECT ENTRY'
// CANCEL
// TAG STEP1
STEP1
// TAG STEP2
STEP2
// TAG STEP3
STEP3
// TAG STEP4
STEP4
```

PAUSE Statement

The PAUSE statement stops the processing of a job and displays a message at the display station that submitted the job. If the PAUSE statement is in a job run from the job queue or a job that released its requester display station, the PAUSE statement displays a message at the system console. The operator can then restart the job by selecting option 0. The system then continues processing the statements that follow the PAUSE statement.

Although the message length can be up to 500 characters, only the first 68 characters are displayed. Also, only the first 113 characters are written to the history file.



S9020044-0

'message text' specifies the message to be displayed when the job pauses. The message must be contained within single apostrophes ('). If the message text contains embedded apostrophes (such as the apostrophe in o'clock), then enter the embedded apostrophes as two apostrophes (for example: 'o''clock').

The message text can contain IGC characters. If an attempt is made to run a procedure that contains an ideographic message following a PAUSE statement at a display station that cannot display ideographic characters, the IGC characters will be replaced by periods.

If no message text is specified, the following message is displayed:

```
PAUSE -- when ready, enter 0 to continue
```

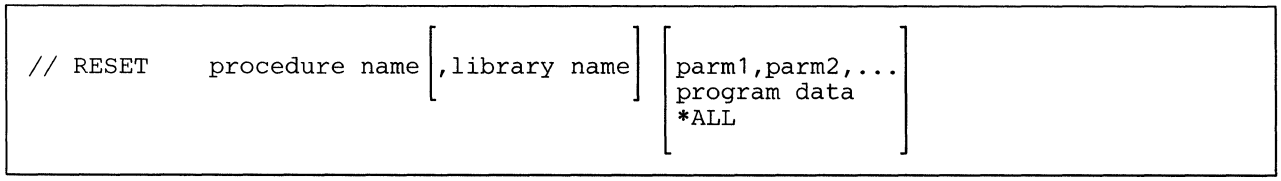
Example

In this example, the PAUSE statement informs the operator that the payroll program is complete. The operator must enter a 0 (zero) to continue.

```
// LOAD PAYROLL
// FILE NAME-EMPLOY
// RUN
// PAUSE 'Payroll program is complete'
```

RESET Statement

You can have up to 255 procedure levels. However, for some applications, many more levels of procedure calls are desired. For example, a procedure might call itself an undetermined number of times. The RESET statement calls a procedure and specifies that it should be treated as a first-level procedure. That is, when the procedure called by the RESET statement completes, control is not returned to the procedure level containing the RESET statement.



S9020045-0

procedure name specifies the name of the procedure being called.

library name specifies the name of the library containing the procedure.

parm1,parm2,... are the optional procedure parameters. The procedure specified by procedure name becomes a first-level procedure. (Control is not returned to the procedure containing the RESET statement.) A maximum of 64 parameters can be specified. See the "INCLUDE OCL Statement" on page 5-63 for more information about how parameters can be specified.

program data is the data to be passed to a program on its first input operation from the display station.

***ALL** specifies that all 64 parameters from the current procedure level are to be passed to the specified procedure.

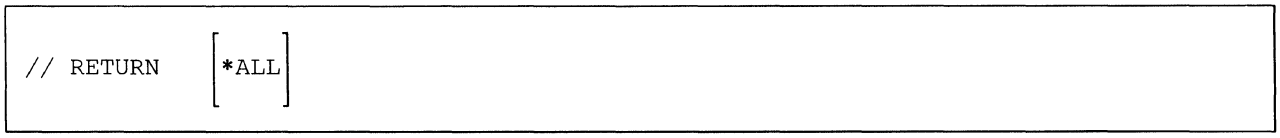
Example

A procedure called PROCA displays a list of options to an operator and tells the operator to enter one of the options. PROCA then calls the procedure specified by the selected option. When that procedure completes, PROCA calls itself and the process is repeated. In this application, PROCA must be able to call itself any number of times. Following are the statements in the procedure member called PROCA.

```
* THIS IS PROCEDURE PROCA
// * 'ENTER ONE OF THE FOLLOWING OPTIONS:'
// * '  1. RUN WEEKLY PAYROLL'
// * '  2. PRINT CURRENT INVENTORY'
// * '  3. RUN WEEKLY BILLING'
// * '  4. STOP'
// IF ?1R?=4 CANCEL
// IF ?1?=1  PAYROLL WEEKLY
// IF ?1?=1  RESET PROCA
// IF ?1?=2  INVENT
// IF ?1?=2  RESET PROCA
// IF ?1?=3  BILLING WEEKLY
// IF ?1?=3  RESET PROCA
// * '?1? IS A WRONG OPTION'
// RESET PROCA
```

RETURN Statement

If the RETURN statement is processed in a first-level procedure, the procedure will return to the requesting display station for the next statement if further input is required. If further input is required and a requesting display station does not exist (for example, the procedure is run from the job queue), an error message is displayed. If further input is not required, the job will terminate. If it is processed in a nested procedure, the RETURN statement returns to the calling procedure for the next statement.



S9020046-0

If *ALL is specified, all 64 procedure parameters from the current procedure level are passed to the previous procedure level. This causes all the parameters for the previous procedure to be completely replaced.

If *ALL is not specified, the parameters for the previous procedure level are left unchanged.

Example 1

This example shows how to end a procedure using a RETURN statement so that the system does not waste time processing comment statements.

```
// LOAD PROGA
// FILE NAME-INPUT
// FILE NAME-OUTPUT
// RUN
// RETURN
* This procedure reads a file named INPUT, and
* creates a file named OUTPUT.
```

Example 2

This example shows how parameters are passed from procedure PROC2 back to the procedure that called PROC2.

```
* Procedure PROC2
// EVALUATE P1=FILEA P2=INPUT P3=OUTPUT
// RETURN *ALL
```

Chapter 4. Procedures

This chapter describes the IBM-supplied System/36 procedures. The procedures that are used to install and verify the program products are not described in this manual. See the manual *Changing Your System Configuration* for information on those procedures.

The following information is given for each procedure described in this chapter:

- The function of the procedure.
- The syntax format of the procedure command that calls the procedure. For a description of the rules used for syntax formats, see "Conventions Used for Describing Syntax Formats" on page 1-3.
- Descriptions of the parameters used in the procedure command.
- One or more examples of how to use the procedure and how to enter the procedure command or statements to run the procedure.

When you are signed on to the System/36, you can use the system help support to run these procedures. The system help support is made up of menus, prompt displays, and help text. The menus allow you to select a task you want to perform. When you select an item from a menu, either:

- Another menu is displayed (as you select options, each one gets more specific about the task you want to perform)
- A prompt display for a procedure or command is shown

The prompt displays allow you to run procedures to do the task. The help text explains the menus, the menu options, the procedures and commands, and the parameters for the procedures and commands.

For more information about the system help support, see the "HELP Procedure" on page 4-199.

Procedures

The following chart shows the name of the SSP utility program or the name of the OCL statement that is run or processed by the SSP procedures that use utilities or OCL statements.

SSP Procedure	SSP Utility Run	OCL Statement Processed
ALERT	$CZUT	
ALOCFLDR	$TMSERV	
ALOCLIBR	$MAINT	
ALTERCOM	$SETCF	
ARCHIVE	$TMSERV	
BLDFILE	$FBLD	
BLDINDEX	$FBLD	
BLDLIBR	$MAINT	
BLDMENU	$BMENU	
	$MAINT	
	$MGBLD	
BUILD	$BUILD	
CACHE	$SVCASRV	
CATALOG	$LABEL	
CHNGEMEM	$MAINT	
COMPRESS	$FREE	
CONDENSE	$MAINT	
	$TMSERV	
COPYDATA	$COPY	
COPYDIAG	$DCOPY	
COPYI1	$DUPRD	
COPYPRT	$UASF	
	$UASC	
CREATE	$MGBLD	
DATE		DATE
DEFINEID	$IDSET	
DEFINEPN	$PNLM	
DEFINX21	$XNLM	
	$XNSH	
DELETE	$DELET	
DELNRD	$SINDL	
DISABLE	$IEDS	
EDITNRD	$SINR	
ENABLE	$IENBL	
ERR	$CPPE	
FORMAT	$SFGR	
FROMLIBR	$MAINT	
HELP	$HELP	
HISTORY	$HIST	
INIT	$INIT	
INITDIAG	$INIT	
JOBSTR	$MAINT	JOBQ
KEYSORT	$DDST	
LIBRLIBR	$MAINT	
LINES		FORMS
LISTDATA	$COPY	
LISTFILE	$BICR	
	$COPY	
	$MAINT	
	$TCOPY	
	$TMSERV	
LISTLIBR	$MAINT	
LISTNRD	$LABEL	
LOG		LOG
MOVEFLDR	$TMSERV	
NOHALT		NOHALT
PASSWORD	$PRPWD	
POST	$POST	
PRINT		FORMS
PRINTKEY	$SETCF	
PRTGRAPH	$DPGP	
REMOVE	$MAINT	
RENAME	$RENAM	
REQUESTX	#GCFR	
RESPONSE	$ARSP	
RESTEXTN	$XREST	
RESTFLDR	$TMSERV	
RESTLIBR	$MAINT	
RESTNRD	$COPY	
	$SINCT	
RESTORE	$COPY	
RETRIEVE	$TMSERV	
SAVE	$COPY	
SAVEEXTN	$XSAVE	
SAVEFLDR	$TMSERV	
SAVELIBR	$MAINT	
SAVENRD	$COPY	
SECDEF	$PRUID	
	$RRESC	

SSP Procedure	SSP Utility Run	OCL Statement Processed
SECEDIT	$PRCED	
	$PRUED	
	$RREDT	
	$RRTED	
SECLIST	$PRCLT	
	$PRLST	
	$RRLST	
	$RRTLT	
SECREST	$PRURS	
	$RRSTR	
SECSAVE	$PRUSV	
	$RRSAV	
SET	$SETCF	
SETALERT	$ARSP	
SETCOMM	$SETCP	
SLIB		LIBRARY
STARTM	$MMST	
STOPM	$MMSP	
SWITCH		SWITCH
SYSLIST		SYSLIST
TAPECOPY	$TCOPY	
TAPEINIT	$TINIT	
TAPESTAT	$FEAIDS	
TOLIBR	$MAINT	
TRANSFER	$BICR	

#STRTUP1 Procedure

You can create the #STRTUP1 procedure to start your own jobs that need to be run during the initial program load (IPL) process **before** other jobs are run. #STRTUP1 is a reserved procedure name that is searched for during the IPL. If the procedure is found, it begins running; if it is not found, no error occurs. No parameters can be passed to the procedure.

The #STRTUP1 procedure is meant to be run on a dedicated system; that is, when no other programs are running. Because this procedure is run before the IPL has completed, certain system functions such as spooling and the job queue have not yet been initialized. Using this procedure, you should not try to run any programs that print output or start other programs.

CAUTION

You must ensure that no procedure you include in the #STRTUP1 procedure contains a never-ending program (NEP) or a multiple requester terminal (MRT) program.

To make your own procedures, see Chapter 2, "Making Your Own Procedures." You can place the #STRTUP1 procedure in the system library (#LIBRARY) or in the system operator's sign-on library.

The following figure shows the relationship between the IPL process and procedure #STRTUP1. #STRTUP2 is another procedure that can be run as part of the IPL process; see the "#STRTUP2 Procedure" on page 4-6.

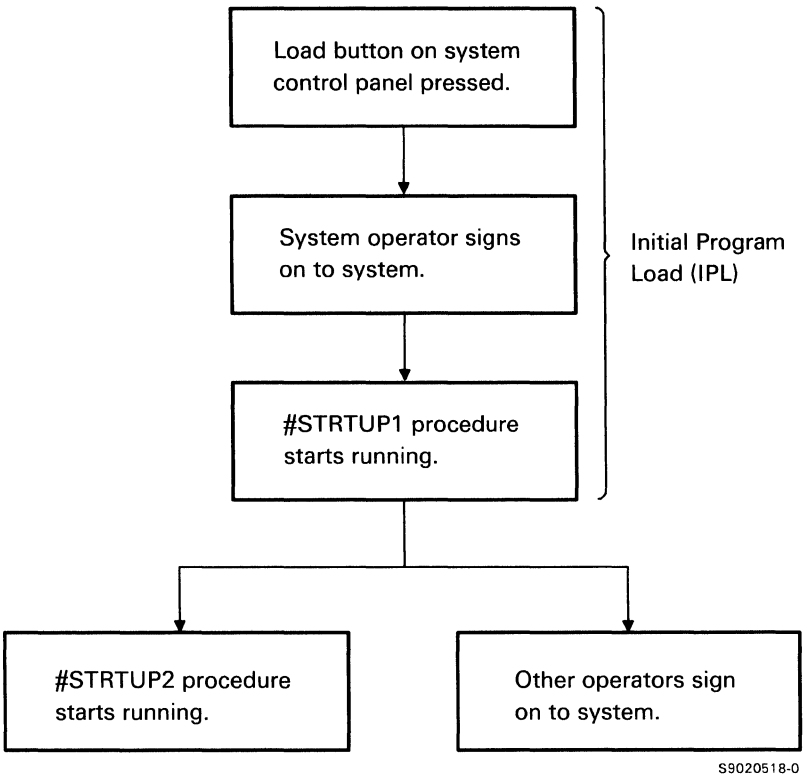


Figure 4-1. Relationship between IPL, #STRTUP1, and #STRTUP2

Example

This example shows three procedures you could include in the #STRTUP1 procedure, which you could store in the system library (#LIBRARY).

```
* #STRTUP1 procedure
* Set the system default automatic response severity level to 3
NOHALT 3,SYSTEM
*
* Condense the system library
CONDENSE #LIBRARY
*
* Compress the disk
COMPRESS
```

#STRTUP2 Procedure

You can create the #STRTUP2 procedure to start your own jobs that need to be run as part of the initial program load (IPL) process **while** other jobs are run. #STRTUP2 is a reserved procedure name that is searched for during IPL. If the procedure is found, it begins running; if it is not found, no error occurs. Other jobs can be started while this job runs, and other operators are allowed to sign on. No parameters can be passed to the procedure.

You can use this procedure to do tasks you want done each time IPL is performed, but you do not want to prevent other jobs from starting while this job runs.

You can place the #STRTUP2 procedure in the system library (#LIBRARY) or in the system operator's sign-on library.

Figure 4-1 on page 4-5 shows the relationship between IPL and procedure #STRTUP2. #STRTUP1 is another procedure that can be run as part of the IPL process; see the "#STRTUP1 Procedure" on page 4-4.

Example

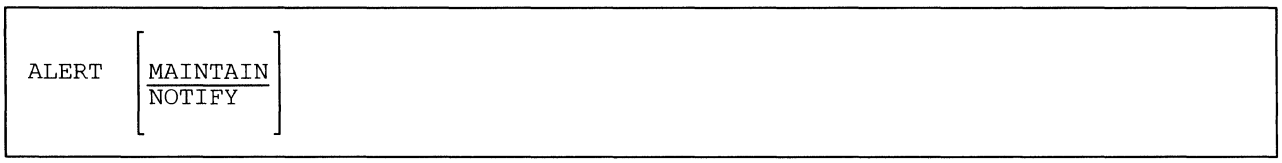
This example shows a procedure you could include in the #STRTUP2 procedure, which you could store in the system library (#LIBRARY).

```
* #STRTUP2 procedure
* List the contents of the disk
CATALOG
*
* Enable a subsystem
ENABLE SUBSYS1
```

ALERT Procedure

The ALERT procedure starts the alert support, which allows you to specify which system messages in a predefined subset of system messages should generate alert messages and provides the interface for sending a notification alert. A notification alert allows you to generate your own message and send it to the remote location designated to receive alerts. Alert messages are sent to the designated location of a communications network via the advanced program-to-program communications (APPC) subsystem or the advanced peer-to-peer networking (APPN) subsystem. For more information about alert messages, see the *Communications and Systems Management Guide*.

The ALERT procedure runs the $CZUT utility program but requires some special setup in the local data area for the parameters; therefore, the user should use the ALERT procedure instead of writing his or her own procedure.



S9020047-1

MAINTAIN specifies that maintenance of the system message member or alert source members is to be performed. If no parameter is specified, MAINTAIN is assumed.

NOTIFY specifies that a notification alert is to be created and sent to the remote location designated to receive alerts.

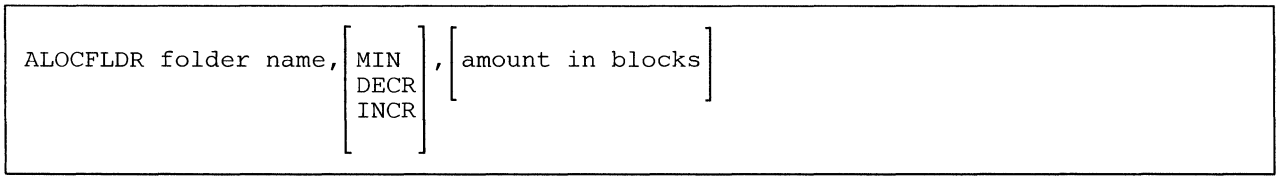
ALOCFLDR Procedure

The ALOCFLDR procedure allows you to reorganize a folder and change its size.

When the folder is reorganized, the folder members are moved together at the front of the folder to reduce as much as possible the number of folder extents. This may improve performance because the folder makes better use of disk space.

You can also increase or decrease the size of a folder, or make it as small as possible. The folder size will not change if you specify ALOCFLDR with *folder name* and no other parameters. The INCR and DECR options are not valid for Personal Services/36 mail log folders. For more information about folder reorganization, see the *Concepts and Programmer's Guide*.

The ALOCFLDR procedure runs the $TMSERV utility program.



S9020476-0

folder name specifies the folder that is to be reorganized and, optionally, have its size changed.

MIN specifies that the folder is to be made as small as possible, containing only the folder members. This is the same as running the CONDENSE procedure with the FOLDER parameter specified.

DECR specifies that the folder is to be decreased by the amount specified. If DECR is specified, an amount in blocks must also be specified.

INCR specifies that the folder is to be increased by the amount specified. If INCR is specified, an amount in blocks must also be specified.

amount in blocks specifies the number of blocks that the folder is to be increased or decreased. You can specify a number from 1 through 6553. One block contains 10 sectors. If an amount is specified, DECR or INCR must also be specified. If an amount is not specified, the folder is reorganized but does not change in size.

Example 1

This example shows a folder named MYFLDR being reorganized without changing its size.

```
ALOCFLDR MYFLDR
```

Example 2

This example shows the size of a folder named MYFLDR being increased by 10 blocks.

```
ALOCFLDR MYFLDR,INCR,10
```

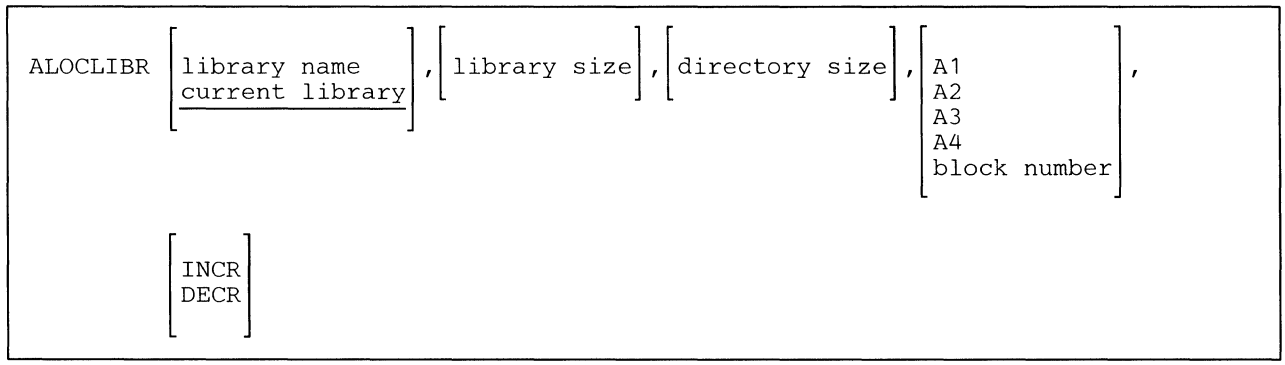
ALOCLIBR Procedure

The ALOCLIBR procedure allows you to increase or decrease the size of a user library or the system library (#LIBRARY). You can also change the size of a user library's directory, or change the disk unit that contains the user library. If the library has an extent (a separate area on disk that contains library members that would not fit in the library), the extent is combined with the library. If the library does not have an extent, the size of the library, the size of the directory, or the disk unit location must be changed. For the system library (#LIBRARY), the directory size and the disk unit or location cannot be specified.

When the ALOCLIBR procedure is running, no other jobs or display stations can use the specified library. If an attempt is made to run the ALOCLIBR procedure for a library that is being used by another job, an error message is displayed and the ALOCLIBR procedure is not run. Once ALOCLIBR begins running, no other jobs are allowed to use the specified library until ALOCLIBR is complete.

You can use the LISTLIBR procedure or the CATALOG procedure to determine whether a library has an extent. See the "LISTLIBR Procedure" on page 4-272 or the "CATALOG Procedure" on page 4-73 for more information. For guidelines for determining the sizes of libraries and directories, see the *Concepts and Programmer's Guide*.

To create a new library, see the "BLDLIBR Procedure" on page 4-62. The ALOCLIBR procedure runs the $MAINT utility program.



S9020048-0

library name specifies the library that is to be changed. If no parameter is specified, the current library is assumed. To change the size of the system library, enter #LIBRARY.

library size specifies the new size, in blocks, of the library. If INCR or DECR is specified, this indicates by how many blocks the library size is to be increased or decreased. The maximum library size is 15000 blocks. One block contains 2560 bytes. If no library size is specified, and an extent exists, the size of the library is increased only by the amount needed to contain the members in the extent. If no library size is specified and no extent exists, the library size is not changed, unless the size of the directory is increased enough to require the library size to increase.

directory size specifies the new size, in sectors, of the directory for the library. If INCR or DECR is specified, this indicates how many sectors the library directory is to be increased or decreased. The minimum directory size is 2 sectors; the maximum directory size is 2500 sectors. One library block contains 10 sectors. If no directory size is specified, the directory size is not changed.

To determine the number of available directory entries for a given number of sectors, multiply the number of sectors by 5 and subtract 7 from that number. For example, for 10 directory sectors, you could have up to 43 entries (5 x 10 = 50, 50 - 7 = 43).

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, another unit (if it exists) is checked. If no location is specified and the system has more than one disk unit, the library is placed on the least used disk unit.

block number specifies the location of the first block of the library. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk.

INCR indicates that the library size and/or directory size are to be increased by the amounts specified. Either a library size, a directory size, or both must be specified.

DECR indicates that the library size and/or directory size are to be decreased by the amounts specified. If no library or directory size is specified, the library and directory will be made as small as possible.

Example 1

This example shows a library named MYLIB being changed to 250 blocks. The size of the directory is being changed to 25 sectors.

```
ALOCLIBR MYLIB,250,25
```

Example 2

This example shows the size of a library named MYLIB being increased by 50 blocks. The size of the directory is being increased by 25 sectors.

```
ALOCLIBR MYLIB,50,25,,INCR
```

Example 3

This example shows the size of a library named MYLIB being decreased by 10 blocks.

```
ALOCLIBR MYLIB,10,,,DECR
```

ALTERBSC Procedure

This procedure is supported only for compatibility with the IBM System/34. The procedure you should use on the System/36 is the "ALTERCOM Procedure" on page 4-11.

ALTERCOM Procedure

The ALTERCOM procedure changes communications items for a specified communications line. Changes only apply to programs run from the same display station as the ALTERCOM procedure. To change items so that all the display stations are affected, see the "SETCOMM Procedure" on page 4-461.

You can use the STATUS COMM command to display the current status of the communications items that you can change with the ALTERCOM procedure.

Following is a chart that indicates the ALTERCOM prompts that can be used for the named subsystems and lines.

ALTERCOM	BSC Subsystem	SNA Subsystem	RWS	Asynchronous	Batch BSC
Line type	X	X	X	X	X
Switch type	X	X	X	X	X
Remote switched line ID					X
Local switched line ID					X
Tributary address					X
Compress/truncate blanks					X
BSC wait time					X
3740 multiple file mode					X
Record separator					X
Modem speed	X	X	X		X
Switched network backup	X	X	X		X
BSC error retries					X
Primary SDLC time-out		X	X		
SDLC error retries		X	X		
Secondary SDLC time-out		X			

If any parameters are omitted, the corresponding values in the display station communications configuration record are not changed.

The ALTERCOM procedure can be run from any display station.

If a communications subsystem is to be enabled and the ALTERCOM procedure is required, you must run the ALTERCOM procedure from the same display station as the ENABLE procedure. If you run ALTERCOM while a subsystem is enabled, the subsystem must be disabled and enabled again before the new values will be used.

ALTERCOM

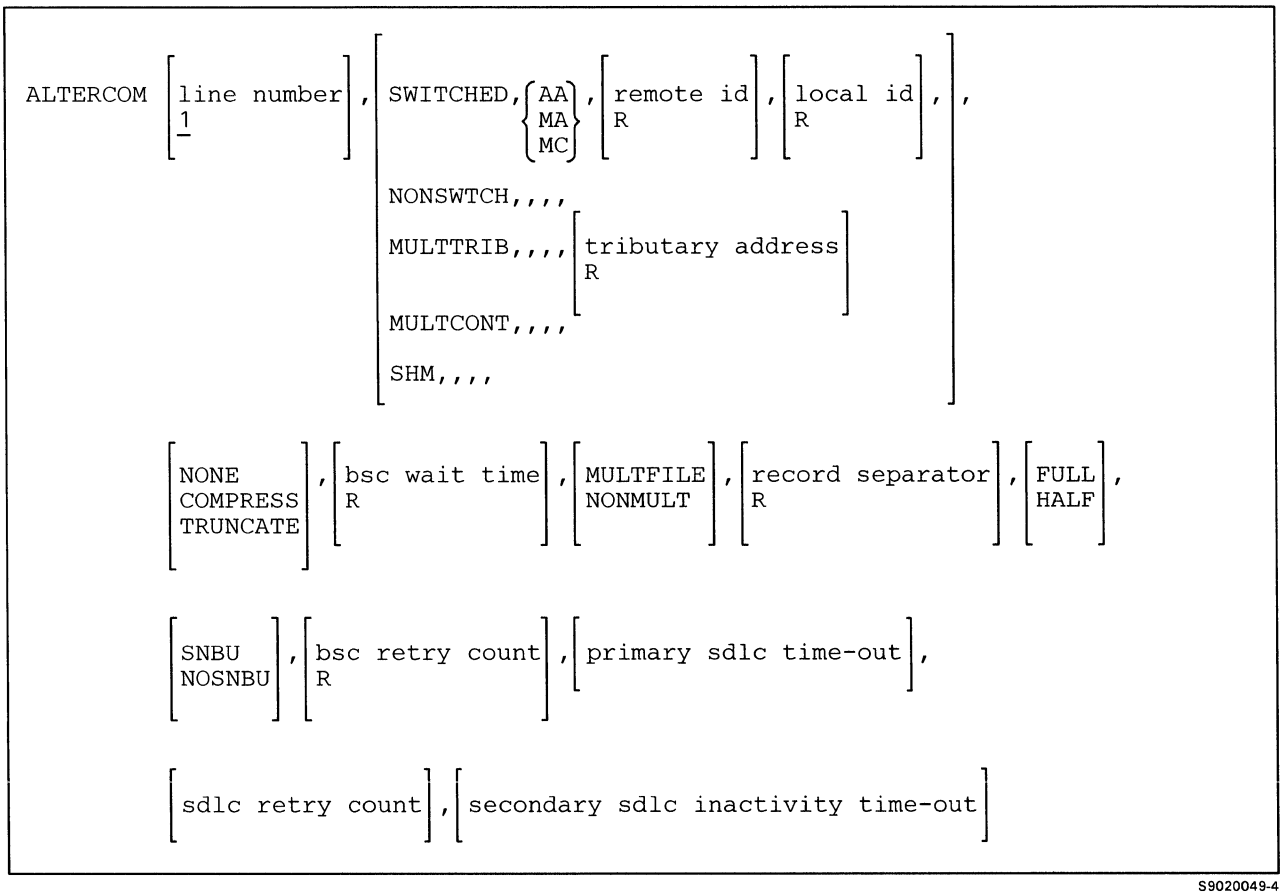
The ALTERCOM procedure cannot be run from the job queue or started by an EVOKE OCL statement.

Changes made by ALTERCOM remain in effect until:

- The items are changed again by the ALTERCOM procedure or the $SETCF utility program.
- The system library is restored by an IPL from diskette.
- The system is configured again.
- An IPL after the SETCOMM procedure was run.

See the manual *Using System/36 Communications* for more information about communications subsystems. See the manuals *Programming with RPG II* and *Programming with Assembler* for information about batch BSC data communications.

The ALTERCOM procedure runs the $SETCF utility program.



S9020049-4

Although each parameter is optional, at least one parameter, other than line number, must be specified.

line number specifies the number of the communications line that is to have one or more of its characteristics changed. If a line number is not entered, line number 1 is assumed.

SWITCHED specifies that the line is a point-to-point, switched communications line.

AA specifies that if the communications modem is in automatic answer mode, the system is to automatically answer the call.

MA specifies that the system operator is to manually answer the call.

MC specifies that the system operator is to manually call the device to be communicated with. For autocall, specify MC and have the phone list to be used specified in the subsystem's configuration.

remote id specifies the hexadecimal value of the remote station ID. Two, four, six, or eight hexadecimal digits can be specified. If the ID of the remote station is longer than eight hexadecimal digits, the ID must be specified in the program.

R specifies that the remote ID is to be reset (set to hex 00).

local id specifies the hexadecimal value of the local station ID. Two, four, six, or eight hexadecimal digits can be specified. If the ID of the local station is longer than eight hexadecimal digits, the ID must be specified in the program.

R specifies that the local ID is to be reset (set to hex 00).

NONSWTCH specifies that the line is to be a point-to-point nonswitched communications line.

MULTTRIB specifies that the system is a multipoint tributary station.

tributary address specifies the address of the system on the multipoint communications line. The tributary address is the hexadecimal value of one of the pair of tributary station addressing characters. For example, to indicate the EBCDIC addressing characters SS, the value E2 would be entered (E2 is the hexadecimal equivalent of the EBCDIC letter S). If no address is specified, the value in the program is used.

R specifies that the tributary address is to be reset (set to hex 00).

See "Tributary Station Addressing and Polling Characters" on page 4-17 for the tributary addressing and polling characters.

MULTCONT specifies that the system is a multipoint control station. This parameter cannot be used with BSC.

SHM specifies that the line uses X.21 short hold mode. See the "DEFINX21 Procedure" on page 4-141 for more information. See the manual *Using System/36 Communications* for more information on short hold mode.

NONE specifies that neither blank compression nor truncation is to be performed.

COMPRESS specifies that embedded blanks are to be compressed.

TRUNCATE specifies that trailing blanks are to be truncated (that is, cut off).

bsc wait time specifies the number of seconds that BSC is to wait for the System/36 user program to issue a BSC request before it indicates that an error has occurred. You can specify any decimal number from 1 through 999.

R specifies that the BSC wait time is to be reset (set to hex 00).

MULTFILE specifies that more than one (3740-type) file can be transmitted or received.

NONMULT specifies that multiple 3740-type files cannot be transmitted or received.

record separator specifies the character to be used to indicate the end of one record and the beginning of another when multiple records per block are sent. All values are valid except 00, 01, 02, 03, 10, 1F, 26, 2D, 32, 37, or 3D. See the manual *Programming with RPG II* or the manual *Programming with Assembler* for more information about record separator characters.

R specifies that the record separator is to be reset to hex 00.

- For RPG II programs, if this parameter is omitted and the COMPRESS or TRUNCATE parameters are specified, a record separator of hex 1E is used. Otherwise, no record separator is used.
- For Assembler programs, the following occurs:
  - If a record separator is specified in the program, that record separator is used.
  - If COMPRESS, TRUNCATE, or variable length records is specified and no record separator is specified in the program, a record separator of hex 1E is used; otherwise no record separator is used.

FULL specifies that the full rated speed of the modem is to be used.

HALF specifies that half of the full rated speed of the modem is to be used. This is only valid if the modem supports half rate.

SNBU specifies that the switched network backup line is to be used if a failure occurs on the nonswitched primary line.

This is only valid if the modem and the remote system support switched network backup.

If SNBU is specified for BSC, and manual call, manual answer, or automatic answer is not specified, the connection either becomes manual call or manual answer depending on the first communications line operation done by the program. If the first line operation is a transmit operation, manual call is assumed. If the first line operation is a receive operation, manual answer is assumed.

NOSNBU specifies that no switched network backup line is to be used.

bsc retry count specifies the number of times a transmission is to be attempted if an error occurs. You can specify any decimal number from 1 through 255. If you specify a retry count of 255, BSC tries indefinitely for a response from a remote system.

R specifies that the BSC retry count is to be reset (set to hex 00).

primary sdlc time-out value specifies the primary sdlc time-out value in half-second increments. You can specify a number from 05 through 80, where the first digit indicates the seconds and the second digit represents the tenths of seconds and must be 0 or 5. For example, 25 indicates two and one-half seconds and 30 indicates 3 seconds.

sdlc retry count specifies the number of retries of a transmission that are to be attempted. You can specify a number from 1 through 5. The number indicates the multiple of seven retries that are attempted; for example, 2 means 14 retries.

secondary sdlc inactivity time-out value specified the time period that determines when the primary system is to be considered no longer active for a secondary sdlc nonswitched line. You can specify a number from 1-20 to indicate the number of 32-second multiples for the timer, or specify 0 to indicate that no secondary inactivity timer should be used.

Example 1

Specify that line 1 is to be switched and automatic answer:

```
ALTERCOM 1,SWITCHED,AA
```

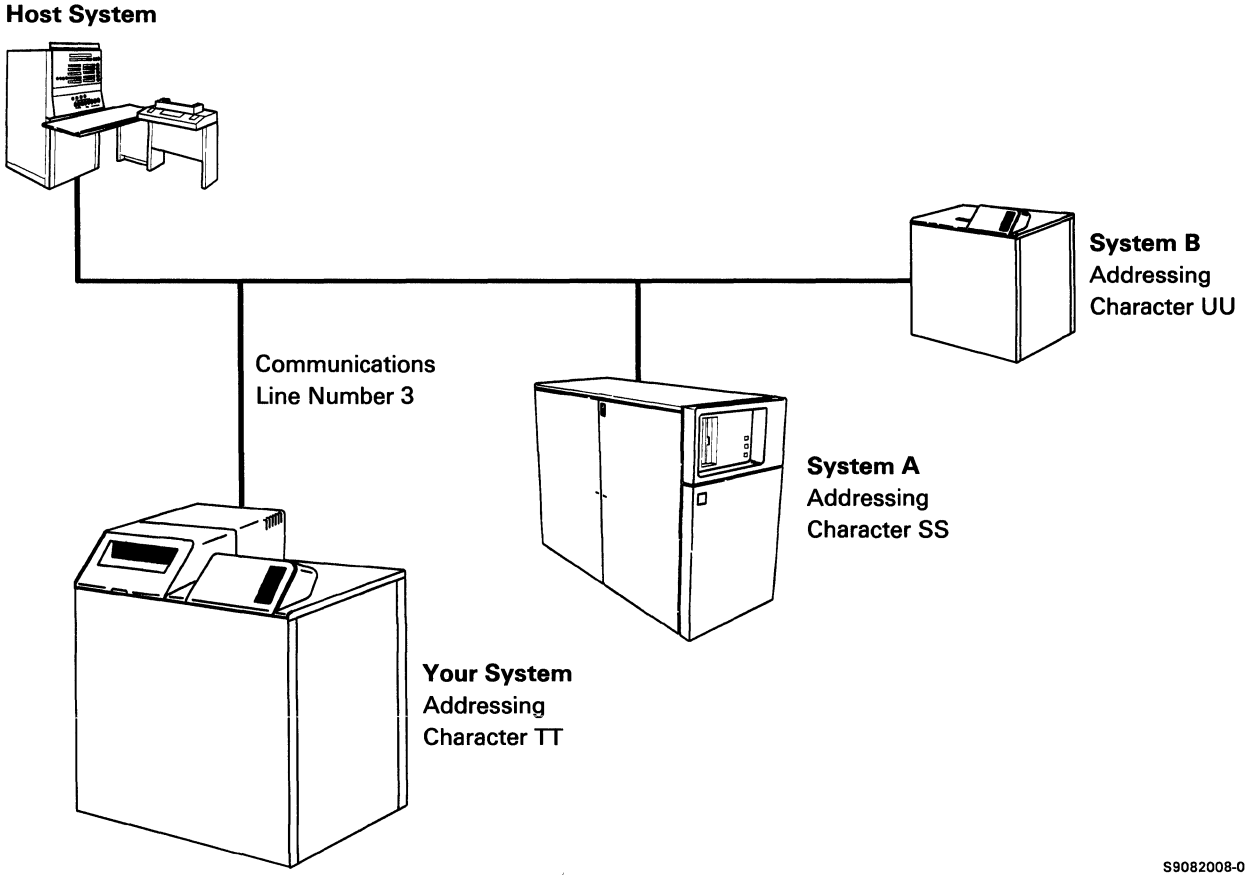
Example 2

This example shows how to change the following:

- The line number is 3
- The tributary address for the multipoint line is 'TT' (EBCDIC E3)

```
ALTERCOM 3,MULTTRIB,,,,E3
```

The multipoint communications network could be viewed as the following:



Tributary Station Addressing and Polling Characters

The following charts show the addressing and polling characters that can be used for System/36 BSC tributary stations. Addressing and polling characters must be used together in certain pairs. That is, once an addressing character is selected, the complimentary polling character must be used. The addressing and polling characters are assigned by the host system.

EBCDIC Addressing and Polling Characters

Addressing Character	Hex Code	Polling Character	Hex Code
SS	E2	BB	C2
TT	E3	CC	C3
UU	E4	DD	C4
VV	E5	EE	C5
WW	E6	FF	C6
XX	E7	GG	C7
YY	E8	HH	C8
ZZ	E9	II	C9
11	F1	JJ	D1
22	F2	KK	D2
33	F3	LL	D3
44	F4	MM	D4
55	F5	NN	D5
66	F6	OO	D6
77	F7	PP	D7
88	F8	QQ	D8
99	F9	RR	D9

ASCII Addressing and Polling Characters

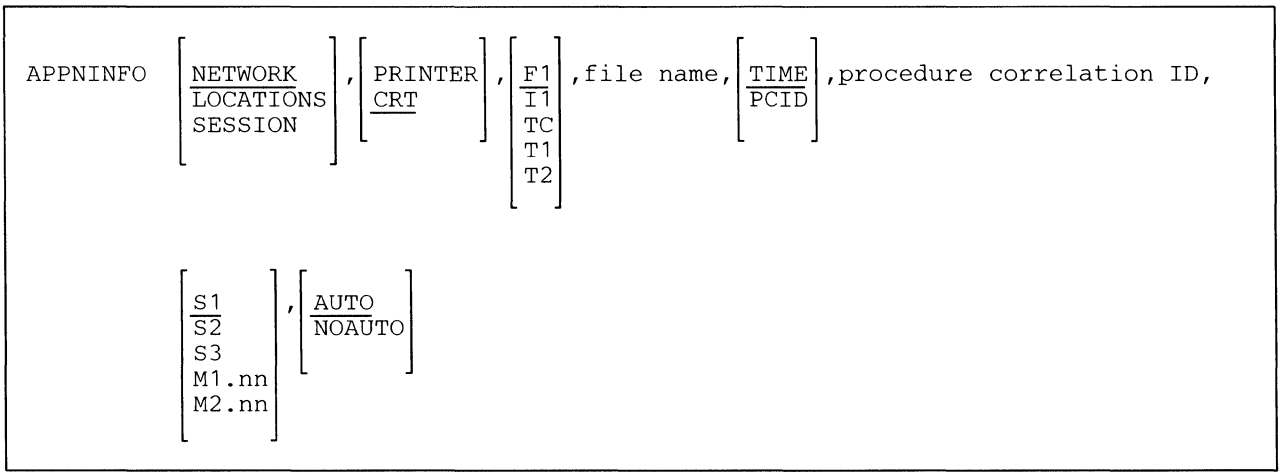
Addressing Character	Hex Code	Polling Character	Hex Code
aa	61	AA	41
bb	62	BB	42
cc	63	CC	43
dd	64	DD	44
ee	65	EE	45
ff	66	FF	46
gg	67	GG	47
hh	68	HH	48
ii	69	II	49
jj	6A	JJ	4A
kk	6B	KK	4B
ll	6C	LL	4C
mm	6D	MM	4D
nn	6E	NN	4E
oo	6F	OO	4F
pp	70	PP	50
qq	71	QQ	51
rr	72	RR	52
ss	73	SS	53
tt	74	TT	54
uu	75	UU	55
vv	76	VV	56
ww	77	WW	57
xx	78	XX	58
yy	79	YY	59
zz	7A	ZZ	5A

ALTERSDL Procedure

This procedure is supported only for compatibility with the IBM System/34; and does a function similar to the ALTERCOM procedure. The procedure you should use on the System/36 is the "ALTERCOM Procedure" on page 4-11.

APPNINFO Procedure

The APPNINFO procedure provides the user access to information needed for network problem determination. The types of data available to the user are session-related data, the network configuration information, and the directory of locations. See the manual *Advanced Peer-to-Peer Networking Guide* for more information.



S9020553-2

NETWORK specifies that the user wants to examine the configuration of nodes and links in the network. You can display an ordered list of the nodes and then select any node and look at the links attached to that node.

LOCATIONS specifies that the user wants to examine the directory of locations known to the networking node.

SESSION specifies that the user wants to examine the session-level data that is captured by the system. This data is logged to a disk file on a continuous basis so that the information will be accessible for problem determination purposes when a failure occurs.

PRINTER specifies that the output is to be listed on the session printer.

CRT specifies that the output is to be displayed at the display station.

If SESSION was specified as the first parameter, the following entries are valid:

F1, I1, TC, T1, or T2 specifies the file location. F1 indicates disk, I1 indicates diskette, TC indicates tape cartridge, T1 indicates tape drive 1, and T2 indicates tape drive 2. If F1 is not specified, the file must be saved on diskette or tape using the SAVE procedure.

file name specifies the name of the file to be printed or displayed. If using the SAVE procedure, to save a file to diskette or tape, the file name then refers to the name of the saved file on diskette or tape. If this parameter is not specified, the file name defaults to a system-created file (#LSSTLOG).

TIME specifies that entries are to be listed by time in reverse order of occurrence, beginning with the most recent entry.

PCID specifies that the entries are to be sorted and listed in ascending order by procedure correlation identifier.

procedure correlation ID specifies the name by which the System/36 uniquely identifies a session at session initiation. The PCID is found on a dump or a trace entry. This parameter is only valid with CRT and PCID. Only the entries for the specified ID will be listed.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. This parameter is valid only if I1 is specified for the file location.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is valid only if I1 is specified for the file location.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.
- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

Example

This example shows how to access session-level data and have it displayed at the display station.

```
APPNINFO SESSION,CRT
```

ARCHIVE Procedure

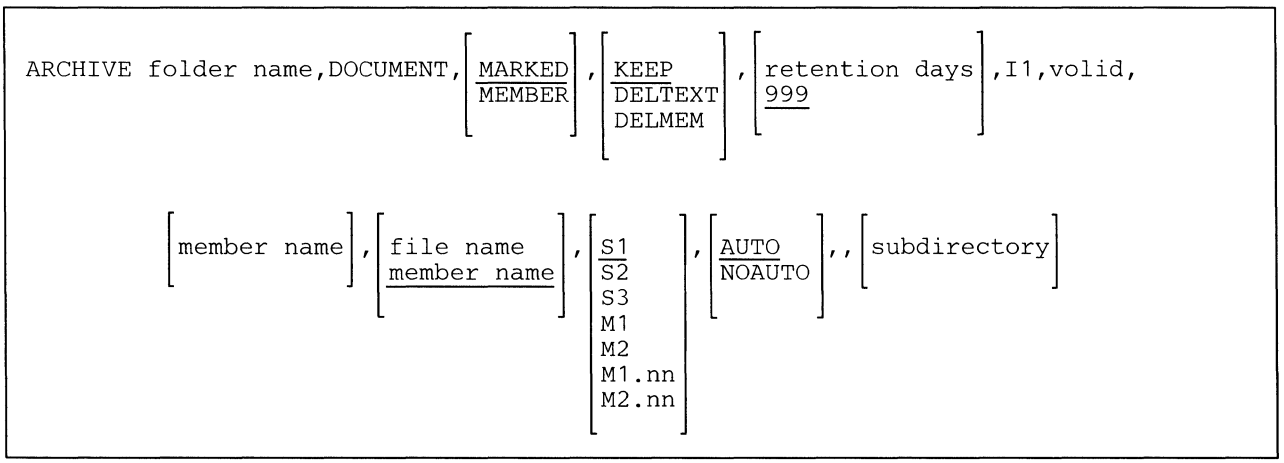
The ARCHIVE procedure copies a folder member or all marked folder members onto disk, diskette, tape, or tape cartridge. The folder member is stored in a file, and the folder directory is updated to indicate where it was copied (disk, diskette, tape, or tape cartridge); the volume ID of the diskette, tape or tape cartridge; the name of the file that received the copy; and the date when the member was copied. The ARCHIVE procedure can leave the member in the folder, remove the text from the folder and leave only the directory, or remove the entire member from the folder after the member is archived.

The ARCHIVE procedure can copy a folder member that has already been archived. The system creates a new file to contain the member, and the folder directory is updated.

If the member to be archived was created with a subdirectory, this subdirectory must be specified on the ARCHIVE procedure.

The ARCHIVE procedure runs the $TMSERV utility program.

To archive a folder member to diskette:



S9020050-1

To archive a folder member to tape or tape cartridge:

```
ARCHIVE folder name,DOCUMENT,[MARKED]  ,[KEEP    ]  ,[retention days]  ,[T1]  ,volid,
                             [MEMBER]  ,[DELTEXT ]  ,[999           ]  ,[T2]
                                       ,[DELMEM  ]                     ,[TC]

      [member name]  ,[file name  ]  ,,[AUTO  ]  ,[REWIND]  ,[subdirectory]
                     [member name]    [NOAUTO]  ,[LEAVE ]
                                               ,[UNLOAD]
```

S9020051-2

To archive a folder member to disk:

```
ARCHIVE folder name,DOCUMENT,[MARKED]  ,[KEEP    ]  ,,F1,,[member name]  ,
                             [MEMBER]  ,[DELTEXT ]
                                       ,[DELMEM  ]

      [file name  ]  ,,,,[subdirectory]
      [member name]
```

S9020562-2

folder name specifies the name of the folder from which the member or members are to be archived.

DOCUMENT specifies that the type of member to be archived is a document. If no parameter is specified, DOCUMENT is assumed.

MARKED specifies that all members that are marked to be archived are archived. If no parameter is specified, MARKED is assumed.

Note: If you are archiving MARKED members to tape, duplicate file names with the same date may exist on the tape. This is because there may be duplicate member names within subdirectories in the same folder. You will have to use OCL statements and sequence numbers to retrieve a member that exists in a tape file that has the same name and date as another tape file.

MEMBER specifies that only the member specified in the eighth parameter is to be archived.

KEEP specifies that the member is not to be removed from the folder after the member is archived. If no parameter is specified, KEEP is assumed.

DELTEXT specifies that the text portion of the member is to be removed from the folder after the member is archived. The directory portion of the member remains in the folder.

DELMEM specifies that the entire member (text and directory) is to be removed from the folder after the member is archived.

retention days specifies how long (in days) the diskette or tape file is to be retained. This number can be any number from 0 through 999. If a retention period is not specified, 999 days are assumed. If a retention period of 999 days is specified, the file is a permanent file. Retention days cannot be specified if archiving to a disk file. For more information on diskette, tape, or tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

I1 specifies that the file containing the archived member is to be created on diskette. If no parameter is specified, I1 is assumed.

F1 specifies that the file containing the archived member is to be created on disk.

T1, T2, or TC specifies that the file containing the archived member is to be created on tape or tape cartridge, and specifies which tape drive will receive the copy of the member.

volid specifies the volume ID of the diskette, tape, or tape cartridge. From 1 through 6 alphameric characters can be specified. Volid cannot be specified if you are archiving to a disk file.

member name specifies the name of the folder member that is to be archived. A valid member name is from 1 to 12 characters. It is made up of a 1 to 8 character document name and may optionally be followed by a period and a 1 to 3 character extension. This parameter is valid only if the third parameter is MEMBER.

file name specifies the name of the file to be created on disk, diskette, tape, or tape cartridge. If the file is located on diskette, tape or tape cartridge, the file name can be from 1 through 12 alphameric characters. If the file is located on disk, the file name must be from 1 through 8 characters long. The first character must be an alpha character, #, @, or $. The remaining characters may be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. Do not use ALL as a file name.

This parameter is valid only if a single member is to be archived and the third parameter is MEMBER. If the third parameter is MARKED, each file created will have the same name as the archived member. If the third parameter is MEMBER and a file name is not specified, the single file created will have the same name as the archived member.

If MEMBER is specified without a file name or MARKED is specified and the member name is not a valid disk file name, you will be prompted for a valid file name.

S1, S2, or S3 specifies the diskette slot containing the diskette to receive the copy of the member. This parameter is valid only if the sixth parameter is I1. If no parameter is specified and the sixth parameter is I1, S1 is assumed. This parameter cannot be specified if archiving to a disk file.

M1.nn or M2.nn specifies the magazine location containing the first diskette to receive the copy of the member. This parameter is valid only if the sixth parameter is I1. This parameter cannot be specified if archiving to a disk file.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored since there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

AUTO cannot be specified if you are archiving to a disk file.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored since there is only one tape cartridge drive.

NOAUTO cannot be specified if you are archiving to a disk file.

REWIND specifies that the tape should be rewound after the member is archived. This parameter is valid only if T1, T2, or TC is specified. If no parameter is specified, REWIND is assumed. REWIND cannot be specified if you are archiving to a disk or diskette file.

LEAVE specifies that the tape should be left where it is after the member is archived. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is valid only if T1, T2 or TC is specified. LEAVE cannot be specified if you are archiving to a disk or diskette file.

UNLOAD specifies that the tape should be rewound and unloaded after the member is archived. This parameter is not valid if either I1 or F1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing. UNLOAD cannot be specified if you are archiving to a disk or diskette file. UNLOAD is not valid for a tape cartridge and will be interpreted to mean REWIND.

subdirectory specifies the subdirectory path that leads to the member to be archived. The subdirectory is a list of subdirectory names separated by forward slashes (/). A subdirectory name can be from 1 to 12 characters long. It is made up of a 1 to 8 character name and may be optionally followed by a period and a 1 to 3 character extension. The total length of the specified subdirectory cannot exceed 62 characters.

A subdirectory is valid only when a specific member is being archived and the third parameter is MEMBER. If not specified, the base folder (root directory) is searched for the member. When all MARKED members are to be archived, the base folder as well as all subdirectories are searched for marked members.

Example 1

This example shows how to archive members marked for archive in a folder named FLDR1 with a retention period of 60 days. The tape that will contain the archived member(s) is in tape drive 2 and the volume ID of the tape is VOL002. After the member(s) is archived, the tape will be rewound and unloaded.

```
ARCHIVE FLDR1,,,,60,T2,VOL002,,,,,UNLOAD
```

Example 2

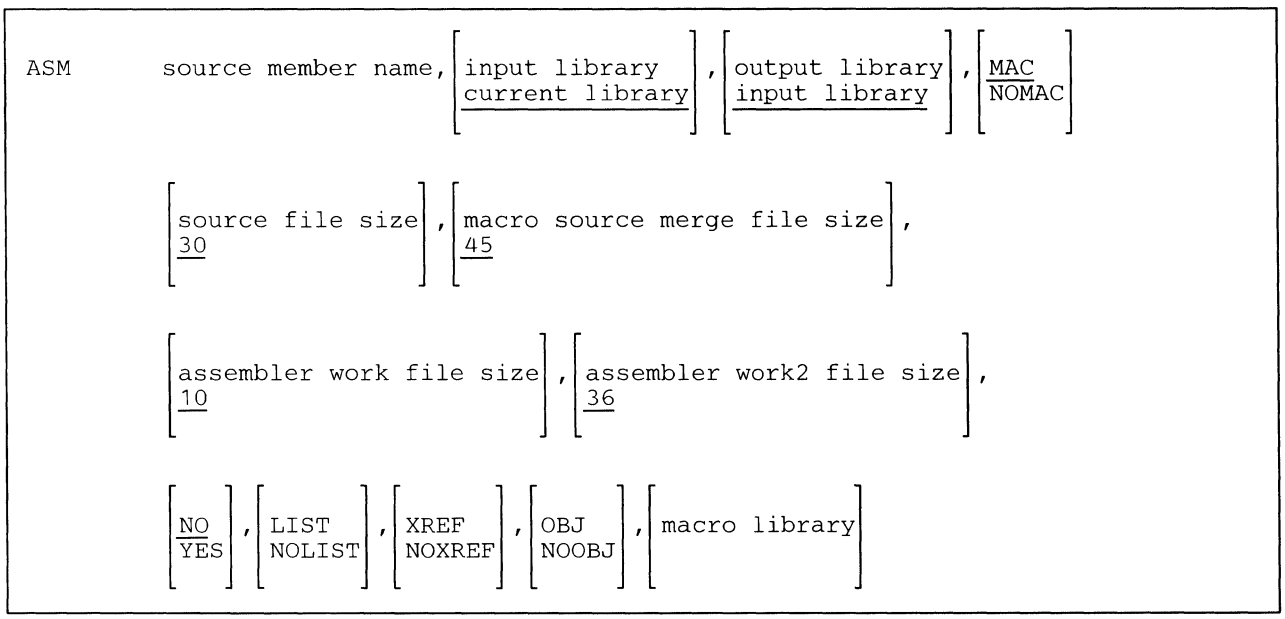
This example shows how to archive a member named MYMEMBER.DOC from folder MYFOLDER with a subdirectory of SUB1.DIR/SUB2.DIR to a diskette in slot S1 with a volume ID of VOL01. After the member has been archived, the text for the member is to be deleted. The label of the file created on diskette will be MYMEMBER.DOC which is the member name.

```
ARCHIVE MYFOLDER,DOCUMENT,MEMBER,DELTEXT,,I1,VOL01,
MYMEMBER.DOC,,,,,SUB1.DIR/SUB2.DIR
```

ASM Procedure

The ASM procedure compiles or assembles an Assembler source program or subroutine. The compiled program must then be link-edited to produce a program that can be run.

For more information on Assembler programming, see the manual *Programming with Assembler*. See the "OLINK Procedure" on page 4-325 for information on how to link-edit programs.



S9020052-0

source member name specifies the source member containing the assembler program or subroutine to be compiled.

input library specifies the name of the library that contains the source member. If this parameter is not specified, the current library is assumed.

output library specifies the name of the library that will contain the compiled program (also called an object module). The compiled program is placed in a library subroutine member and must be link-edited to be run. If the library is not specified, the input library is assumed.

MAC specifies that the macroinstruction processor is to be used. If the parameter is not specified, MAC is assumed.

NOMAC specifies that the macroinstruction processor is not to be used.

source file size specifies the number of blocks to use for the $SOURCE file used by the compiler, and can be any number from 1 through 999. If the parameter is not specified, 30 blocks are assumed.

macro source merge file size specifies the number of blocks to use for the $ASMINPT file used by the compiler, and can be any number from 1 through 999. If the parameter is not specified, 45 blocks are assumed.

assembler work file size specifies the number of blocks to use for the $WORK work file used by the compiler, and can be any decimal number from 1 through 999. If the parameter is not specified, 10 blocks are assumed.

assembler work2 file size specifies the number of blocks to use for the $WORK2 work file used by the compiler, and can be any decimal number from 1 through 999. If the parameter is not specified, 36 blocks are assumed.

NO specifies that the ASM procedure is not to be placed on the job queue, but it is to run from the display station. If no parameter is specified, NO is assumed.

YES specifies that the ASM procedure is to run from the job queue.

LIST or NOLIST specifies the listing option to be used. If no parameter is specified, the list option specified on the assembler OPTIONS statement is used.

LIST specifies that the ASM procedure is to produce a compiler listing.

NOLIST specifies that the ASM procedure is not to produce a compiler listing. Only errors and control statements are printed.

XREF or NOXREF specifies the cross-reference option to be used. If no parameter is specified, the list option specified on the assembler OPTIONS statement is used.

XREF specifies that the ASM procedure is to produce a cross-reference listing of the program.

NOXREF specifies that the ASM procedure is not to produce a cross-reference listing.

OBJ or NOOBJ specifies whether the ASM procedure should place the compiled program in the specified library. If no parameter is specified, the object option specified on the assembler OPTIONS statement is used.

OBJ specifies that the ASM procedure is to place the compiled program in the library as a subroutine member.

NOOBJ specifies that the ASM procedure is not to place the compiled program in the library.

macro library specifies the library that is searched for assembler macroinstructions. If this parameter is not specified, or if #ASMLIB is specified, the normal search order will be used (search #ASMLIB first, then #LIBRARY). If #LIBRARY is specified, #LIBRARY is searched first, then #ASMLIB. If a user library is specified, it is searched first, followed by #ASMLIB, and then #LIBRARY.

Example

This example shows how to compile an assembler program named PROGRAM1. The source member is contained in the current library and the subroutine member is to be placed in the current library. Also, a complete compiler listing and a cross-reference listing of the program is to be generated.

```
ASM PROGRAM1,,,,,,,,,LIST,XREF
```

ASMLOAD Procedure

The ASMLOAD procedure creates a library named #ASMLIB and copies the Assembler support from diskette into that library. ASMLOAD copies additional support into the system library (#LIBRARY). The ASMLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the ASMSAVE procedure. See the "ASMSAVE Procedure" on page 4-30 for information about how to save the Assembler support on diskette.

The ASMLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the ASMLOAD procedure to restore support that has been saved by ASMSAVE.

If Assembler is not currently on the system, you must use the TOLIBR procedure to copy the diskette file ASM into #LIBRARY before running ASMLOAD.

If #LIBRARY was backed up with Assembler on the system and then replaced before ASMLOAD is run, you do not have to copy the diskette file ASM using the TOLIBR procedure.

```
ASMLOAD  [A1 ]  [S1   ]  [library size]  [directory size]
         [A2 ]  [S2   ]  [180         ]  [55           ]
         [A3 ]  [S3   ]
         [A4 ]  [M1.nn]
                [M2.nn]
```

S9020053-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #ASMLIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #ASMLIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

library size specifies the size, in blocks, for #ASMLIB. One block contains 2560 bytes. The maximum library size is 6553 blocks. The minimum library size is 180 blocks. If no parameter is specified, 180 blocks are assumed.

directory size specifies the size, in sectors, of the directory for #ASMLIB. The maximum directory size is 1024 sectors. The minimum directory size is 55 sectors. One library block contains 10 sectors. If no parameter is specified, 55 sectors are assumed.

Example

Create #ASMLIB on disk and copy the Assembler support from diskette.

```
ASMLOAD
```

ASMSAVE Procedure

The ASMSAVE procedure copies the Assembler support to diskette. The Assembler support from the libraries #ASMLIB and #LIBRARY is copied. You should use the "ASMLOAD Procedure" on page 4-29 to load the Assembler support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPASM and be located in diskette slot S1.

```
ASMSAVE
```

S9020054-0

The ASMSAVE procedure has no parameters.

Example

Copy the Assembler support to diskette.

```
ASMSAVE
```

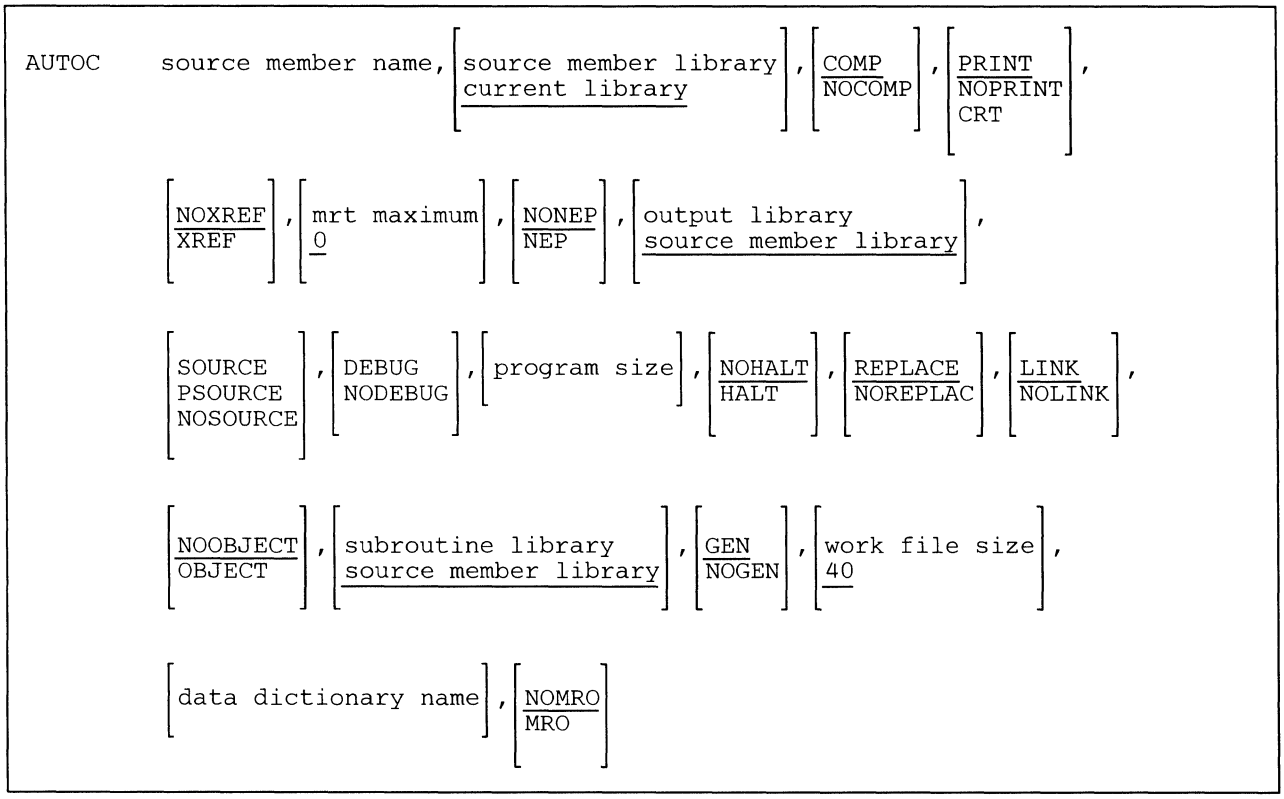
AUTO Procedure

This procedure is supported only for compatibility with the IBM System/34. The procedure you should use on the System/36 is the "AUTOC Procedure" on page 4-31.

AUTOC Procedure

The AUTOC procedure compiles an RPG II program that contains auto report specifications.

For information on auto report, see the manual *Programming with RPG II.*



S9020055-1

source member name specifies the library source member that contains the auto report program specifications.

source member library specifies the name of the library that contains the source member to be compiled. If this parameter is not specified, the current library is assumed.

COMP specifies that the RPG compiler should be run as part of the auto report function. If no parameter is specified, COMP is assumed.

NOCOMP specifies that the RPG compiler should not be run as part of the auto report function.

PRINT specifies that the compiler listing created by the AUTOC procedure should be printed. If no parameter is specified, PRINT is assumed.

NOPRINT specifies that no listing is to be printed or displayed.

CRT specifies that the compiler listing created by the AUTOC procedure is to be displayed at the display station that is running the AUTOC procedure.

NOXREF specifies that the AUTOC procedure should not produce a cross-reference listing of the RPG II program. If no parameter is entered, NOXREF is assumed.

XREF specifies that a cross-reference listing is to be produced.

mrt maximum specifies the maximum number of display stations (also called multiple requester terminals or MRTs) that can use the program at the same time. This number can be any number from 0 through 99. If the parameter is not entered, a value of 0 is assumed. A value of zero indicates that the program is a single requester terminal (SRT) program; that is, each display station that runs the program is really running its own copy of the program. A value of 1 or more indicates that the program is a multiple requester terminal (MRT) program.

For more information on MRT programs, see the *Concepts and Programmer's Guide.*

NONEP specifies that the program is not a never-ending program. If no parameter is entered, NONEP is assumed.

NEP specifies that the program is a never-ending program. See the NEP parameter of the "ATTR OCL Statement" on page 5-11 for more information on never-ending programs.

output library specifies the name of the library that is to contain the compiled program. If this parameter is not specified, the source member library is assumed.

SOURCE, PSOURCE, or NOSOURCE specifies the print option to be used instead of the entry in column 11 of the control specification in the source program. If no parameter is specified, the entry in column 11 of the control specification is used. See the manual *Programming with RPG II* for the information printed by each option.

SOURCE specifies that the AUTOC procedure is to create a full compiler listing.

PSOURCE specifies that the AUTOC procedure is to create a partial compiler listing.

NOSOURCE specifies that the AUTOC procedure is to create no compiler listing.

DEBUG or NODEBUG specifies the debug option to be used instead of the entry in column 15 of the control specification in the source program. If no parameter is specified, the value specified in column 15 of the control specification is used.

DEBUG specifies that the debug operation in the source program is to be used.

NODEBUG specifies that the debug operation in the source program is not to be used.

program size specifies the program size in K bytes (one K byte equals 1024 bytes) to be used instead of the program size entered in columns 12 through 14 of the control specification in the source program. The size can be an even number from 2 through 64. If no size is specified, the size specified in in columns 12 through 14 of the control specification is used.

NOHALT specifies that the compiler should not stop and display an error message if a warning or terminating error is found in the program. If no parameter is specified, NOHALT is assumed.

HALT specifies that an error message should be displayed if a warning or terminating error is found in the program.

REPLACE specifies that if a load member or subroutine member is being created and a load member or subroutine member with the same name as your program already exists in the output library, the newly compiled program is to replace the existing load member or subroutine member (no message is displayed indicating the replace). If no parameter is specified, REPLACE is assumed.

NOREPLAC specifies that if a load member or subroutine member is being created and a load member or subroutine member with the same name as your program already exists in the output library, a message is to be displayed; then you can either replace the member or cancel the procedure.

LINK specifies that a load member is to be created. This parameter creates a program that can be run, without having to first link-edit the program by using the OLINK procedure. If no parameter is entered, LINK is assumed.

NOLINK specifies that no load member is to be created.

NOOBJECT specifies that a subroutine member is not to be created. If no parameter is specified, NOOBJECT is assumed.

OBJECT specifies that a subroutine member is to be created. This member must be link-edited using the OLINK procedure or the overlay linkage editor before it can be run.

subroutine library specifies the name of the library that contains one or more Assembler subroutines to be combined with the program being compiled. If no parameter is specified, the source member library is assumed.

GEN specifies that if the source program contains a CONSOLE file, the display formats for that CONSOLE file are to be created. If no parameter is specified, GEN is assumed.

NOGEN specifies that no display formats for a CONSOLE file are to be created.

work file size specifies the size of the AUTOC work files in blocks. If no size is specified, 40 blocks are assumed.

data dictionary name specifies the data dictionary that contains the communications file definition to be used with the program being compiled.

MRO or NOMRO specifies whether the compiler is to use memory resident overlays. If no parameter is specified, NOMRO is assumed.

MRO specifies that the compiler is to use memory resident overlays.

NOMRO specifies that the compiler is not to use memory resident overlays.

Example

This example shows how to compile an RPG II program named PAYROLL. The source program is in the current library; the compiled and link-edited load member is to be placed in the current library. A source listing and a cross-reference listing are to be generated, and the RPG debug operation is to be used.

```
AUTOC PAYROLL,,,,XREF,,,,SOURCE,DEBUG
```

BACKUP Procedure

The BACKUP procedure is supported only for compatibility with the IBM System/34. The procedure you should use on the System/36 is the "SAVELIBR Procedure" on page 4-431.

BALPRINT Procedure

The BALPRINT procedure balances spooled output among a group of printers by redirecting spool file entries from the busy printers to the idle printers in a printer group. The BALPRINT procedure allows you to:

- Start and stop printer load balancing as well as display information about active printer groups
- Specify a printer group identification number for the printers for which spooled output is to be balanced
- Specify a time interval at which balancing is to occur
- Specify the printer IDs of the printers that are to be in the printer group

If password security is active, the following apply to running the BALPRINT procedure:

- If you have master security, security officer, or system operator authority, you can run this procedure from any display station, you can evoke this procedure, and you can submit a job to the job queue to run the procedure. Any of the printers attached to the system may be specified as members of a group.
- If you have subconsole operator authority, you can run this procedure from a subconsole. Only the printers that are controlled by the subconsole may be specified as members of a group. If you use the STOP option and specify a group of ALL, the procedure will stop all groups that consist solely of printers controlled by the subconsole.
- If you are a display station operator, you cannot start or stop the procedure, but you can display information about active printer groups.

If password security is not active, the BALPRINT procedure can be started and stopped at the system console, the system service device, and subconsoles, and information about active printer groups can be displayed from any display station.

- If you run the procedure from the system console or the system service device, any printer attached to the system may be specified as a member of a group.
- If you run the procedure from a subconsole, only those printers controlled by the subconsole may be specified as members of a group.
- If you run the procedure from a subconsole and specify the STOP option with a group of ALL, the procedure will stop all the groups that consist solely of printers controlled by the subconsole.

| To start printer load balancing:

```
BALPRINT  [START],[group],[time interval],[list of printer ids]
          [     ] [1    ] [60           ] [                  ]
```

S9020612-0

| To stop printer load balancing:

```
BALPRINT  STOP[,group]
              [ALL  ]
              [1    ]
```

S9020613-0

| To display information about active printer groups:

```
BALPRINT  DISPLAY[,group]
                 [ALL  ]
```

S9020614-0

| **START** specifies that printer load balancing is to be started for the specified printer group.

| *Note: The START option cannot be canceled or inquired. See the "ATTR OCL Statement" on page 5-11 for additional information regarding its CANCEL and INQUIRY parameters.*

| If the first parameter is not specified, START is assumed.

| **group** specifies the printer group identification number for the specified list of printer IDs for which spooled printer output is to be balanced. You can use any whole number from 1 to 255; however, you cannot use the same number for more than one group.

| If a group identification number is not specified, a group number of 1 is assumed. You can use the DISPLAY option of the BALPRINT procedure to determine the printer group identification number(s) of all active printer group(s).

time interval specifies the amount of time, in seconds, that will elapse between attempts to balance spooled printer output. You can use any whole number from 10 to 900. When choosing a time interval, you should consider:

- The average amount of data that is printed from a spool file entry
- The speed of the printers in the group
- How much idle printer time (that is, for printers in the group) is acceptable to you
- How much other work the system is performing while printer load balancing is in effect

If no parameter is specified, 60 seconds is assumed. It is suggested that you start with 60 seconds and then try different values to determine if one of those values is better for your particular environment.

list of printer ids specifies the 2-character printer IDs for the printers that are to be members of the group. You must specify at least two printer IDs, and you may specify a maximum of ten printer IDs. To create a group consisting of printers P1, P2, P3, and P4, you would specify P1,P2,P3,P4. If you specify more than ten printers, the printer IDs in excess of the tenth will be ignored. If you include a null entry in the list, for example, P1,P2,,P3, the null entry ends the list and only P1 and P2 are considered members of the group. You may specify each printer as a member of only one group.

When choosing the members of a printer group, you should consider:

- The proximity and speeds of the printers in the group.
- The capabilities of the printers in the group. If spooled output is redirected to a printer with different characteristics, printing or programming errors may occur, and your output may not print properly. In particular, DW/36 jobs may not complete successfully.
- Whether or not a printer has been specified for MSRJE Host Printer Balancing. If this is so, you should not include that printer unless you include all of the printers from the same MSRJE forms control table entry. You can determine which printers have been specified for MSRJE Host Printer Balancing by using the RJTABLE procedure and either printing or reviewing the table entries.

STOP specifies that printer load balancing is to be stopped for the specified printer group(s).

Note: The STOP option cannot be canceled or inquired. See the "ATTR OCL Statement" on page 5-11 for additional information regarding its CANCEL and INQUIRY parameters.

The CANCEL control command can be used to stop printer load balancing but only after the message "BALPRINT procedure started for group nnn" has been issued. It is suggested that you use the STOP option of the BALPRINT procedure to stop printer load balancing. This will avoid the delay that may occur if the CANCEL control command is used.

group specifies the printer group identification number for which printer load balancing is to be stopped. Any whole number from 1 to 255 can be specified. You can use the DISPLAY option of the BALPRINT procedure to determine the printer group indentification number(s) of all active printer group(s).

If no parameter is specified, a group number of 1 is assumed.

ALL specifies that printer load balancing for all active printer groups is to be stopped.

DISPLAY specifies that information is to be displayed for the specified printer group(s).

Note: The DISPLAY option of the BALPRINT procedure is not available when the procedure is run from the job queue or evoked.

group specifies the printer group identification number for which information is to be displayed. Any whole number from 1 to 255 can be specified.

ALL specifies that information about all active printer groups is to be displayed.

Example 1

This example shows how you could start printer load balancing so that it balances spooled output between printers P1 and P2 every 60 seconds.

```
BALPRINT START,1,60,P1,P2
```

or:

```
BALPRINT ,,,P1,P2
```

Example 2

This example shows how you could stop printer load balancing for printer group 1.

```
BALPRINT STOP,1
```

or:

```
BALPRINT STOP
```

Example 3

This example shows how you could display information about all active printer groups.

```
BALPRINT DISPLAY
```

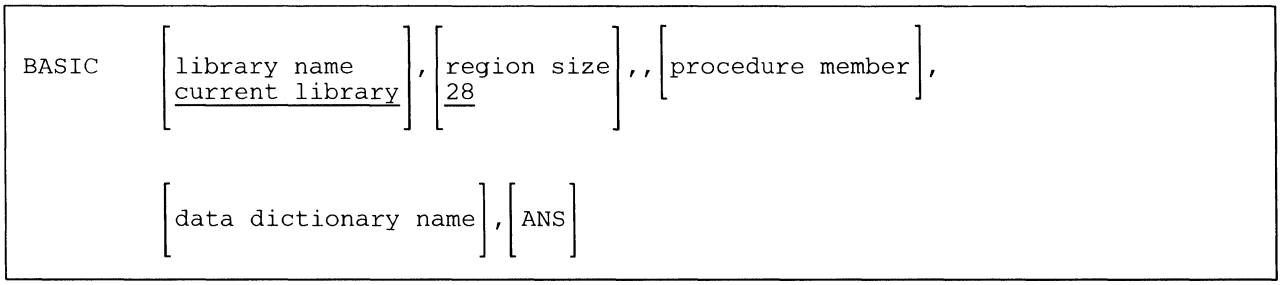
This following display is shown (this display is shown with some sample information):

```
                        STATUS OF PRINTER LOAD BALANCING              ALL
------------------------------------------------------------------------------

Press Enter for current status.

PRINTER        TIME INTERVAL        PRINTER
GROUP            (SECONDS)          IDS

  001               060             P1, P2, P3, P4, P5, P6, P7, P8, P9, S1

  002               090             S5, S6









Cmd3=Go back            Roll=Page
(...message text, if any, appears in these positions...)    COPR IBM Corp. 1987
```

Figure 4-2. Sample DISPLAY Option Output

BASIC Procedure

The BASIC procedure starts a BASIC session. You can enter, change, save, or remove BASIC programs, BASIC procedures, and other data; run programs immediately; and run BASIC procedures. See the manual *Programming with BASIC* for more information about BASIC.



S9020056-0

library name specifies the library for the BASIC session. If a library name is not specified, the current library is assumed.

region size specifies the number of K-bytes (1024 bytes) of storage to reserve for the BASIC session. The entry should be an even number from 28 through 64. If you do not enter a number, 28K bytes are assumed.

The third parameter position is reserved for compatibility with IBM System/34 BASIC

procedure member specifies the name of a library procedure member that contains OCL statements to be used when programs are run.

The procedure member must be in the current library or in the system library (#LIBRARY). The procedure member can contain any OCL statements that can be placed between the LOAD and RUN OCL statements. See Chapter 5, "OCL Statements" for descriptions of the OCL statements.

data dictionary name specifies the data dictionary to be used for this BASIC session.

ANS specifies that your BASIC programs are run using the American National Standard for Minimal BASIC (ANS X3.60-1978). If you do not specify ANS, your BASIC programs are run using the System/36 rules.

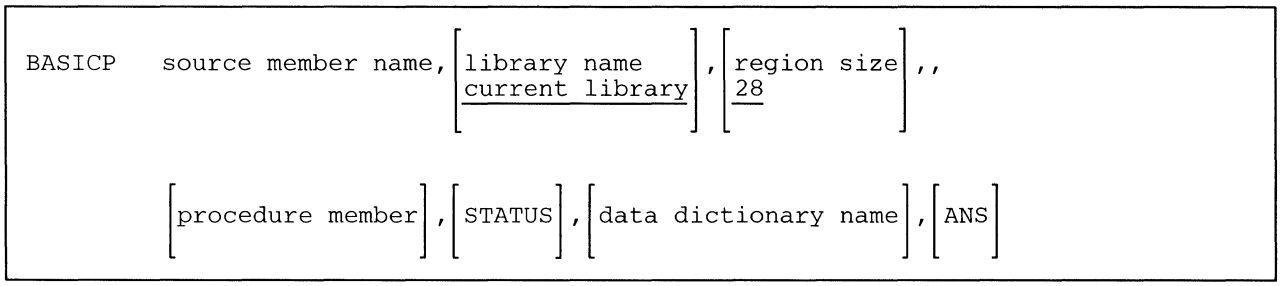
Example

To start a BASIC session:

```
BASIC
```

BASICP Procedure

The BASICP procedure runs BASIC procedures that contain no errors and have been saved as library source members. For more information about BASIC, see the manual *Programming with BASIC*.



S9020057-0

source member name specifies the library source member that contains the BASIC procedure to be run.

library name specifies the library that contains the source member and is to be the current library during the BASIC session. If a library name is not specified, the current library is assumed.

region size specifies the number of K-bytes (1024 bytes) of storage to reserve for the BASIC session. This entry should be an even number from 28 through 64. If you do not enter a number, 28K bytes are assumed.

The fourth parameter position is reserved for compatibility with IBM System/34 BASIC.

procedure member specifies the name of a library procedure member that contains OCL statements to be used when programs are run. The procedure member must be in the current library or in the system library (#LIBRARY). The procedure member can contain any OCL statements that can be put between the LOAD and RUN OCL statements. See Chapter 5, "OCL Statements" for descriptions of the OCL statements.

STATUS prints information about the BASIC session.

data dictionary name specifies the data dictionary to be used for this BASIC session.

ANS specifies that your BASIC programs are run using the American National Standard for Minimal BASIC (ANS X3.60-1978). If you do not specify ANS, your BASIC programs are run using the System/36 rules.

Example

This example shows how to run a BASIC procedure named BASPROC. The procedure is contained in the current library.

```
BASICP BASPROC
```

BASICR Procedure

The BASICR procedure runs BASIC programs that contain no errors and have been saved as library subroutine members. For more information about BASIC, see the manual *Programming with BASIC*.

```
BASICR   subroutine member name,[library name    ],[region size],,
                                 [current library]  [28         ]

         [procedure member],[STATUS],[data dictionary name],[ANS]
```

S9020058-0

subroutine member name specifies the library subroutine member that contains the BASIC program to be run.

library name specifies the library that contains the subroutine member and is to be the current library for the BASIC session. If a library name is not specified, the current library is assumed.

region size specifies the number of K-bytes (1024 bytes) of storage to reserve for the BASIC session. This entry should be an even number from 28 through 64. If you do not enter a region size, 28K bytes are assumed.

The fourth parameter position is reserved for compatibility with IBM System/34 BASIC.

procedure member specifies the name of a library procedure member that contains OCL statements to be used when the program is run. The procedure member must be in the current library or in #LIBRARY. The procedure member can contain any OCL statements that can be put between the LOAD and RUN OCL statements. See Chapter 5, "OCL Statements" for descriptions of the OCL statements.

STATUS prints information about the BASIC session.

data dictionary name specifies the data dictionary to be used for this BASIC session.

ANS specifies that your BASIC programs are run using the American National Standard for Minimal BASIC (ANS X3.60-1978). If you do not specify ANS, your BASIC programs are run using the System/36 rules.

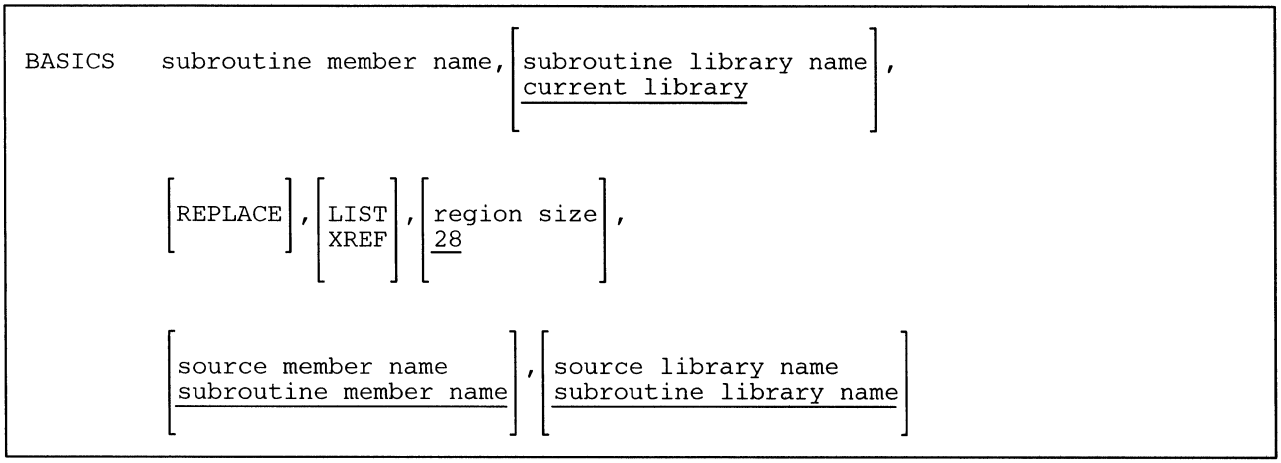
Example

This example shows how to run the BASIC program named BASPROG. The program is contained in the current library.

```
BASICR BASPROG
```

BASICS Procedure

The BASICS procedure converts a library source member that contains a BASIC program into a library subroutine member. The procedure can print a listing of the program, and indicates any syntax errors in the program; this can help you to convert a program from another system. For more information about BASIC, see the manual *Programming with BASIC*.



S9020059-0

subroutine member name specifies the library subroutine member to be created.

subroutine library name specifies the library that is to contain the subroutine member. If you do not enter a library name, the current library is assumed.

REPLACE specifies that if a subroutine member exists in the specified library, the new subroutine member is to replace the old subroutine member. If REPLACE is not entered and the subroutine member already exists, a message is displayed and the operator can either replace the existing member or cancel the BASICS procedure.

LIST specifies that the contents of the library source member are to be printed on the printer assigned to the display station, along with any format error messages.

XREF specifies that the contents of the library source member and a cross-reference listing of the program are to be printed on the printer assigned to the display station, along with any format error messages.

Note: If LIST or XREF is not entered and the library source member contains errors, a message is displayed and the operator must cancel the BASICS procedure.

region size specifies the number of K-bytes (1024 bytes) of storage to reserve for the BASIC session. This entry should be an even number from 28 through 64. If a region size is not specified, 28K bytes are assumed.

source member name specifies the library source member containing the program to be converted by BASICS. If you do not enter the source member name, the subroutine member name is assumed.

source library name specifies the library containing the source member to be converted. If you do not enter the source library name, the subroutine library name (if given) or the current library is assumed.

Example

This example shows how to convert the BASIC program in the source member named BASPROG into a subroutine member. The source member is in the current library and is to be listed. The generated subroutine member is to be placed in the current library.

```
BASICS BASPROG,,,LIST
```

BASLOAD Procedure

The BASLOAD procedure creates a library named #BLLIB and copies the BASIC support from diskette into that library. BASLOAD copies additional support into the system library (#LIBRARY). BASLOAD can also create a library named #BLHPLIB to contain the BASIC help support. The BASLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the BASSAVE procedure. See the "BASSAVE Procedure" on page 4-45 for information about how to save the BASIC support on diskette.

The BASLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the BASLOAD procedure to restore support that has been saved by BASSAVE.

If BASIC is not currently on the system, you must use the TOLIBR procedure to copy the diskette file BASIC into #LIBRARY before running BASLOAD.

If #LIBRARY was backed up with BASIC on the system and then replaced before BASLOAD is run, you do not have to copy the diskette file BASIC using the TOLIBR procedure.

```
BASLOAD  [ A1 ]  , [ S1      ]  , [ Y ]
         [ A2 ]    [ S2      ]    [ N ]
         [ A3 ]    [ S3      ]    [   ]
         [ A4 ]    [ M1.nn   ]
         [    ]    [ M2.nn   ]
                   [         ]
```

S9020060-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #BLLIB and #BLHPLIB are placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #BLLIB and #BLHPLIB are placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Y or N specifies whether the BASIC help support is to be loaded from diskette. If no parameter is specified, you will be prompted for the value.

Y specifies the support is to be loaded. A library named #BLHPLIB is created and contains the BASIC help support.

N specifies the support is not to be loaded.

Example

Create #BLLIB and #BLHPLIB on disk and copy the BASIC support from diskette.

```
BASLOAD ,,Y
```

BASSAVE Procedure

The BASSAVE procedure copies the BASIC support to diskette. The BASIC support from the libraries #BLLIB, #BLHPLIB (if it exists), and #LIBRARY is copied. You should use the "BASLOAD Procedure" on page 4-44 to load the BASIC support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPBASC and be located in diskette slot S1.

```
BASSAVE
```

S9020061-0

The BASSAVE procedure has no parameters.

Example

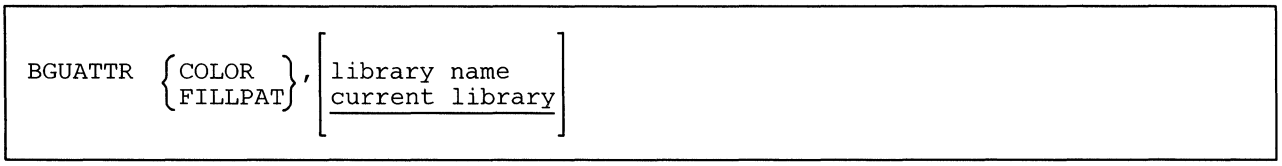
Copy the BASIC support to diskette.

```
BASSAVE
```

BGUATTR Procedure

The BGUATTR procedure changes the printer fill patterns or color palettes to be used by the Business Graphics Utilities/36 (BGU/36).

See the *Business Graphics Utilities/36* manual for more information on BGUATTR.



S9020497-1

COLOR specifies the color palettes to be changed in a library load member with the name of #BGUTABL in the library specified by the library name parameter. Color palettes 1 through 9 are provided with BGU/36. Color palette 0, the default palette, cannot be changed.

FILLPAT specifies the fill patterns to be created or changed in a library load member named #BGFPRTN in the library specified by the library name parameter.

library name specifies the name of the library containing the display color palette or the printer fill pattern to be changed. If no library is specified, the current library is assumed.

Example

This example shows how to change the fill patterns in the library OWNLIB.

```
BGUATTR FILLPAT,OWNLIB
```

BGUCHART Procedure

The BGUCHART procedure creates or updates a chart, prints a chart on a graphics-capable printer, displays a chart on a graphics-capable display station, or plots a chart on a plotter.

See the *Business Graphics Utilities/36* manual for more information on BGUCHART.

To design a chart:

```
BGUCHART  [DESIGN],[chart name],[library name      ],[1     ]
                                 [current library  ]  [page #]
```

S9020525-2

To plot or view a chart:

```
BGUCHART  {PLOT},[chart name],[library name      ],[ALL   ]
          {VIEW}              [current library  ]  [page #]
```

S9020500-3

To print a chart on a graphics-capable printer:

```
BGUCHART  PRINT,[chart name],[library name      ],[ALL   ],[prID],[1     ],
                             [current library  ]  [page #]       [copies]

          [forms #],[HOLD  ]
                    [NOHOLD]
```

S9020499-2

DESIGN specifies that a chart should be created or updated. If the first parameter is not specified, DESIGN is assumed.

PLOT specifies that a chart should be sent to the plotter attached to the requesting display station. If the requesting display station is not a BGU/36 supported graphics display station with an attached and powered on plotter, an error message is issued.

PRINT specifies that a chart should be printed on a BGU/36 supported graphics matrix line printer.

VIEW specifies that a chart should be displayed at a BGU/36 supported graphics display station. If the requesting display station is not graphics-capable, an error message is issued and the program is terminated.

chart name specifies the name of a chart member. If a chart name is not specified, the member selection list is displayed.

library name specifies the name of the library that contains the chart member specified in parameter 2. If a library name is not specified, the current library is used.

ALL specifies that all chart pages defined in a chart member are to be plotted, printed, or viewed. If this parameter is not specified and PLOT, PRINT, or VIEW is specified as the first parameter, ALL is assumed.

page # specifies the page number of the chart to be plotted, printed, or viewed. If DESIGN is specified as the first parameter, page # specifies the page on which to start the edit session. If this parameter is not specified and DESIGN is specified as the first parameter, 1 is assumed. If this parameter is not specified and PLOT, PRINT, or VIEW is specified as the first parameter, ALL is assumed.

prID specifies the ID of the output printer. If a printer ID is not specified, the session printer is used. This parameter is only valid if PRINT is specified.

copies specifies the number of copies of a chart that is to be printed. If a number of copies is not specified, one copy is assumed. This parameter is only valid if PRINT is specified.

forms # specifies the forms number to be used to print a chart. If a forms number is not specified, the default session forms number for the requesting display station is used. This parameter is only valid if PRINT is specified.

HOLD specifies that a chart print file should be held in the spool file after all copies have been printed. If this parameter is not specified, NOHOLD is assumed. This parameter is only valid if PRINT is specified.

NOHOLD specifies that a chart print file should not be held in the spool file after all copies have been printed. This parameter is only valid if PRINT is specified.

Example 1

This example shows how to create or update a chart named PROJECT in the library OWNLIB.

```
BGUCHART ,PROJECT,OWNLIB
```

Example 2

This example shows how to output the graphs defined for page 5 in a chart member named PROJECT in the library OWNLIB on a plotter.

```
BGUCHART PLOT,PROJECT,OWNLIB,5
```

Example 3

This example shows how to display the graphs defined on page 1 in a chart member named PROJECT in the library OWNLIB on a graphics-capable display.

```
BGUCHART VIEW,PROJECT,OWNLIB,1
```

Example 4

This example shows how to print all the graphs defined in a chart named PROJECT in the library OWNLIB on a graphics-capable printer. Two copies of the chart are to be printed and the chart file is to be held in the spool file after the two copies are printed.

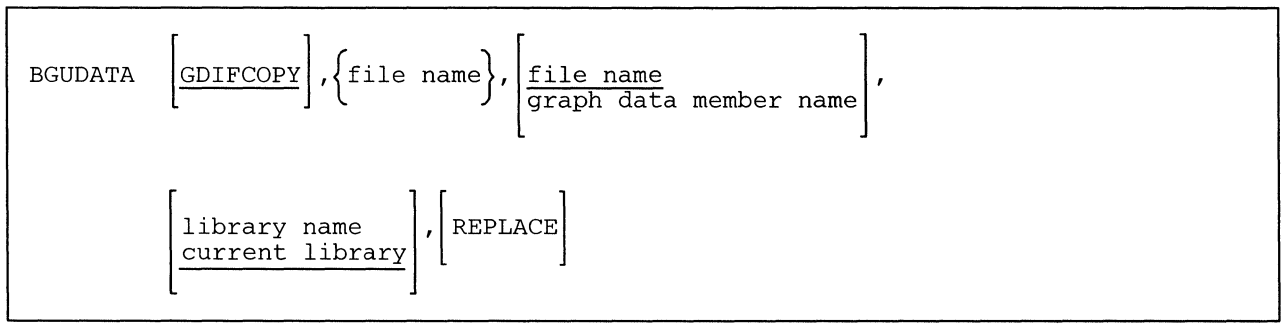
```
BGUCHART PRINT,PROJECT,OWNLIB,,P2,2,,HOLD
```

BGUDATA Procedure

The BGUDATA procedure copies a user-generated graph data input file (GDIF) to a data member and stores the data member in the library specified by the library name parameter.

See the *Business Graphics Utilities/36* manual for more information on BGUDATA.

The BGUDATA procedure can be run by selecting the appropriate option on the BGU/36 menu, or by running the following procedure:



S9020498-2

GDIFCOPY specifies that a user-generated graph data input file (GDIF) is to be copied to a data member that is stored in the user's current library. If the first parameter is not specified, GDIFCOPY is assumed.

file name specifies the name of a graph data input file that is to be copied to a graph data member. If a file name is not specified, or if the specified file is not found, an error message is issued and the procedure is terminated.

graph data member name specifies the name of a graph data member that is created by BGUDATA. If a graph data member name is not specified, the file name specified in the second parameter is used as the graph data member name.

library name specifies the name of the library that contains the data member created by BGUDATA. If a library name is not specified, the current library is used.

REPLACE specifies that if the library member already exists with the specified library member name, it is to be replaced. If REPLACE is specified, the new member replaces the existing member with the duplicate name and no message regarding the replacement is displayed.

If REPLACE is not specified, the member is placed in the library unless a duplicate is found, at which time the system displays a message telling the operator that a duplicate exists. In response to the message, the operator can either cancel the job or continue processing. If the job is continued, the new member replaces the existing member in the library.

Example

This example shows how to copy the file PAYROLL into the member called PAYROLL (by default) in the library PAYLIB. If the member PAYROLL exists, it will be replaced.

```
BGUDATA GDIFCOPY,PAYROLL,,PAYLIB,REPLACE
```

BGUGRAPH Procedure

The BGUGRAPH procedure creates or updates a graph, prints a graph on a graphics-capable printer, displays a graph on a graphics-capable display station, plots a graph on a plotter, or creates a graph object file version of the graph.

See the *Business Graphics Utilities/36* manual for more information on BGUGRAPH.

To create a graph object file version of a graph:

```
BGUGRAPH   BLDFILE,[graph format member name],[graph data member name],

           [library name      ],[graph object file name      ]
           [current library   ],[graph format member name    ]
```

S9020501-2

To design a graph, plot a graph on a plotter, or display a graph on a graphics-capable display station:

```
BGUGRAPH   [DESIGN ],[graph format member name],[graph data member name],
           [PLOT   ]
           [VIEW   ]

           [library name      ]
           [current library   ]
```

S9020502-1

To print a graph on a graphics-capable printer:

```
BGUGRAPH   PRINT,[graph format member name],[graph data member name],

           [library name      ],[prID],[1     ],[forms #],[HOLD  ]
           [current library   ]        [copies]           [NOHOLD]
```

S9020503-2

BLDFILE specifies that a graph object file should be created.

DESIGN specifies that a graph should be created or updated. If the first parameter is not specified, DESIGN is assumed.

PLOT specifies that the graph should be sent to the plotter attached to the requesting display station. If the requesting display station is not a BGU/36 supported graphics display station with an attached and powered-on plotter, an error message is issued.

PRINT specifies that the graph should be printed on a BGU/36 supported graphics matrix line printer.

VIEW specifies that the graph should be displayed at the requesting display station. If the requesting display station is not graphics-capable, an error message is issued and the program is terminated.

graph format member name specifies the name of the graph format member to be used to generate the graph. If a graph format member name is not specified, the member selection list is displayed.

graph data member name specifies the name of the graph data member to be used to generate the graph. If a graph data member name is not specified, the graph data member pointed to by the graph format member is used. If the graph data member is not found in the specified library, an error message is issued.

library name specifies the name of the library that contains the graph members specified by the graph format member name and graph data member name. If a library name is not specified, the current library is used.

graph object file name specifies the name to be assigned to the graph object file created. If a file name is not specified, the file will be assigned the same name as the graph format member name. This parameter is only valid if BLDFILE is specified as the first parameter.

prID specifies the ID of the output printer. If a printer ID is not specified, the session printer is used. This parameter is only valid if PRINT is specified.

copies specifies the number of copies of the graph that is to be printed. If a number of copies is not specified, one copy is assumed. This parameter is only valid if PRINT is specified.

forms # specifies the forms number to be used to print the graph. If a forms number is not specified, the default session forms number for the requesting display station is used. This parameter is only valid if PRINT is specified.

HOLD specifies that the chart print file should be held in the spool file after all copies have been printed. If this parameter is not specified, NOHOLD is assumed. This parameter is only valid if PRINT is specified.

NOHOLD specifies that the chart print file should not be held in the spool file after all copies have been printed. This parameter is only valid if PRINT is specified.

Example 1

This example shows how to create a graph object file version of the graph STAT in the library MYLIB. The graph object file will be named STATFILE and STAT is the graph format member name.

```
BGUGRAPH BLDFILE,STAT,,MYLIB,STATFILE
```

Example 2

This example shows how to design a new graph in the library MYLIB. The graph format member name will be STAT and the graph data member name will be STATNUM.

```
BGUGRAPH DESIGN,STAT,STATNUM,MYLIB
```

Example 3

This example shows how to plot a graph named STAT. The graph format member name is STAT, the graph data member name will default. The graph will be in the library named MYLIB.

```
BGUGRAPH PLOT,STAT,,MYLIB
```

Example 4

This example shows how to print a graph named STAT in the library named MYLIB. The graph format member is named STAT and the graph data member will default. The ID of the graphics-capable printer is P2. Three copies of the graph will be printed on form number 7654 and the graph print file will be held in the spool file after the three copies are printed.

```
BGUGRAPH PRINT,STAT,,MYLIB,P2,3,7654,HOLD
```

Example 5

This example shows how to view a graph named STAT on a graphics-capable display station. The graph format member name is STAT, the graph data member name will default, the library name is MYLIB.

```
BGUGRAPH VIEW,STAT,,MYLIB
```

BGULOAD Procedure

The BGULOAD procedure creates a library named #BGULIB and copies the Business Graphics Utilities/36 (BGU/36) support from diskette into that library. BGULOAD copies additional support into the system library (#LIBRARY). BGU/36 help support is optionally loaded into #BGUHLIB. The BGULOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the BGUSAVE procedure.

The BGULOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the BGULOAD procedure to restore support that has been saved by BGUSAVE.

If BGU/36 is not currently on the system, you must use the TOLIBR procedure to copy the diskette file BGU into #LIBRARY before running BGULOAD.

If #LIBRARY was backed up with BGU/36 on the system and then replaced before BGULOAD is run, you do not have to copy the diskette file BGU using the TOLIBR procedure.

```
BGULOAD [A1|A2|A3|A4],[S1|S2|S3|M1.nn|M2.nn],[Y|N]
```

S9020534-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #BGULIB is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, #BGULIB is placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Y or N specifies whether or not to load the BGU/36 help support into #BGUHLIB. If this parameter is not specified, the user will be prompted for either a Y or N. If anything other than N is entered for the prompt, Y is assumed. N specifies that the BGU/36 help will not be loaded. Y specifies that the BGU/36 help support will be added.

Example

Create #BGULIB on disk and copy the BGU support from diskette.

```
BGULOAD
```

BGUSAVE Procedure

The BGUSAVE procedure copies the Business Graphics Utilities/36 (BGU/36) support to diskette. The BGU/36 support from #LIBRARY, #BGULIB, and #BGUHLIB is copied. You should use the BGULOAD procedure to load the BGU support from the backup diskettes. The diskettes to contain the saved copy must have a volume ID of PPBGU.

```
BGUSAVE
```

S9020532-0

The BGUSAVE procedure has no parameters.

Example

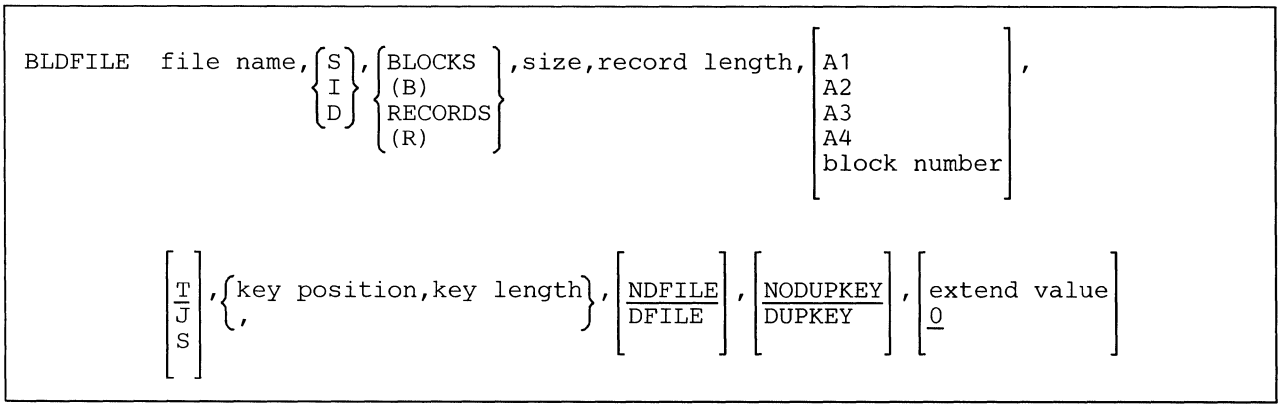
Copy BGU/36 support from #LIBRARY and #BGULIB to diskette.

```
BGUSAVE
```

BLDFILE Procedure

The BLDFILE procedure creates a disk file that contains no data. The file can then be referenced as an existing file by following jobs and job steps, which can place data into the file. The BLDFILE procedure can be used to create a file on a remote system.

The BLDFILE procedure runs the $FBLD utility program.



S9020062-0

file name specifies the file to be created. A file name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, @, or $). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. Do not use ALL as a file name.

S specifies that a sequential disk file is to be created.

I specifies that an indexed disk file is to be created.

D specifies that a direct disk file is to be created.

BLOCKS or B specifies that space for the file is to be allocated (or reserved) by blocks.

RECORDS or R specifies that space for the file is to be allocated (or reserved) by records.

size specifies the size of the file to be created. If BLOCKS or B is specified, the size of the file can be from 1 through the maximum number of blocks of disk storage configured on the system. If RECORDS or R is specified, the size of the file can be 1 through 8000000 records. The number cannot be greater than the number of unused blocks or records available for user files. You can use the CATALOG procedure to determine the amount of space available for files.

record length specifies the length of the record in bytes, and can be any decimal number from 1 through 4096. One byte contains one character of data.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked and the file is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, the file is placed on the least used disk unit.

block number specifies the location of the first block in the file. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk. Use the block number parameter with caution because the COMPRESS and RESTORE procedures can cause a file to be moved from the block number specified when the file was created. If space is not available, an error message will be issued. See the "FILE OCL Statement (for Disk Files)" on page 5-32 for more information about locating a file by a block number. If space is not available, an error message will be issued.

T, J, or S specifies the disk file retention type. If the file retention is not specified, T is assumed.

T specifies that the file is a **resident** file.

J specifies that the file is a **job** file.

S specifies that the file is a **scratch** file.

Note: For information about file retention types, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

key position specifies the starting position of the key area within the record, and must be a number from 1 through 4,096. The key position is required if an indexed file is being created (I is specified for parameter 2). If a key position is specified but an indexed file is not being created, an error message is displayed. The entire key, defined by the key position and key length, must be within the record.

key length specifies the length of the key area within the record, and must be a number from 1 through 120. The key length is required if an indexed file is being created (I is specified for parameter 2). If a key length is specified but an indexed file is not being created, an error message is displayed. The entire key, defined by the key position and key length, must be within the record.

DFILE specifies that the file is to be delete-capable.

Note: Deleting a record from a delete-capable file does not cause the record to be removed. Instead, the record is marked as deleted (hex FF is placed in the first position of the record).

NDFILE specifies the file is not to be a delete-capable file. If this parameter is not specified, NDFILE is assumed.

DUPKEY specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored.

NODUPKEY specifies that duplicate keys are not to be allowed in the indexed file being created. If this parameter is not specified, NODUPKEY is assumed. If the file being created is not an indexed file, this parameter is ignored.

extend value specifies the value by which a file can be extended each time additional space is needed in the file. The value specified is the number of records (if the file is allocated in records) or blocks (if the file is allocated in blocks) by which the file is to be extended.

The number cannot exceed 8000000 records or 312815 blocks. If no value or 0 is specified, the file being created will not be able to be extended; that is, it will not be automatically extended each time additional space is needed.

Example 1

Create a resident, delete-capable file that is 13 blocks long. The file is INVOICE, the record length is 50 bytes, the preferred file placement is on the first disk, and each record contains a 4-byte key beginning at position 9 in the record.

```
BLDFILE INVOICE,I,BLOCKS,13,50,A1,T,9,4,DFILE
```

Example 2

To create a job file named JOB for the PAYROLL procedure, you could do the following. The file is to be a sequential file, 10 blocks in size, and is to have a record length of 100.

```
* Create a job file
BLDFILE JOB,S,BLOCKS,10,100,,J
* Run the payroll procedure
PAYROLL
```

BLDINDEX Procedure

The BLDINDEX procedure creates a resident file that contains the alternative index for a physical file. The physical file must be a resident direct, sequential, or indexed file. If the physical file is a direct file, it must be delete-capable. The BLDINDEX procedure can be used to create an alternative index file on a remote system by specifying the local file label as it is defined in the system network resource directory (NRD). The physical file that the alternative index is based on must also be defined in the NRD, and the entries for the two files must indicate that the files are at the same remote location. See the *Concepts and Programmer's Guide* for more information on alternative indexes.

An alternative index file can be thought of as a file that is identical to the original physical file, but the key is defined differently. Note that the physical file is not copied; the keys in the alternative index file provide a different way to access the records in the physical file.

The BLDINDEX procedure runs the $FBLD utility program.

```
BLDINDEX alternative index file name,key position1,key length1,

        physical file name,[mmddyy ],[DUPKEY   ],[A1           ],
                           [ddmmyy ] [NODUPKEY ] [A2           ]
                           [yymmdd ] [         ] [A3           ]
                                     [         ] [A4           ]
                                     [         ] [block number ]

        [key position2,key length2],[key position3,key length3]
```

S9020063-0

alternative indexed file name specifies the new file that will contain the alternative index for the physical file.

key position1 specifies, for the alternative index, the starting position within the record for the entire key area or the starting position of the first field in the key. The position must be a number from 1 through 4096. The entire key, defined by the key position and key length, must be within the record.

key length1 specifies, for the alternative index, the total length of the key area or the length of the first field in the key. The key can occupy a single area within the record or be divided into two or three fields. The total key length of all fields cannot exceed 120 bytes.

physical file name specifies the physical file for which the alternative index is being created. The physical file must be a resident file.

mmddyy, ddmmyy, or yymmdd specifies the date the physical file was created. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If this parameter is not specified and more than one file exists with the name, the latest file is used.

DUPKEY specifies that duplicate index keys are allowed in the alternative index file. If no parameter is entered, DUPKEY is assumed.

NODUPKEY specifies that duplicate index keys are not allowed in the alternative index file.

A1, A2, A3 or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and the file is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, the file is placed on the least used disk unit.

block number specifies the location of the first block in the file. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk. Use the block number parameter with caution because the COMPRESS and RESTORE procedures can cause a file to be moved from the block number specified when the file was created. See the "FILE OCL Statement (for Disk Files)" on page 5-32 for more information about locating a file by a block number. If space is not available, an error message will be issued.

key position2 specifies, for the alternative index, the starting position of the second field in the key. The position must be a number from 1 through 4096. A key field cannot overlap another key field. If this parameter is not specified, no values may be specified for the third key field or third key length, and the key must occupy a single area within the record.

key length2 specifies, for the alternative index, the length of the second field in the key. The total key length of all fields cannot exceed 120 bytes. A key field cannot overlap another key field. If this parameter is not specified, no values may be specified for the third key field or third key length, and the key must occupy a single area within the record.

key position3 specifies, for the alternative index, the starting position of the third field in the key. The position must be a number from 1 through 4096. A key field cannot overlap another key field. If this parameter is not specified, the key either occupies a single area within the record or is divided into the fields specified by the first two key position and key length parameters.

key length3 specifies, for the alternative index, the length of the third field in the key. The total key length of all fields cannot exceed 120 bytes. A key field cannot overlap another key field. If this parameter is not specified, the key either occupies a single area within the record or is divided into the fields specified by the first two key position and key length parameters.

Example 1

This example shows how to create an alternative index file from a physical file named CUSTOMER. The CUSTOMER file is keyed on a customer number in positions 1 through 4 of the record. The alternative index file is to be keyed on the customer's name, which is in positions 5 through 20 of the record (a key length of 16), is to be named CUSTNAME, and is to have a preferred location of A2.

```
BLDINDEX CUSTNAME,5,16,CUSTOMER,,,A2
```

Example 2

This example shows how to create an alternative index file with a key divided into two fields from a physical file named EMPLOYEE. The EMPLOYEE file is a direct, delete-capable file. The alternative index file is to be keyed on the employee's department number, which is in positions 40 through 44 of the record (a key length of 5), and employee number, which is in positions 1 through 8 of the record (a key length of 8). The alternative index file is to be named EMPLDEPT, and is to be placed on the least-used disk unit.

```
BLDINDEX EMPLDEPT,40,5,EMPLOYEE,,,,1,8
```

Example 3

This example shows how to create an alternative index file with a key divided into three fields from a physical file named INVNTRY, which has a creation date of June 25, 1984. The INVNTRY file is an indexed file that is keyed on the part number, positions 1 through 8 of the record (a key length of 8). The alternative index file is to be keyed on the date each item of inventory was last ordered, which is in positions 64 through 69 of the record (a key length of 6), the supplier code, which is in positions 70 through 75 of the record (a key length of 6), and the part number. The alternative index file is to be named LOINVTRY, and is to have a preferred location of A3.

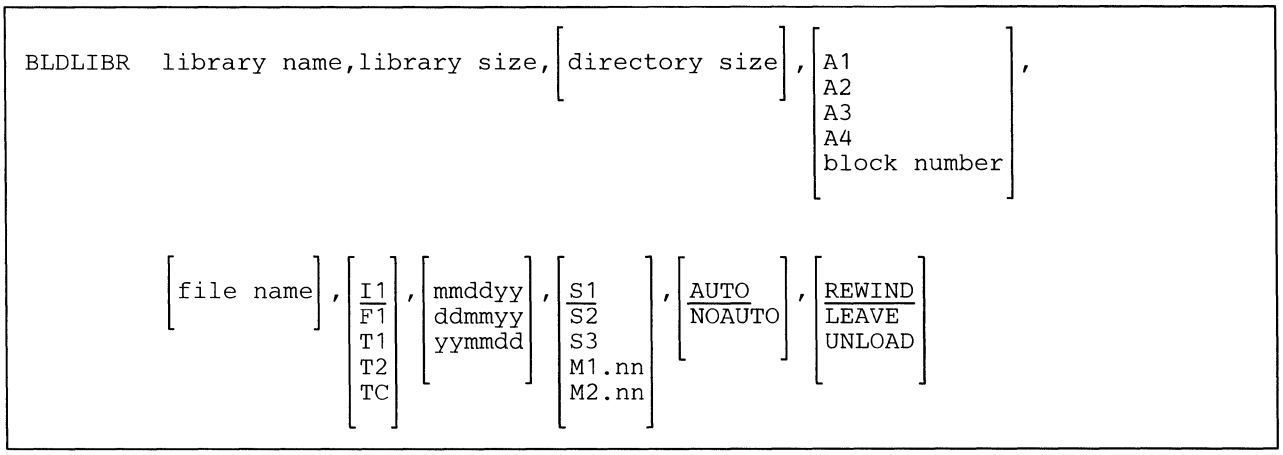
```
BLDINDEX LOINVTRY,64,6,INVNTRY,062584,,A3,70,6,1,8
```

BLDLIBR Procedure

The BLDLIBR procedure creates a new library and, optionally, copies a disk, diskette, tape, or tape cartridge file containing one or more library members into the new library.

The file containing members to be copied must have been created by the FROMLIBR procedure or by the $MAINT utility. See the "FROMLIBR Procedure" on page 4-193 or the "$MAINT Utility" on page A-55 for more information about creating this file.

The BLDLIBR procedure runs the $MAINT utility program.



S9020065-1

library name specifies the new library. A library name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because these have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), minus signs (-), and equal signs (=). Do not use #LIBRARY, F1, READER, PRINT, DISK, TAPE, or ALL as a library name.

library size specifies the size, in blocks, of the new library. One block contains 2560 bytes. The maximum library size is 15000 blocks.

directory size specifies the size, in sectors, of the directory for the new library. The minimum directory size is 2 sectors. The maximum directory size is 2500 sectors. One library block contains 10 sectors.

If a directory size is not specified, a value of 1/100 of the library size is assumed (down to a minimum of 2 sectors). For example, if you specify a library size of 100 blocks and you do not specify a directory size, 10 directory sectors are reserved (remember that 10 sectors is 1 block).

To determine the number of available directory entries for a given number of sectors, multiply the number of sectors by 5 and subtract 7 from that number. For example, for 10 directory sectors, you could have up to 43 entries (5 x 10 = 50, 50 - 7 = 43).

The directory contains the name, size, and other general information about each member in the library.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and the library is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, the library is placed on the least used disk unit.

block number specifies the preferred location of the first block of the library. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk. If space is not available, an error message will be issued.

file name specifies the file containing one or more library members to be copied into the new library.

I1 specifies that the file containing the library members to be copied is on diskette. If a file name is specified but I1, F1, T1, T2, or TC is not, I1 is assumed.

F1 specifies the file containing the library members to be copied is on disk.

T1 specifies that the file containing the library members to be copied is on tape in tape drive 1.

T2 specifies that the file containing the library members to be copied is on tape in tape drive 2.

TC specifies that the file containing the library members to be copied is on tape cartridge.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file containing the library members to be copied. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If F1 is specified and a date is not specified, the file with the specified name and the most recent creation date is used. If I1 is specified or assumed and a date is not specified, the program processes the first file in the diskette VTOC with the specified file name.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed. This parameter is valid only if the sixth parameter is I1.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is valid only if the sixth parameter is I1.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual diskette slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If this parameter is not specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must then insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the BLDLIBR procedure has run. If this parameter is not specified, REWIND is assumed. This parameter is not allowed if F1 or I1 is specified.

LEAVE specifies that the tape should be left where it is after the BLDLIBR procedure has run. The next step within a procedure that accesses a tape will start at this position if the same tape unit is specified. This parameter is not allowed if F1 or I1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the BLDLIBR procedure has run. This parameter is not allowed if F1 or I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

Example 1

Create a new library called MYLIB. It should have a size of 100 blocks. A directory size of 10 sectors is assumed.

```
BLDLIBR MYLIB,100
```

Example 2

This example shows how to create a new library and copy one or more library members from a diskette file. The library is to be named YOURLIB, be 75 blocks in size, have a directory of 15 sectors. The diskette file is named MEMBERS; the diskette is in diskette slot 1.

```
BLDLIBR YOURLIB,75,15,,MEMBERS
```

BLDMENU Procedure

The BLDMENU procedure generates, from source members, the library members required to display a menu. A menu allows an operator to start a job by selecting a menu option number instead of entering a command or procedure. Up to 24 item numbers (1 through 24) can be defined in a menu.

Information about how to build menus can be found in the manual *Creating Displays*.

Two types of menus can be created by BLDMENU: free-format and fixed-format.

To display a menu, the operator can enter the menu name on the sign on display or use the MENU command. To remove a currently displayed menu, the operator enters a 0 (zero) instead of an option number.

The BLDMENU procedure runs the $BMENU utility programs. It also runs the $MAINT utility program (LIBRLIBR procedure) and the $MGBLD utility program (CREATE procedure).

The input to the BLDMENU procedure is:

- A second-level message source member called a **command source member**. The command source member is required and contains the commands and procedures used as input when the operator selects items from the menu.

- A first-level message member called a **menu text source member**. The menu text source member is required for free-format menus, but is optional for fixed-format menus. It defines the text that appears on the menu.

The output from the BLDMENU procedure is:

- A second-level message load member, called a **command load member**, that BLDMENU creates from the command source member. BLDMENU places the command load member in the output library specified on the BLDMENU procedure. This member must be created to display a menu.

- A first-level message load member, called a **menu text load member**, that BLDMENU creates from the optional menu text source member. BLDMENU places the menu text load member in the output library specified on the BLDMENU procedure. This member is used to create the display format load member for the menu.

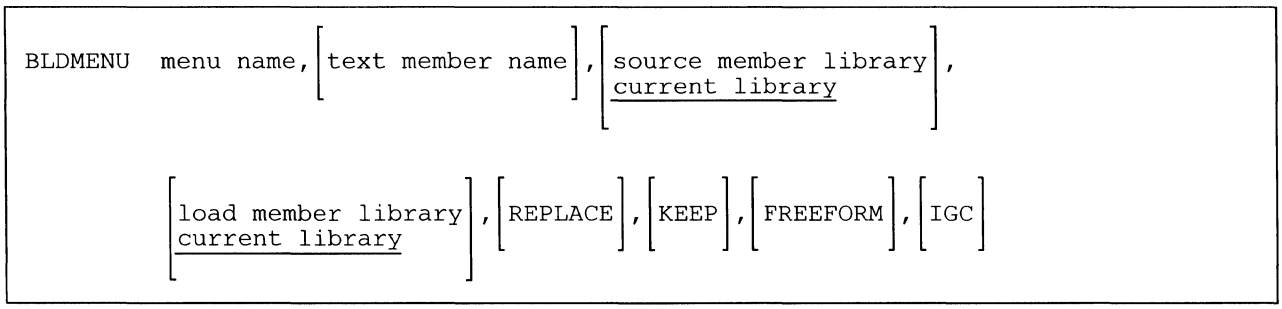
- The **display format load member** for the menu. BLDMENU places the display format load member in the output library specified on the BLDMENU procedure. This member must be created to display a menu.

- A listing that contains:

  - A partial $SFGR listing containing warning and terminal messages. If no errors were found during the compile of the menu, a $SFGR listing will not be printed.

  - The item numbers and the corresponding statements from the command source member.

  - The actual display text that will appear on the display.



S9020066-0

menu name specifies the 1- to 6-character name of the menu (the name given to the display format load member for the menu). The names of the command text source member and the command text load member (the name specified in the first line of the command text source member) must be **menu name##**. For example, if the name of a menu is MENU, the name of the command text load member must be MENU##. Because ## is added to the menu name by BLDMENU, you will receive an error message if the menu name has more than 6 characters.

text member name specifies the 1- to 8-character name of the menu text source member, if one exists. The name cannot be the same as the menu name. The name of the menu text load member (the name specified in the first record of the menu text source member) must be the same as the name of the source member. The text name is optional only if FREEFORM is not requested. If the text name is not specified, BLDMENU uses the information in the command load member to generate the descriptive text for the items in the fixed-format menu.

Note: If you are using the BLDMENU procedure to update a menu that was created using the Screen Design Aid (SDA), use only the name of the menu for the text member name parameter. In addition, you must specify the KEEP parameter. For more information, see the manual ***Creating Displays: Screen Design Aid and System Support Program****, in Chapter 10, "Creating Menus Without SDA."*

source member library specifies the library that contains the source statements. If an input library is not specified, the current library is assumed.

load member library specifies the library that will contain the display format load member for the menu, the command load member, and, if KEEP is specified, the menu text load member. If an output library is not specified, the current library is assumed.

Note: If the input library and the output library are not the same, BLDMENU copies the input source members to the output library when it begins processing. Thus, when BLDMENU is run, the output library must have enough space to contain the two input source members, as well as the two load members created by BLDMENU. BLDMENU removes the source members from the output library before ending.

REPLACE specifies that:

- If a display format load member already exists in the output library with the same name as the menu being created, BLDMENU automatically deletes the existing member from the output library.
- If a load member that is not a display format load member already exists in the output library with the same name as the menu being created, BLDMENU displays an error message. The operator must then decide whether to replace the existing member in the output library or to cancel the job.

- If the name of the command text source member or the menu display text source member is the same as the name of a source member or load member in the output library, BLDMENU automatically replaces the existing members in the output library.

If REPLACE is not specified, the following occurs:

- If a load member already exists in the output library with the same name as the menu being created, BLDMENU displays an error message. The operator must then decide whether to delete the existing member in the output library or to cancel the job.

- If the name of the command text source member or the menu text source member is the same as the name of a source member in the output library, BLDMENU displays an error message and cancels the job.

Note: If a menu is rebuilt while it is being displayed at a display station, you should request the rebuilt menu by entering a MENU control command. This must be done because the old version of the menu is still displayed, and it may not be the same as the command input defined for the new menu.

KEEP specifies that if the text member name parameter is also specified, the display text load member created by BLDMENU is to remain in the output library. If KEEP is not specified, the menu text load member is deleted from the output library before BLDMENU ends.

The menu text load member is only used by the $BMENU utility program to create the display format load member and is not needed after the menu has been built.

FREEFORM specifies that a free-format menu is to be created. If FREEFORM is not specified, a fixed-format menu is created. If FREEFORM is specified, a text name must also be specified.

IGC specifies that the system-generated text (for example, the ENTER NUMBER, COMMAND, OR OCL prompt) should be displayed as ideographic characters and ideographic data can be entered into the input field of the menu screen. The IGC parameter is for the ideographic version of the SSP and is ignored for nonideographic systems.

Notes:

1. *If you try to print output with ideographic headings at a nonideographic printer, blanks are printed in place of the ideographic characters.*

2. *If the menu screen is printed, the READY FOR OPTION NUMBER OR COMMAND prompt contains extraneous blanks between some of the ideographic characters. These blanks are not shown when the menu screen is displayed, however.*

CAUTION
If you build a menu and the IGC parameter is used, the menu cannot be displayed from a nonideographic display station.

Example

You have created a command text source member named MENU2## and a menu text source member named MENU2DT; both members are stored in a library named MYLIB. To build a menu named MENU2 using the BLDMENU procedure, you could enter the following. The library members created by BLDMENU are to be placed in a library called MYLIB, and the menu text load member created by BLDMENU is to be deleted from MYLIB before BLDMENU ends.

```
BLDMENU MENU2,MENU2DT,MYLIB,MYLIB,,,FREEFORM
```

BUILD Procedure

The BUILD procedure allows you to display and correct data on the disk after a disk error occurs. If password security is active, BUILD can be run only by an operator with a security classification of system operator or higher; or by an operator at a system service display station. The BUILD procedure can be run from any display station, but only one operator can run the BUILD procedure at a time.

The BUILD procedure searches all the sectors of the disk for data that was not readable because of a read error. When it finds an error, each sector containing unreadable data is printed, along with the sector logically preceding and the sector logically following the sector containing unreadable data. Each sector containing unreadable data is also displayed. Using the BUILD display, you can then leave the data as it is or correct the data.

If the COMPRESS procedure was being run when the disk error occurred, and you subsequently run the BUILD procedure, the COMPRESS procedure is automatically called by the BUILD procedure to finish the original disk compress.

The BUILD procedure runs the $BUILD utility program.

```
BUILD
```

S9020067-0

The BUILD procedure has no parameters.

Example

To check a file after a disk file input/output error, the system operator would enter the following:

```
BUILD
```

The data is displayed and printed in both character and hexadecimal format. The data is displayed in character format on the first line. If the character cannot be displayed, it is replaced with a blank. The data is displayed in hexadecimal format on the second and third lines. The leftmost hexadecimal digit is displayed on the second line; the rightmost hexadecimal digit is displayed on the third line. For example:

```
6686FF FFD201C6 C9D3C540 40404000 H 00E20000 01000008 00050800 0202C4C5 H *  f

FFFFCC4CCCFFFCF4CFCFCFFF4FFFFFFFFF3COFFCFFFFF4FTFF+FFF4FFFFTTTF4CFFFCFCF3C0546FF
66866606642013603943354004040400048400520000001000008000050800042023435485C066FF

CUL=00001      SS-0348901      FILENAME-DATAMAST
          Cmd1 - Rewrite sector      Cmd3 - Bypass sector
```

Displayed below the character and hexadecimal data is the FILENAME where the sector with unreadable data occurs. A file name of NOFILE indicates that the sector with unreadable data is not in a file, folder, or library. The unreadable data is in unused space on the disk. If #CSLIB is displayed, you may need to reload the system microcode. If #LIBRARY is displayed, you may need to reload the SSP.

After the sector with unreadable data is displayed, you can do one of the following:

- Press command key 3 to cause the next sector with unreadable data to be displayed. The sector that was displayed is not changed, and will be displayed again the next time the BUILD procedure is run (unless a job is run that writes data to that sector, thus fixing the unreadable data).

- Correct the unreadable data by using the Roll keys to display the portion of the sector containing the data to be corrected. Enter the correct data on either the character line or the hexadecimal lines.

  After you have corrected the sector, press command key 1. The corrected sector is written back to the disk, and the next sector with unreadable data is displayed (if any). The corrected sector will not be displayed the next time the BUILD procedure is run.

CACHE Procedure

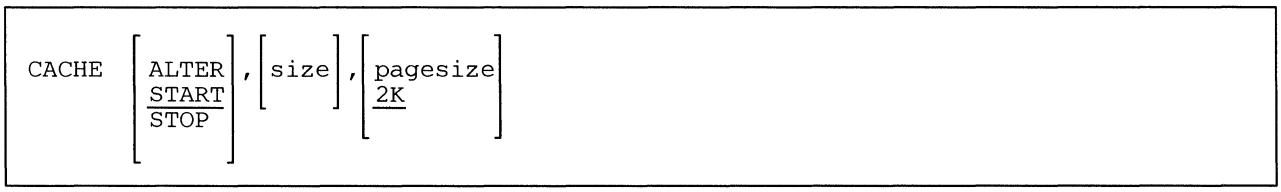
The CACHE procedure allows the system operator to create, change, or delete the disk cache. The cache is a buffer used to keep disk data in main storage to reduce the number of accesses to the disk.

As disk read operations are issued, the cache is first checked to determine if it contains the requested data. If so, the data is simply moved to the user's buffer. If not, an entire cache page containing the requested data is read into the cache and then the requested data is moved into the user's buffer.

Disk write operations will update both the disk and the cache buffer.

When the CACHE procedure is used to alter the cache, the cache is freed and rebuilt with the new size and page size. When the CACHE procedure is used to stop the cache, the memory is freed for general use.

The CACHE procedure runs the $SVCASRV utility program. The CACHE procedure cannot be evoked or run from the job queue.



ALTER specifies that the memory or page size of the cache is to be altered.

START specifies that the cache is to be started. If this parameter is not specified, START is assumed.

STOP specifies that the cache is to be stopped. Size and pagesize are not valid with STOP.

size specifies the size of the cache in kilobytes. For systems with 2 MB or less of main storage, the minimum size that can be specified is 64 kilobytes. For systems with more than 2 MB of main storage, the minimum size that can be specified is 1/32 of the main storage size.

pagesize specifies the size of the individual cache pages in kilobytes. A cache page is the smallest amount of contiguous disk data that can be held in the cache. The valid page sizes that can be specified are 1, 2, 4, 8, or 16 kilobytes. If this parameter is not specified, 2 kilobytes are assumed.

Note: For systems with 2 MB or less of main storage, the size and pagesize parameters should be specified so that there are at least 32 cache pages. For systems with more than 2 MB of main storage, the size and pagesize parameters should be specified so that the number of cache pages is at least 1/64 of the main storage size.

Example

This example shows how to start a 64 kilobyte cache with a page size of 2 kilobytes.

```
CACHE START,64,2
```

CATALOG Procedure

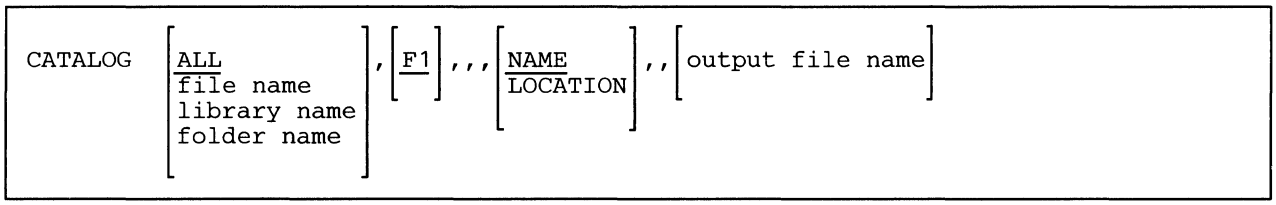
The CATALOG procedure lists the names of the files, libraries, and folders on disk, diskette, tape, or tape cartridge. You can list all the entries on disk, diskette, tape, or tape cartridge; or list a single entry. You can also list the labels of files defined in your system network resource directory (NRD). The CATALOG procedure reads the volume table of contents (VTOC) for disk and diskette. The disk VTOC contains an entry for each file, library, and folder on the disk, and a diskette VTOC contains an entry for each file on the diskette. Each VTOC entry identifies the related file, library, or folder by name, creation date, and location. The CATALOG procedure reads the volume label and header label information contained on a standard labeled tape or tape cartridge.

The output from the CATALOG procedure can be directed to the system list device or, if the unit parameter is F1, to a disk file. If an output file name is specified, output is directed to that file; otherwise, output is directed to the system list device. To determine your system list device, enter the STATUS SESSION command. To change the system list device, see the "SYSLIST Procedure" on page 4-498 or the "PRINT Procedure" on page 4-350.

The CATALOG procedure runs the $LABEL utility program.

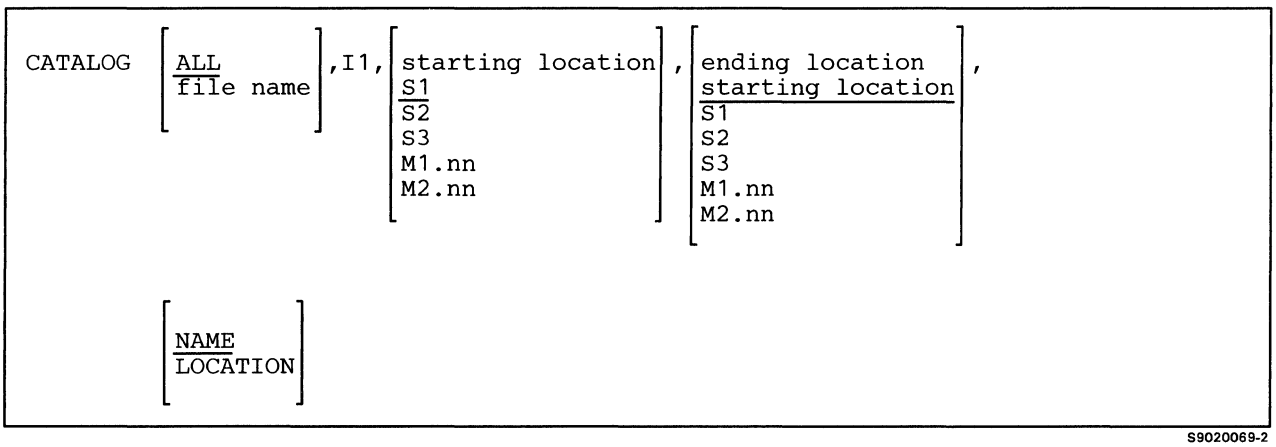
If you try to print with ideographic headings at a nonideographic printer, blanks are printed in place of the ideographic characters. The labels of files defined in your system network resource directory will be listed when disk entries are listed. To list only the network resource directory entries, use the LISTNRD procedure.

To list **disk** information:



S9020068-1

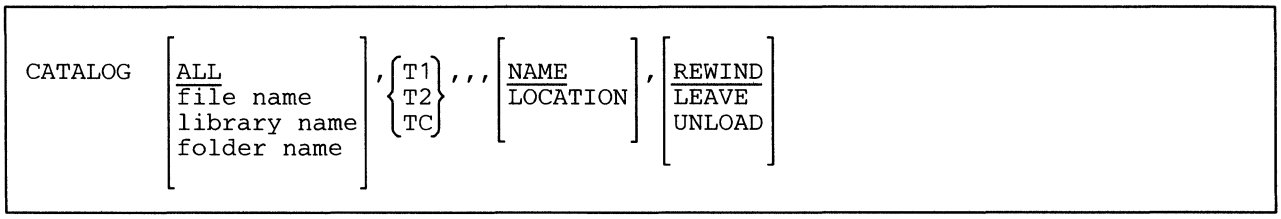
To list **diskette** information:



S9020069-2

CATALOG

To list **tape** or **tape cartridge** information:



S9020070-4

ALL specifies that all file, library, and folder names are to be listed. If the first parameter is not specified, ALL is assumed.

file name or library name or folder name specifies the file, library, or folder whose VTOC information is to be listed. For tape, the label information is listed. If more than one file exists with the specified name, the information for all files with the specified name is listed.

F1 specifies that VTOC entries for the disk are to be listed. If the second parameter is not specified, F1 is assumed. See "Sample Disk VTOC Listings" on page 4-77 for a description of the items on the listing.

I1 specifies that VTOC entries for a diskette are to be listed. See "Sample Diskette VTOC Listings" on page 4-84 for a description of the items on the listing.

T1 specifies that label information for the tape mounted on tape drive 1 is to be listed. See "Sample Tape Label Listings" on page 4-87 for a description of the items on the listing.

T2 specifies that label information for the tape mounted on tape drive 2 is to be listed. See "Sample Tape Label Listings" on page 4-87 for a description of the items on the listing.

TC specifies that label information for the tape cartridge is to be listed. See "Sample Tape Label Listings" on page 4-87 for a description of the items on the listing.

NAME specifies that entries are to be sorted and listed by name in alphabetical order. If the fifth parameter is not specified, NAME is assumed.

LOCATION specifies that entries are to be sorted and listed by their location sequence on the disk, diskette, or tape.

starting location specifies the starting location of a diskette slot to be used by CATALOG. If this parameter is not specified, S1 is assumed. This parameter is not allowed if F1, T1, T2, or TC is specified; it is ignored if the system does not have a diskette magazine drive. The starting location must be less than or equal to the ending location.

ending location specifies the ending location of a diskette slot to be used by CATALOG. If this parameter is not specified, the starting location is assumed. This parameter is not allowed if F1, T1, T2, or TC is specified; it is ignored if the system does not have a diskette magazine drive.

The ending location (if specified) must be greater than or equal to the starting location and must not cause a jump from the diskette slots (S1, S2, and S3) to the magazine slots (M1 and M2). For example, specifying a starting location of S3 and an ending location of M1 is not correct. You can specify a starting location in magazine 1 and an ending location in magazine 2.

The starting and ending locations can be any of the following:

S1, S2, or S3 specifies one of the three diskette slots.

M1 or M2 specifies one of the two diskette magazine slots. If M1 or M2 is specified for the starting location and no ending location is specified, the entire magazine (all 10 positions) is processed. If M1 or M2 is specified for the ending location, it is the same as specifying M1.10 or M2.10, respectively.

M1.nn or M2.nn specifies a position within a magazine. **nn** is a number from 01 through 10, specifying the position within the magazine.

REWIND specifies that the tape should be rewound after the CATALOG procedure has run. REWIND is assumed if this parameter is not specified. This parameter is not allowed if F1 or I1 is specified.

LEAVE specifies that the tape should be left where it is after the CATALOG procedure has run. The next step within a procedure that accesses a tape starts at this position if the same tape unit is specified. This parameter is not allowed if F1 or I1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the CATALOG procedure has run. This parameter is not allowed if F1 or I1 is specified. If UNLOAD and TC are specified, the tape is rewound after processing.

output file name specifies the name of the disk file to which the output of the CATALOG procedure is to be written. The file name can have from one to eight alphameric characters and must not contain blanks, commas, or quotes. ALL is not a valid file name. If not specified, output is written to the SYSLIST device. The specified file is created as a sequential file on the local system. The size is determined by the size of the VTOC to be listed. The record length is 132 bytes. This parameter is not valid for listing diskette or tape information.

Note: A file or library with the specified name and the current date must not already exist on the disk. Output to a disk file is only valid when the second parameter is F1.

Example 1

Display VTOC information for the disk file called PAYROLL:

```
CATALOG PAYROLL
```

Example 2

List, by name, the names of the files on a diskette in slot 1:

```
CATALOG ALL,I1
```

Example 3

List, by name, the names of all the files, libraries, and folders on the disk:

```
CATALOG
```

Example 4

List, by location, the names of all the files in magazine slot 1:

```
CATALOG ,I1,M1,,LOCATION
```

Example 5

List, by location, the names of all the files in positions 1 through 5 of magazine slot 2:

```
CATALOG ,I1,M2.01,M2.05,LOCATION
```

Example 6

List, by name, the names of all files on a tape on tape drive 1 and rewind the tape when done:

```
CATALOG ALL,T1
```

Example 7

List, by location, the names of all the files on a tape on tape drive 2 and leave the tape positioned after the last file:

```
CATALOG ,T2,,,LOCATION,LEAVE
```

Example 8

List, by name, the names of all files on the disk, and direct the output to a file named MYLIST:

```
CATALOG ,,,,,,MYLIST
```

Sample Disk VTOC Listings

The following listings show sample disk VTOC printouts. The first example shows the VTOC listed by name and the second example shows the VTOC listed by location. Following the examples is a description of the items on the listings.

Disk VTOC Listing by Name

```
                                        D I S K   V T O C   D I S P L A Y                                       PAGE    1
PACK - SGSSP      OWNER ID - 850401              B Y   F I L E N A M E
                                                                                                                DATE  - 85/04/04
DEVICE CAPACITY -   60.02  MEGABYTES  /  23448 BLOCKS                                                          TIME  - 14.00

STATUS CODES   1 - DELETE CAPABLE            3 - DUP KEYS NOT ALLOWED     5   PREVIOUSLY SECURED          7 - EXTERNALLY DESCRIBED
               2 - DUP KEYS ALLOWED          4   SECURED                  6 - ACTIVE TRACE FILE          8   ON REMOTE SYSTEM

---- -- - ------ --FILE - --- --- -- - -     -  ---RECORDS  - -     KEY -     ALLOCATION      PARENT OR  -- LOCATION --
 LABEL      DATE      ORG    TYPE     STATUS    LEN     USED  AVAILABLE  POS  LEN  RECORDS BLOCKS   NO. ALTER  BLOCK NO.  PREF
##JOBQ     85/04/04          SYSTEM                                                         6                 23686
##SPOOL1   85/04/04          SYSTEM                                                       400                 23692    A2
#LIBRARY                     SYSTEM                                                      8500                  2238
#MESSAGE   85/04/04          SYSTEM                                                        25                 23661    A2
#SERVLOG   85/04/02          SYSTEM                                                         5                 23352
#SYSHIST   85/04/02          SYSTEM                                                        12                   676
#SYSTASK   85/04/02          SYSTEM                                                      1550                   688
#SYSWORK   85/04/02          SYSTEM                                                        26                   650
#NRD.FLE   85/04/04    T               1,3                              3    9              2                 22389
#CNFGLIB                     LIBRARY                                                        6                 24092
#DUMP.00   85/04/04    D     DUMPFILE                                                      34                 22355
#DUMP.01   85/04/04    D     DUMPFILE                                                      36                 22319
DICTJAA    84/09/24   DCT    FOLDER                                                       178                 22974
DICTJAA               DCT    FOLDER                                                       200                 22774
DIRFILE    85/04/04    D                          512     1000       0              1000  200                 22391    A2
LMPJAA     84/09/24    S               7           49        0     104               104    2                 22641
FILE.1                                 8                                                                    CHICAGO
FILE.2                                 8                                                                    NEWYORK
FILE.3                                 8                                                                    CHICAGO
FILE.4                                 8                                                                    CHICAGO
FILE.5                                 8                                                                    NEWYORK
FILLA      85/04/04    I               2          256      234     237      1   12             50                 22591
HISTCOPY   85/04/04    S                          256       93       7               100       10                 22307
JAAMAIL    84/09/26   MLG    FOLDER                                                        11                 22763
JAASAVE                      LIBRARY                                                      100                 23152
JPROCS                       LIBRARY                                                      100                 10738
NEWMSGLB                     LIBRARY                                                      100                 23252
PLDFILE    85/04/02    D                          128       80       0                80    4                 23657
SSI02                        LIBRARY                                                      300                 23357
TXTJAA     84/09/21   DOC    FOLDER                                                        20                 22743
TXTJAA                DOC    FOLDER                                                       100                 22643
XFILE.A    85/04/04    T               2          512      164       0      2   14             34                 22223
XFILE.1    85/04/04    S                          128      237     763                     50                 22257    A4
XFILE.5    85/04/04    D                          256     1000       0              1000  100                 22123

SPACE CURRENTLY AVAILABLE              LOCATION   BLOCKS
                                         10838    11265
                                         22317        2
                                                  11267
USER VTOC ENTRIES - USED     21 / AVAILABLE       119
```

```
                                         J O B   F I L E   D I S P L A Y                                         PAGE    2
  JOBNAME -   JOBCAT
                                                                                                                 DATE  - 85/04/04
DEVICE CAPACITY -   60.02  MEGABYTES  /  23448 BLOCKS                                                           TIME  - 14.00

STATUS CODES   1 - DELETE CAPABLE            3 - DUP KEYS NOT ALLOWED     5   PREVIOUSLY SECURED          7   EXTERNALLY DESCRIBED
               2 - DUP KEYS ALLOWED          4 - SECURED                  6 - ACTIVE TRACE FILE          8 - ON REMOTE SYSTEM

------------------- FILE-------------- ------   - --   RECORDS-- --    -KEY -     ALLOCATION--   PARENT OR     LOCATION  -
 LABEL      DATE      ORG    TYPE     STATUS    LEN     USED  AVAILABLE  POS  LEN  RECORDS BLOCKS   NO. ALTER  BLOCK NO.  PREF
JOB2       85/04/04    I               1,3         50        0     317      1   10     317      8                 22103
JOB1       85/04/04    S                          100        0     307             307     12                 22111
*****   END OF VTOC DISPLAY    *****
```

```
                   NETWORK RESOURCE DIRECTORY                PAGE   1
                        BY LOCAL LABEL
                                                             DATE - 85/04/04
DIRECTORY LABEL - #NRD.FLE                                   TIME - 14.00

                  LOCAL        RESOURCE      REMOTE        REMOTE
                  LABEL           ID         LOCATION      LABEL
                 --------     ---------     --------      --------
                 FILE.1       DATAFILE      CHICAGO       CHI.1
                 FILE.2       DATAFILE      NEWYORK       NY.1
                 FILE.3       DATAFILE      CHICAGO       CHI.2
                 FILE.4       DATAFILE      CHICAGO       CHI.3
                 FILE.5       DATAFILE      NEWYORK       NY.2

                 NRD ENTRIES - USED      5/ AVAILABLE      29
                **** END OF NETWORK RESOURCE DIRECTORY DISPLAY ****
```

Disk VTOC Listing by Location

```
                                                      D I S K    V T O C    D I S P L A Y                                PAGE     1
PACK - SGSSP      OWNER ID - 850401                          B Y     L O C A T I O N
                                                                                                                        DATE - 85/04/04
DEVICE CAPACITY -    60.02  MEGABYTES  /  23448 BLOCKS                                                                 TIME - 14.00

STATUS CODES    1 - DELETE CAPABLE              3 - DUP KEYS NOT ALLOWED      5 - PREVIOUSLY SECURED         7 - EXTERNALLY DESCRIBED
                2 - DUP KEYS ALLOWED            4 - SECURED                   6 - ACTIVE TRACE FILE          8 - ON REMOTE SYSTEM

---------------------FILE--------------------------  -----------RECORDS-----------  --KEY---  ---ALLOCATION--  PARENT OR  ---LOCATION--
 LABEL       DATE     ORG     TYPE      STATUS       LEN       USED   AVAILABLE     POS   LEN  RECORDS BLOCKS  NO. ALTER  BLOCK NO. PREF
#SYSWORK   85/04/02          SYSTEM                                                                      26                    650
#SYSHIST   85/04/02          SYSTEM                                                                      12                    676
#SYSTASK   85/04/02          SYSTEM                                                                    1550                    688
#LIBRARY                     SYSTEM                                                                    8500                   2238
JPROCS                       LIBRARY                                                                    100                  10738
XFILE.5    85/04/04   D                              256       1000        0                    1000    100                  22123
XFILE.A    85/04/04   I                  2           512        164        0          2    14             34                  22223
XFILE.1    85/04/04   S                              128        237      763                             50                  22257     A4
HISTCOPY   85/04/04   S                              256         93        7                     100     10                  22307
#DUMP.01   85/04/04   D      DUMPFILE                                                                    36                  22319
#DUMP.00   85/04/04   D      DUMPFILE                                                                    34                  22355
#NRD.FLF   85/04/04   I                  1,3                                          3     9              2                  22389
DIRFILE    85/04/04   D                              512       1000        0                    1000    200                  22391     A2
FILEA      85/04/04   I                  2           256        234      237          1    12             50                  22591
EMPJAA     84/09/24   S                  7            49          0      104                     104      2                  22641
TXTJAA               DOC     FOLDER                                                                     100                  22643
TXTJAA     84/09/21  DOC     FOLDER                                                                      20                  22743
JAAMAIL    84/09/26  MLG     FOLDER                                                                      11                  22763
DICTJAA              DCT     FOLDER                                                                     200                  22774
DICTJAA    84/09/24  DCT     FOLDER                                                                     178                  22974
JAASAVE                      LIBRARY                                                                    100                  23152
NEWMSGLB                     LIBRARY                                                                    100                  23252
#SERVLOG   85/04/02          SYSTEM                                                                       5                  23352
SS102                        LIBRARY                                                                    300                  23357
PLIFILE    85/04/02   D                              128         80        0                      80      4                  23657
#MESSAGE   85/04/04          SYSTEM                                                                      25                  23661     A2
##JOBQ     85/04/04          SYSTEM                                                                       6                  23686
##SPOOL1   85/04/04          SYSTEM                                                                     400                  23692     A2
#CNFGLIB                     LIBRARY                                                                      6                  24092

SPACE CURRENTLY AVAILABLE   --    LOCATION    BLOCKS
                                    10838      11265
                                    22317          2
                                               11267
USER VTOC ENTRIES   USED    21 / AVAILABLE    119
```

```
                                                      J O B     F I L E     D I S P L A Y                                PAGE     2
  JOBNAME -  JOBCAT
                                                                                                                        DATE - 85/04/04
DEVICE CAPACITY -    60.02  MEGABYTES  /  23448 BLOCKS                                                                 TIME - 14.00

STATUS CODES    1 - DELETE CAPABLE              3 - DUP KEYS NOT ALLOWED      5 - PREVIOUSLY SECURED         7 - EXTERNALLY DESCRIBED
                2 - DUP KEYS ALLOWED            4 - SECURED                   6 - ACTIVE TRACE FILE          8 - ON REMOTE SYSTEM

---------------------FILE--------------------------  -----------RECORDS-----------  --KEY---  ---ALLOCATION--  PARENT OR  ---LOCATION--
 LABEL       DATE     ORG     TYPE      STATUS       LEN       USED   AVAILABLE     POS   LEN  RECORDS BLOCKS  NO. ALTER  BLOCK NO. PREF
JOB2       85/04/04   I                  1,3          50          0      317          1    10     317      8                  22103
JOB1       85/04/04   S                              100          0      307                     307     12                  22111
*****   END OF VTOC DISPLAY   *****
```

```
                      NETWORK RESOURCE DIRECTORY                 PAGE  1
                           BY LOCATION ID
                                                                 DATE   85/04/04
DIRECTORY LABEL  - #NRD.FLF                                      TIME - 14.00

                LOCAL          RESOURCE        REMOTE          REMOTE
                LABEL             ID           LOCATION         LABEL
               --------        -------         --------        -------
               FILE.1          DATAFILE        CHICAGO         CHI.1
               FILE.3          DATAFILE        CHICAGO         CHI.2
               FILE.4          DATAFILE        CHICAGO         CHI.3
               FILE.2          DATAFILE        NEWYORK         NY.1
               FILE.5          DATAFILE        NEWYORK         NY.2

               NRD ENTRIES  - USED       5/ AVAILABLE      29
             **** END OF NETWORK RESOURCE DIRECTORY DISPLAY ****
```

Disk VTOC Listing by Single Name

```
                                          D I S K   V T O C   D I S P L A Y                                     PAGE    1
PACK - 8088P      OWNER ID   840926
                                                                                                                DATE  - 10/01/84
DEVICE CAPACITY - 398.74 MEGABYTES  / 155758 BLOCKS                                                             TIME  - 10.58

STATUS CODES   1 - DELETE CAPABLE          3   DUP KEYS NOT ALLOWED      5 - PREV SECURED FILE/LIBR      7 - EXTERNALLY DESCRIBED
               2   DUP KEYS ALLOWED        4   SECURED FILE/LIBR         6 - ACTIVE TRACE FILE

-------------- - -   FILE -          -           - -   - RECORDS          KEY       ALLOCATION -    PARENT OR   ---LOCATION---
 LABEL        DATE      ORG    TYPE    STATUS    LEN     USED  AVAILABLE  POS  LEN  RECORDS BLOCKS   NO. ALTER   BLOCK NO. PREF
EMPJAA     09/24/84    S               7         49       0      104                   104     2          01        80055
     DICTIONARY - DICTJAA           DEFINITION    EMPJAA
EMPJAA2    09/24/84    X               2         49       0      104     NO   31                   3   EMPJAA          80052
                                                                           9   20
                                                                          29   10
                                                                          39    1

SPACE CURRENTLY AVAILABLE    --    LOCATION    BLOCKS
                                    55544      22674
                                    78975         73
                                    79989          7
                                    80028          4
                                               22758
USER VTOC ENTRIES - USED  557 AVAILABLE 1245



                                          J O B   F I L E   D I S P L A Y                                       PAGE    2
   JOBNAME -   CATLIST
                                                                                                                DATE  - 10/01/84
DEVICE CAPACITY    398.74 MEGABYTES  / 155758 BLOCKS                                                            TIME    10.58

STATUS CODES   1 - DELETE CAPABLE          3 - DUP KEYS NOT ALLOWED      5   PREV SECURED FILE/LIBR      7 - EXTERNALLY DESCRIBED
               2 - DUP KEYS ALLOWED        4   SECURED FILE/LIBR         6   ACTIVE TRACE FILE

---------------- --FILE --         -    --              RECORDS          KEY       ALLOCATION      PARENT OR   --LOCATION
 LABEL        DATE      ORG    TYPE    STATUS    LEN     USED  AVAILABLE  POS  LEN  RECORDS BLOCKS   NO. ALTER   BLOCK NO. PREF
JOB2       10/01/84    I               1,3        50      0      317       1   10    317      8                 80032
JOB1       10/01/84    S                         100      0      307                 307     12                 80040
*****    END OF VTOC DISPLAY    *****
```

Description of Items on the Disk VTOC Listing

PAGE indicates the page number of the printout.

PACK specifies the volume ID of the disk volume label.

OWNER ID specifies the owner ID of the disk volume label.

JOB NAME specifies the name of the job from which CATALOG was run. This appears only when a job file exists.

DATE specifies the system date in the current format (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

DEVICE CAPACITY specifies the size of the disk in megabytes and in blocks (one megabyte equals one million bytes).

TIME specifies the time.

STATUS CODES specify status codes about the files, libraries, and folders on disk.

1 The file is a delete-capable file.

2 Duplicate keys are allowed in the file.

3 Duplicate keys are not allowed in the file.

4 The file, library, or folder is secured by resource security.

5 The file, library, or folder was secured by resource security, but resource security is not currently active.

6 The file is an active trace file.

7 The file is externally described.

8 The file is on a remote system.

FILE LABEL specifies the label of the file, library, or folder.

FILE DATE specifies when the file was created. For libraries and folders, this field is blank.

FILE ORG specifies how the file is organized:

S	Sequential
D	Direct
I	Indexed
X	Alternative index
Blank	System file or a library

or the type of folder:

DCT Dictionary

DOC Document

MLG Mail log

PRF Profile

ML Mail

DWK Document work

FILE TYPE specifies the type of file:

Blank Indicates a data file

DUMPFILE Indicates a dump file

FOLDER Indicates a folder

LIBRARY Indicates a library

LIBREXT Indicates a library extent

LIBRFILE Indicates a file created by $MAINT

PCDISK Indicates a personal computer virtual disk

RESERVE Indicates a reserve area

SYSTEM Indicates a system file

TRACEFIL Indicates a trace file

FILE STATUS specifies the status of the file, library, or folder:

1 Indicates a delete-capable file

2 Indicates duplicate keys are allowed

3 Indicates duplicate keys are not allowed

4 Indicates a secured file, library, or folder

5 Indicates a file, library, or folder that was secured, but security is not currently active.

6 Indicates an active trace file

7 Indicates an externally described file

8 Indicates a remote file

RECORD LENGTH specifies the length of the records in the file. This is a decimal number showing the number of bytes in each record. For libraries and folders, this field is blank.

RECORDS USED specifies the number of records currently containing data in the file. For libraries, folders, and system files, this field is blank.

The number of records used does not reflect any records added by a user currently using the file. The VTOC is not updated until the last user is done with the file.

RECORDS AVAILABLE specifies the number of records that can still be used in the file. For libraries, folders, and system files, this field is blank.

KEY POS specifies, for an indexed file, the starting position of the key field in the records. If the key is divided into two or three fields, NC (noncontiguous) will be shown.

KEY LEN specifies, for an indexed file, the length of the key field in the records. If the key is divided into two or three fields, the total length of the key will be shown.

Note: Additional lines may be shown for the key position and key length fields if a specific file is being listed. For alternative indexed files with keys divided into two or three fields, the starting position and length of the fields that make up the key will be shown.

ALLOCATION RECORDS specifies the number of records (in decimal) allocated for a file. The number shown may be larger than the number of records requested because the SSP allocates disk space in blocks. The number of records allocated is rounded up to the next higher block. This number is only shown if the file was allocated by records.

ALLOCATION BLOCKS specifies the number of blocks (in decimal) allocated for a file, library, or folder.

PARENT OR NO. ALTER. specifies, for a physical file that has one or more alternative index files, the number of alternative index files. For an alternative index file, this specifies the physical file that the alternative index file was built from.

Note: If a specific file is being listed and that file is a physical file, additional lines will be shown identifying its alternative indexes. If the file is an alternative index file, an additional line will be shown identifying the associated physical file.

LOCATION BLOCK NUMBER specifies the block number (in decimal) of the first block of the file, library, or folder. For information on the number of blocks of disk space available, see the *Concepts and Programmer's Guide*.

LOCATION PREF specifies the preferred disk location (disk unit A1, A2, A3, or A4), if a preferred location was specified when the file, library, or folder was created.

SPACE CURRENTLY AVAILABLE specifies the location and the number of blocks of disk space that is available. If no space is available, the following message is listed:

```
NO DISK SPACE ON THIS SYSTEM
```

USER VTOC ENTRIES specifies the number of user disk VTOC entries used and the number of entries that can still be used.

Note: If a specific file is being listed and that file is externally described, an additional line will be shown identifying the associated data dictionary and definition.

Following is a chart that indicates the beginning column and the length of the field that may be used for each item on disk VTOC listings.

Beginning Column	Field Length	Item
123	4	PAGE
8	6	PACK
29	14	OWNER ID
14	8	JOB NAME
125	8	DATE
19	7	DEVICE CAPACITY (megabytes)
41	6	DEVICE CAPACITY (blocks)
125	5	TIME
1	8	FILE LABEL
11	8	FILE DATE
22	1	FILE ORG (file organization)
21	3	FILE ORG (folder type)
26	8	FILE TYPE
36	7	FILE STATUS
45	4	RECORD LENGTH
50	8	RECORDS USED
59	8	RECORDS AVAILABLE
70	4	KEY POS
75	3	KEY LEN
79	8	ALLOCATION RECORDS
88	6	ALLOCATION BLOCKS
96	8	PARENT OR NO. ALTER.
106	9	LOCATION BLOCK NUMBER
118	2	LOCATION PREF
34	7	SPACE CURRENTLY AVAILABLE (location)
45	6	SPACE CURRENTLY AVAILABLE (size)
36	5	USER VTOC ENTRIES (used)
44	5	USER VTOC ENTRIES (available)
19	12	Dictionary
48	12	Definition

Sample Diskette VTOC Listings

The following listings show sample diskette VTOC printouts. The first example shows the VTOC listed by name and the second example shows the VTOC listed by location. Following the examples is a description of the items on the listings.

Diskette VTOC Listing by Name

```
                D I S K E T T E   D I S P L A Y   B Y   F I L E N A M E

  VOLUME ID - TEST      OWNER ID - JAA                    S1                                DATE  06/04/86     TIME  10.38

 SPACE AVAILABLE ON THIS VOLUME IS   769  SECTORS -EACH 1024 BYTES

USER VTOC ENTRIES - USED    10 / AVAILABLE    61

STATUS CODES     1 - SAVED USING SAVE ALL       2 - EXTENT       3 - COMPRESSED

      FILE          FILE    SECTORS    FILE     FILE      FILE    RECORD    EXPIRATION      MVF   SEQUENCE     CREATING
      NAME          DATE    IN FILE    TYPE     STATUS  LOCATION  LENGTH      DATE        FILE    NUMBER       SYSTEM

  DICTJAA          09/24/84     23    SAVEFLDR  3          0023      2560    PROTECT                          IBMSYSTEM36
  DICT0.99         09/24/84     19    SAVEFLDR  2,3        0046      2560    PROTECT                          IBMSYSTEM36
  EMPJAA           09/24/84      1    COPYFILE  3          0065        49    PROTECT                          IBMSYSTEM36
  EMPJAA2          09/24/84      1    COPYFILE  3          0066        49    PROTECT                          IBMSYSTEM36
  JAAMAIL          09/26/84     29    SAVEFLDR             0081      2560    PROTECT                          IBMSYSTEM36
  JFILE            10/02/84      1    COPYFILE             0113        16    PROTECT                          IBMSYSTEM36
  JPROCS           10/01/84      3    SAVELIBR             0110       128    PROTECT                          IBMSYSTEM36
  LETTERA          09/24/84      6    ARCHIVE              0067       128    PROTECT                          IBMSYSTEM36
  TXTJAA           07/23/85     51    SAVEFLDR             0114      2560    PROTECT                          IBMSYSTEM36
  TXTJ0.99         07/23/85    251    SAVEFLDR  2          0165      2560    PROTECT                          IBMSYSTEM36
*****                  END OF VTOC DISPLAY                    *****
```

Diskette VTOC Listing by Location

```
                D I S K E T T E   D I S P L A Y   B Y   L O C A T I O N

  VOLUME ID - TEST      OWNER ID - JAA                    S1                                DATE  06/04/86     TIME  10.39

 SPACE AVAILABLE ON THIS VOLUME IS   769  SECTORS -EACH 1024 BYTES

USER VTOC ENTRIES - USED    10 / AVAILABLE    61

STATUS CODES     1 - SAVED USING SAVE ALL       2 - EXTENT       3 - COMPRESSED

      FILE          FILE    SECTORS    FILE     FILE      FILE    RECORD    EXPIRATION      MVF   SEQUENCE     CREATING
      NAME          DATE    IN FILE    TYPE     STATUS  LOCATION  LENGTH      DATE        FILE    NUMBER       SYSTEM

  DICTJAA          09/24/84     23    SAVEFLDR  3          0023      2560    PROTECT                          IBMSYSTEM36
  DICT0.99         09/24/84     19    SAVEFLDR  2,3        0046      2560    PROTECT                          IBMSYSTEM36
  EMPJAA           09/24/84      1    COPYFILE  3          0065        49    PROTECT                          IBMSYSTEM36
  EMPJAA2          09/24/84      1    COPYFILE  3          0066        49    PROTECT                          IBMSYSTEM36
  LETTERA          09/24/84      6    ARCHIVE              0067       128    PROTECT                          IBMSYSTEM36
  JAAMAIL          09/26/84     29    SAVEFLDR             0081      2560    PROTECT                          IBMSYSTEM36
  JPROCS           10/01/84      3    SAVELIBR             0110       128    PROTECT                          IBMSYSTEM36
  JFILE            10/02/84      1    COPYFILE             0113        16    PROTECT                          IBMSYSTEM36
  TXTJAA           07/23/85     51    SAVEFLDR             0114      2560    PROTECT                          IBMSYSTEM36
  TXTJ0.99         07/23/85    251    SAVEFLDR  2          0165      2560    PROTECT                          IBMSYSTEM36
*****                  END OF VTOC DISPLAY                    *****
```

Description of Items on the Diskette VTOC Listing

VOLUME ID specifies the volume ID of the diskette.

OWNER ID specifies the owner ID of the diskette.

DATE specifies the system date in the current format (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

TIME specifies the time.

S1, S2, S3, M1.nn, M2.nn specify the location of the diskette in the magazine drive.

SPACE AVAILABLE ON THIS VOLUME specifies the number (in decimal) of sectors available on the diskette. The number represents the amount of space following the last unexpired file on the diskette.

USER VTOC ENTRIES specify the number of diskette VTOC entries used and the number of entries that can still be used.

FILE NAME specifies the name of the file contained on the diskette.

FILE DATE specifies the session date (that the file was saved) in the current format (mm/dd/yy, dd/mm/yy, or yy/mm/dd). If a file was saved with a SAVE ALL (file status 1), the date shown is the creation date of the file on disk.

SECTORS IN FILE specify the number (in decimal) of diskette sectors that the file uses.

FILE TYPE specifies the type of the file, as indicated by the following:

APARFILE indicates that the file was created by the APAR procedure or the $FEAIDS utility program.

ARCHIVE indicates a file that contains a folder member that was created by the ARCHIVE procedure or the $TMSERV utility program.

BACKUP indicates that the file containing the library was created by the IBM System/34 BACKUP procedure or the IBM System/34 $BACK utility program.

COPYFILE indicates that the file was created by the SAVE procedure or the $COPY utility program.

EXCHANGE indicates a basic data exchange file that was created by the TRANSFER procedure or the $BICR utility program.

IFORMAT indicates an I-exchange file that was created by the TRANSFER procedure or the $BICR utility program.

IGC EXTN indicates an ideographic extended character file that was created by the IBM System/34 XSAVE procedure or the IBM System/34 $XSAVE utility program.

LIBRFILE indicates a file, containing one or more library members, that was created by the FROMLIBR procedure or the $MAINT utility program.

OLMVFILE indicates the file is an IBM System/34 offline multivolume file.

PROFILE indicates that the file containing the library was created by the IBM System/34 PRSAVE procedure or the IBM System/34 $PRSV utility program.

SAVEFLDR indicates a file that contains a library that was created by the SAVEFLDR procedure or the $TMSERV utility program.

SAVELIBR indicates a file that contains a library that was created by the SAVELIBR procedure or the $MAINT utility program.

UNKNOWN indicates that the file is not known by the System/36.

FILE STATUS specifies status codes about the files on diskette:

1 indicates the file was saved using the SAVE ALL procedure.

2 indicates a folder extent.

3 indicates compressed data. If the file is continued on another diskette, this status code is shown only for the first volume.

FILE LOCATION specifies the number (in decimal) of the first diskette sector of the file.

RECORD LENGTH specifies the length of the records in a file. This is a decimal number indicating the number of bytes in each record. A LIBRFILE file or SAVELIBR file has a record length of 8.

EXPIRATION DATE specifies when the file on diskette is to expire (that is, when the file is no longer needed and can be removed). PROTECT indicates that the file is permanent, and can only be removed by using the DELETE procedure.

MVF FILE specifies whether the file is continued on another diskette, according to the following indicators:

blank indicates the file is contained on one diskette and is not continued.

C indicates the file is continued on one or more diskettes.

L indicates this is the last diskette containing a continued file.

SEQUENCE NUMBER specifies, for a continued file, the sequence number of the diskette. If the file is not continued, the column is blank.

CREATING SYSTEM specifies the IBM system that created the file. If the System/36 cannot tell what the creating system was, UNKNOWN is shown.

Sample Tape Label Listings

The following listings show sample tape label listings. The first example shows the labels listed by name and the second example shows the labels listed by location. Following the examples is a description of the items on the listings.

Tape Label Listing by Name

```
                                    TAPE  LABEL  DISPLAY                                          PAGE      1
                                          BY  NAME

  VOLUME ID - TEST     UNIT - T1    LABEL TYPE - SL     OWNER ID - ACCOLA                  DATE - 85/04/04    TIME - 14.01

STATUS CODES     1 - SAVED USING SAVE ALL      2 - EXTENT      3 - FILE HAS NO HDR2 LABEL

                  FILE   REC  RECORD  BLOCK  ALLOC                          CREATE   EXPIRE   MULTIVOLUME      CREATING      SET
FILE LABEL       SEQNUM  FMT  LENGTH  LENGTH  RECS   FILETYPE   STATUS      DATE     DATE     SEQNUM VOLID     SYSTEM        NAME

DICTJAA            3     FB     256   24576          SAVEFLDR              84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
DICTO.99           4     FB     256   24576          SAVEFLDR   2          84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
JAA.1              7     FB      32   24576      0   COPYFILE   1          84/12/06 PROTECT    1  TEST   IBM SYSTEM/36  JAA
JAA.2              8     FB      64   24576      0   COPYFILE   1          84/12/06 PROTECT    1  TEST   IBM SYSTEM/36  JAA
JAA.3              9     FB     128   24576      0   COPYFILE   1          84/12/06 PROTECT    1  TEST   IBM SYSTEM/36  JAA
JFILE              5     FB      16   24576      0   COPYFILE              84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
JFILE2            10     F       32      32     98   EXCHANGE              84/12/07 84/12/08   1  TEST   IBM SYSTEM/36
JFILE3            11     FB      32     256    104   EXCHANGE              84/12/07 84/12/13   1  TEST   IBM SYSTEM/36
JFILE4            12     FB      32   24576     98   COPYFILE              84/12/07 84/12/26   1  TEST   IBM SYSTEM/36
JPROCS            13     FB     256   24576     23   SAVELIBR              84/12/10 PROTECT    1  TEST   IBM SYSTEM/36
LETTERA            6     FB      32   16416          ARCHIVE               84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
SAVEMOD           14     FB     256    4096          LIBRFILE              84/12/10 84/12/11   1  TEST   IBM SYSTEM/36
SAVEMOD           15     FB     256    4096          LIBRFILE              84/12/10 84/12/11   1  TEST   IBM SYSTEM/36
TXTJAA             1     FB     256   24576          SAVEFLDR              84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
TXTJO.99           2     FB     256   24576          SAVEFLDR   2          84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
XFILE.A           17     FB     512   24576    164   COPYFILE   1          85/04/04 PROTECT    1  TEST   IBM SYSTEM/36  XFILE
XFILE.1           16     FB     128   24576    237   COPYFILE   1          85/04/04 PROTECT    1  TEST   IBM SYSTEM/36  XFILE
XFILE.5           18     FB     256   24576   1000   COPYFILE   1          85/04/04 PROTECT    1  TEST   IBM SYSTEM/36  XFILE
** END OF TAPE LABEL DISPLAY **
```

Tape Label Listing by Location

```
                                    TAPE  LABEL  DISPLAY                                          PAGE      1
                                          BY  LOCATION

  VOLUME ID - TEST     UNIT - T1    LABEL TYPE - SL     OWNER ID - ACCOLA                  DATE - 85/04/04    TIME - 14.02

STATUS CODES     1 - SAVED USING SAVE ALL      2 - EXTENT      3 - FILE HAS NO HDR2 LABEL

                  FILE   REC  RECORD  BLOCK  ALLOC                          CREATE   EXPIRE   MULTIVOLUME      CREATING      SET
FILE LABEL       SEQNUM  FMT  LENGTH  LENGTH  RECS   FILETYPE   STATUS      DATE     DATE     SEQNUM VOLID     SYSTEM        NAME

TXTJAA             1     FB     256   24576          SAVEFLDR              84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
TXTJO.99           2     FB     256   24576          SAVEFLDR   2          84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
DICTJAA            3     FB     256   24576          SAVEFLDR              84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
DICTO.99           4     FB     256   24576          SAVEFLDR   2          84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
JFILE              5     FB      16   24576      0   COPYFILE              84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
LETTERA            6     FB      32   16416          ARCHIVE               84/10/05 PROTECT    1  TEST   IBM SYSTEM/36
JAA.1              7     FB      32   24576      0   COPYFILE   1          84/12/06 PROTECT    1  TEST   IBM SYSTEM/36  JAA
JAA.2              8     FB      64   24576      0   COPYFILE   1          84/12/06 PROTECT    1  TEST   IBM SYSTEM/36  JAA
JAA.3              9     FB     128   24576      0   COPYFILE   1          84/12/06 PROTECT    1  TEST   IBM SYSTEM/36  JAA
JFILE2            10     F       32      32     98   EXCHANGE              84/12/07 84/12/08   1  TEST   IBM SYSTEM/36
JFILE3            11     FB      32     256    104   EXCHANGE              84/12/07 84/12/13   1  TEST   IBM SYSTEM/36
JFILE4            12     FB      32   24576     98   COPYFILE              84/12/07 84/12/26   1  TEST   IBM SYSTEM/36
JPROCS            13     FB     256   24576     23   SAVELIBR              84/12/10 PROTECT    1  TEST   IBM SYSTEM/36
SAVEMOD           14     FB     256    4096          LIBRFILE              84/12/10 84/12/11   1  TEST   IBM SYSTEM/36
SAVEMOD           15     FB     256    4096          LIBRFILE              84/12/10 84/12/11   1  TEST   IBM SYSTEM/36
XFILE.1           16     FB     128   24576    237   COPYFILE   1          85/04/04 PROTECT    1  TEST   IBM SYSTEM/36  XFILE
XFILE.A           17     FB     512   24576    164   COPYFILE   1          85/04/04 PROTECT    1  TEST   IBM SYSTEM/36  XFILE
XFILE.5           18     FB     256   24576   1000   COPYFILE   1          85/04/04 PROTECT    1  TEST   IBM SYSTEM/36  XFILE
** END OF TAPE LABEL DISPLAY **
```

Description of Items on the Tape Label Listing

VOLUME ID specifies the volume ID of the tape.

UNIT specifies whether tape drive 1 (T1), tape drive 2 (T2), or tape cartridge (TC) contains the tape.

LABEL TYPE specifies the type of labeling on the tape. SL indicates a standard label tape.

OWNER ID specifies the owner ID of the tape.

DATE specifies the system date in the current format (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

TIME specifies the time.

FILE LABEL specifies the name of the file contained on the tape.

FILE SEQNUM specifies where the file is located on tape.

REC FMT specifies the format of the records in the file. This field will not be displayed if the status code is 3.

D indicates variable length, unblocked records in ASCII format.

DB indicates variable length, blocked records in ASCII format.

F indicates fixed length, unblocked records.

FB indicates fixed length, blocked records.

V indicates variable length, unblocked records.

VB indicates variable length, blocked records.

U indicates the record format is undefined.

Note: F, FB, and V are supported on System/36.

RECORD LENGTH specifies (in decimal) the number of bytes per record for a file that has a fixed length record format. For variable length record format, the value shown is the maximum record length in the file. This field will not be specified if the status code is 3.

BLOCK LENGTH specifies (in decimal) the number of bytes in a block of records for a file that has a fixed length blocked record format. This field will not be specified if the status code is 3.

ALLOC RECS specifies the number of records in a file. For a file with unblocked records, the value is the number of records in the file. For a file with blocked records, the value is the maximum number of records that the file can contain. Files with a status code of 3 will indicate the file size in blocks.

FILE TYPE specifies the type of the file, as indicated by the following:

APARFILE indicates that the file was created by the APAR procedure or the $REAIDS utility program.

ARCHIVE indicates a file that contains a folder member that was created by the ARCHIVE procedure or the $TMSERV utility program.

COPYFILE indicates that the file was created by the SAVE procedure or the $COPY utility program.

EXCHANGE indicates a file that is not one of the other types listed. It may have been created by the TAPECOPY procedure or the $TCOPY utility program.

LIBRFILE indicates a file, containing one or more library members, that was created by the FROMLIBR procedure or the $MAINT utility program.

SAVEFLDR indicates a file that contains a folder that was created by the SAVEFLDR procedure or the $TMSERV utility program.

SAVELIBR indicates a file that contains a library that was created by the SAVELIBR procedure or the $MAINT utility program.

STATUS specifies the status of the file.

1 indicates that the file was saved using the SAVE ALL procedure.

2 indicates a folder extent.

3 indicates that the file has no HDR2 label.

CREATE DATE specifies the session date that the file was saved in the current format. If a file was saved with a SAVE ALL (file status 1), the date shown is the creation date of the file on disk.

EXPIRE DATE specifies when the file on tape is to expire (that is, when the file is no longer needed and can be removed). PROTECT indicates that the file is permanent, and can only be removed by initializing the tape.

MULTIVOLUME specifies whether the file is continued on another tape.

SEQNUM specifies the volume number of a tape within the multivolume file. SEQNUM is 1 if the first file on the tape is not continued from another tape. It is the same for every file on the tape and is incremented by 1 whenever a file is continued to another tape. A letter may follow the sequence number.

C indicates that the file is continued on one or more tapes.

L indicates that the tape is the last tape containing a continued file.

VOLID specifies the volume ID of the multivolume file. VOLID is the volume ID of the first tape in a multivolume group.

CREATING SYSTEM specifies the IBM system that created the file.

SET NAME specifies the name given to a set of files saved using SAVE ALL

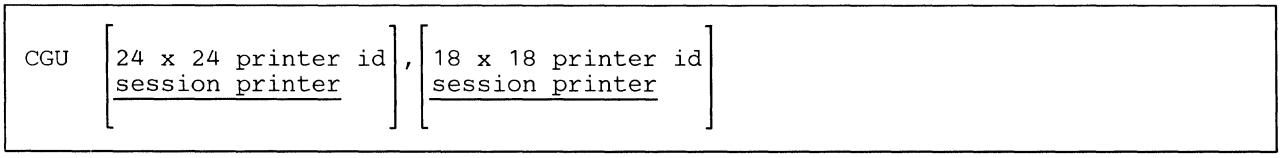
Note: An asterisk (*) in any field of a file label display indicates a nondisplay character was found in the tape label.

CGU Procedure

The CGU procedure allows you to start a character generator utility (CGU) session as well as to define the printer.

For more information on the CGU procedure, refer to the *Character Generator Utility Guide.*

Note: The CGU procedure supports only the Japanese extended character files. Extended character files from Korea, Taiwan, and People's Republic of China are not supported.



S9020071-0

24x24 printer id specifies the work station ID of the printer for which 24x24 output is to be printed. If this parameter is not specified, the session printer is assumed.

18x18 printer id specifies the work station ID of the printer for which 18x18 output is to be printed. If this parameter is not specified, the session printer is assumed.

Example

This example starts a CGU session and sets printer P2 as the printer to print the 24x24 ideographic output.

```
CGU P2
```

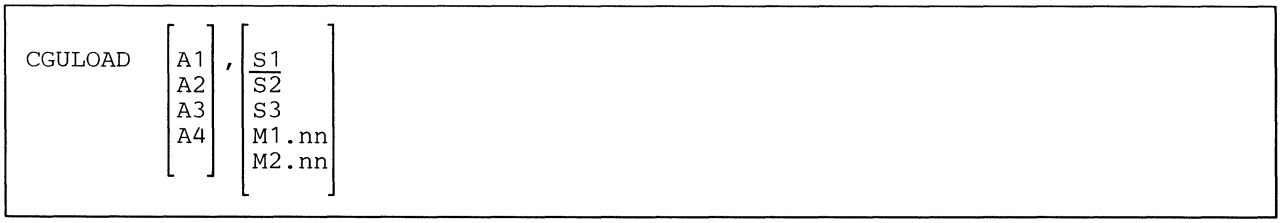
CGULOAD Procedure

The CGULOAD procedure creates a library named #CGULIB and copies the character generator utility (CGU) support from diskette into that library. CGULOAD copies additional support into the system library (#LIBRARY). The CGULOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the CGUSAVE procedure. See the "CGUSAVE Procedure" on page 4-93 for information about how to save the CGU support on diskette.

The CGULOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the CGULOAD procedure to restore support that has been saved by CGUSAVE.

If CGU is not currently on the system, you must use the TOLIBR procedure to copy the diskette file CGU into #LIBRARY before running CGULOAD.

If #LIBRARY was backed up with CGU on the system and then replaced before CGULOAD is run, you do not have to copy the diskette file CGU using the TOLIBR procedure.



S9020072-0

A1, A2, A3 or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If no location is specified and the system has more than one disk unit, the file is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #CGULIB on disk and copy the CGU support from diskette.

```
CGULOAD
```

CGUSAVE Procedure

The CGUSAVE procedure copies the character generator utility (CGU) support to diskette. The CGU support from the libraries #CGULIB and #LIBRARY is copied. You should use the "CGULOAD Procedure" on page 4-92 to load the CGU support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPSSP and be located in diskette slot S1.

```
CGUSAVE
```

S9020073-0

The CGUSAVE procedure has no parameters.

Example

Copy the CGU support to diskette.

```
CGUSAVE
```

CHGXLATE Procedure

The CHGXLATE procedure allows you to modify the tables that are used to translate characters from ASCII to EBCDIC and EBCDIC to ASCII.

For more information about how to run the CHGXLATE procedure, see the *PC Support/36 Technical Reference*.

To modify a translation table:

```
CHGXLATE
```

S9020549-0

The CHGXLATE procedure has no parameters.

Example

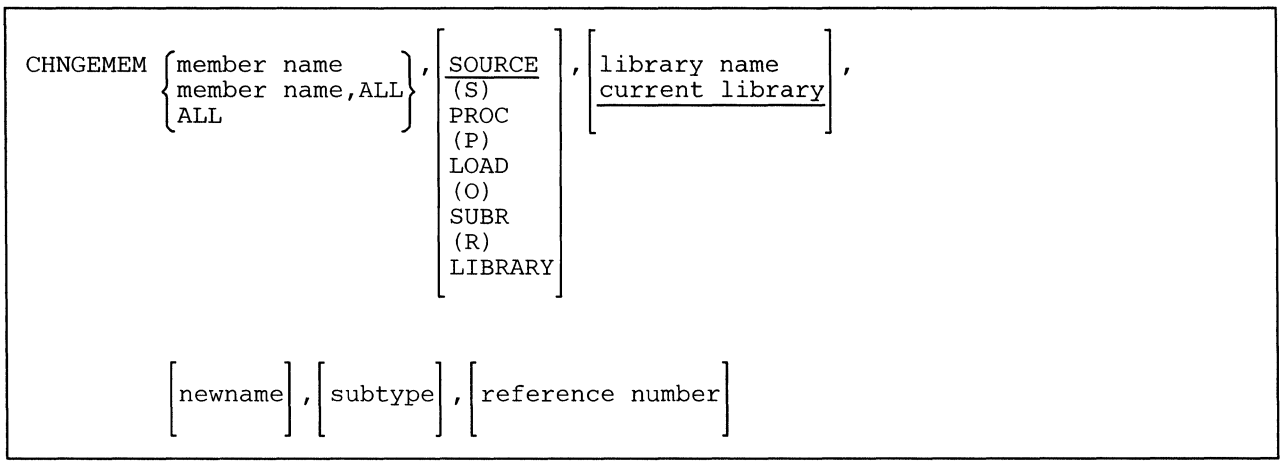
To modify a translation table using the CHGXLATE procedure, you would enter:

```
CHGXLATE
```

CHNGEMEM Procedure

The CHNGEMEM procedure changes the name, subtype, or reference number of a library member. The CHNGEMEM procedure cannot change IBM-supplied library members.

The CHNGEMEM procedure runs the $MAINT utility program.



S9020074-0

member name specifies the library member to be changed.

member name,ALL specifies that the library members whose names begin with the specified characters (member name) are to be changed. Up to 7 characters can be specified for member name.

ALL specifies that all members of the library are to be changed. An error message is given if this parameter is specified with the newname parameter.

SOURCE or S specifies that only the library source members are to be changed. If no parameter is specified, SOURCE is assumed.

PROC or P specifies that only the library procedure members are to be changed.

LOAD or O specifies that only the library load members are to be changed.

SUBR or R specifies that only the library subroutine members are to be changed.

LIBRARY specifies that all library member types (SOURCE, PROC, LOAD, and SUBR) for the specified library name are to be changed.

library name specifies the library containing the members to be changed. If no library name is specified, the current library is assumed.

newname specifies the new member name of the changed members. If *member name.ALL* is specified, the new member name must contain the same number of characters as specified in the *member name.ALL* parameter. The remaining characters in each member name are not changed.

subtype specifies the new subtype for the changed members. Valid subtypes are:

ARP RPG auto report member
ARS Automatic response member
ASM Assembler member
BAP BASIC procedure (source member)
BAS BASIC member
BGC Business graphics chart
BGD Business graphics data
BGF Business graphics format
COB COBOL member
DFU Data file utility member
DTA Data member
FMT Display format member
FOR FORTRAN member
ICF CNFIGICF procedure Interactive Communications feature member
MNU Menu member
MSG Message member
PHL Phone list member
RPG RPG member
SRT Sort member
SSP CNFIGSSP procedure system configuration member
UNS Unspecified
WSU Work station utility member

reference number specifies the user-defined reference number to which the members are to be changed. The value specified must be numeric and within the range of 0 to 999999.

Example 1

This example changes the name of all the members that begin with PAY to DAY. The members are in a library called PAYROLL.

```
CHNGEMEM PAY,ALL,LIBRARY,PAYROLL,DAY
```

Example 2

This example changes the subtype and reference number of a source member named PAYR01 in a library named PAYROLL. The new subtype is SRT with a new reference number of 123.

```
CHNGEMEM PAY01,SOURCE,PAYROLL,,SRT,123
```

CNFIGICF Procedure

The CNFIGICF procedure configures or sets up a communications subsystem that consists of a line member and a subsystem member. Communications subsystems are required for the Interactive Communications features (SSP-ICF), MSRJE, 3270 device emulation, C & SM, and document distribution using Personal Services/36.

For information on defining a line member, see the manual *Using System/36 Communications.*

For information on a specific communications subsystem, see the appropriate manual for that subsystem.

```
CNFIGICF
```

S9020075-0

The CNFIGICF procedure has no parameters.

Example

To run the CNFIGICF procedure:

```
CNFIGICF
```

CNFIGSSP Procedure

The CNFIGSSP procedure configures or sets up the System/36. For example, you use it to do the following:

- Define local and remote display stations and printers.
- Add or delete program products, optional SSP, and SSP features.
- Define your base SSP values; for example, whether print spooling or the job queue are to be active.
- Specify the sizes for the history file, the disk VTOC, and the task work area.

For information about displays shown by CNFIGSSP, see the manual *Changing Your System Configuration.*

```
CNFIGSSP
```

S9020076-0

The CNFIGSSP procedure has no parameters.

Example

To run the CNFIGSSP procedure:

```
CNFIGSSP
```

CNFIGX25 Procedure

The CNFIGX25 procedure allows you to define the network configuration, logical channel configuration, and virtual circuit configuration. The CNFIGX25 procedure is described in the manual *Using System/36 Communications.*

```
CNFIGX25
```

S9020077-0

The CNFIGX25 procedure has no parameters.

Example

To run the CNFIGX25 procedure:

```
CNFIGX25
```

COBLOAD Procedure

The COBLOAD procedure creates a library named #COBLIB and copies the COBOL support from diskette into that library. COBLOAD copies additional support into the system library (#LIBRARY). The COBLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the COBSAVE procedure. See the "COBSAVE Procedure" on page 4-103 for information about how to save the COBOL support on diskette.

The COBLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the COBLOAD procedure to restore support that has been saved by COBSAVE.

If COBOL is not currently on the system, you must use the TOLIBR procedure to copy the diskette file COBOL into #LIBRARY before running COBLOAD.

If #LIBRARY was backed up with COBOL on the system and then replaced before COBLOAD is run, you do not have to copy the diskette file COBOL using the TOLIBR procedure.

```
COBLOAD  [A1]  ,[S1]
         [A2]   [S2]
         [A3]   [S3]
         [A4]   [M1.nn]
                [M2.nn]
```

S9020078-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on the unit, the other units (if they exist) are checked, and #COBLIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #COBLIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #COBLIB on disk and copy the COBOL support from diskette.

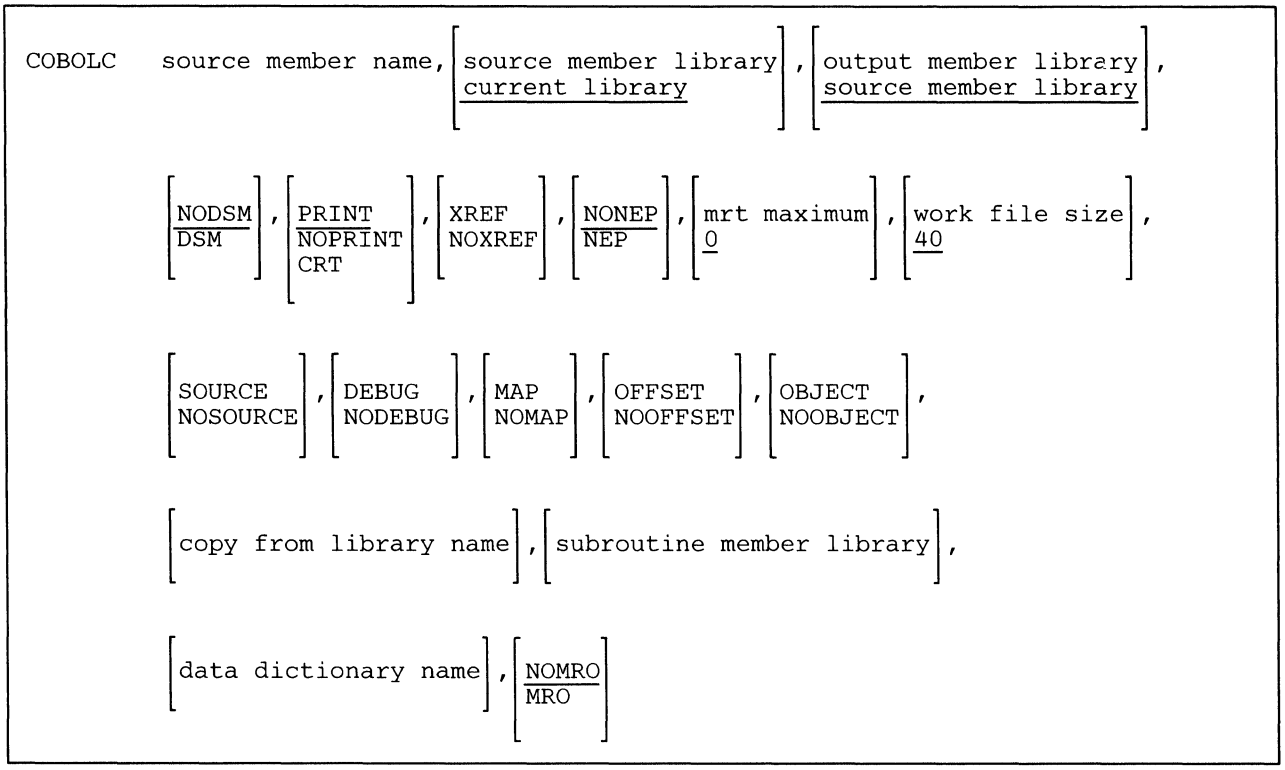
```
COBLOAD
```

COBOL Procedure

This procedure is only supported for compatibility with the IBM System/34. See the "COBOLC Procedure" on page 4-99 for information about compiling COBOL programs.

COBOLC Procedure

The COBOLC procedure compiles and link-edits a COBOL source program. For more information on COBOL, see the manual *Programming with COBOL*.



S9020079-1

source member name specifies the source member to be compiled.

source member library specifies the name of the library that contains the source member. If this parameter is not specified, the current library is assumed.

output member library specifies the name of the library that will contain the load member or subroutine member produced by the compile. If the library is not specified on the PROCESS statement or by using this parameter, the source member library is assumed.

NODSM specifies that no diagnosed source member is to be created. If no parameter is specified, NODSM is assumed.

DSM specifies that a diagnosed source member is to be created.

PRINT specifies that the compiler output is to be printed. If no parameter is specified, PRINT is assumed.

NOPRINT specifies that no compiler output is to be displayed or printed.

CRT specifies that the compiler output is to be displayed.

XREF or NOXREF specifies whether a cross-reference listing of the COBOL program is to be produced. If no parameter is specified, the option specified in the COBOL program's PROCESS statement is assumed.

XREF specifies that a cross-reference listing is to be produced.

NOXREF specifies that no cross-reference listing is to be produced.

NONEP specifies that the program is not a never-ending program. If no parameter is specified, NONEP is assumed.

NEP specifies that the program is a never-ending program. For more information about never-ending programs, see the *Concepts and Programmer's Guide*.

mrt maximum identifies the program being compiled as a multiple requester terminal (MRT) program and specifies the maximum number of requester display stations that can be using the program at one time. The maximum value is 99. If the value is 0 or is not specified, the program is not an MRT program.

work file size specifies the number of blocks to use for the compiler's work files. The maximum work file size is 9999. If this parameter is not specified, 40 blocks is assumed.

SOURCE or NOSOURCE specifies the print option to be used instead of the print option entered in the COBOL program's PROCESS statement. If no parameter is specified, the option specified in the PROCESS statement is used.

SOURCE specifies that the compiler is to produce a listing of the source statements in the COBOL program along with any error messages.

NOSOURCE specifies that the compiler is not to produce a listing of the source statements in the COBOL program; only the error messages are to be listed.

DEBUG or NODEBUG specifies the debug option to be used instead of the debug option entered in the COBOL program's PROCESS statement. If no parameter is specified, the option specified in the PROCESS statement is used.

DEBUG specifies that the compiler is to use the COBOL debug facility.

NODEBUG specifies that the compiler is not to use the COBOL debug facility.

MAP or NOMAP specifies whether the COBOL program's Data Division mapping option to be used instead of the option entered in the COBOL program's PROCESS statement. If no parameter is specified, the option specified in the PROCESS statement is used.

MAP specifies that the compiler is to map the program's Data Division.

NOMAP specifies that the compiler is not to map the program's Data Division.

OFFSET or NOOFFSET specifies whether the COBOL program's Procedure Division mapping option to be used instead of the option entered in the COBOL program's PROCESS statement. If no parameter is specified, the option specified in the PROCESS statement is used.

OFFSET specifies that the compiler is to map the program's Procedure Division.

NOOFFSET specifies that the compiler is not to map the program's Procedure Division.

OBJECT or NOOBJECT specifies whether or not the compiler is to produce an object module that must be link-edited. If no parameter is specified, the option specified in the COBOL program's PROCESS statement is used.

OBJECT specifies that the compiler is to produce an object module.

NOOBJECT specifies that the compiler is not to produce an object module.

copy from library name specifies the library to be searched when a COBOL COPY statement is encountered. See the manual *Programming with COBOL* for more information. If this parameter is not specified, the library specified in the COBOL program's PROCESS statement is assumed.

subroutine member library specifies the library that contains one or more subroutines that are to be included with the compiled program. If this parameter is not specified, the library specified in the COBOL program's PROCESS statement is assumed.

data dictionary name specifies the data dictionary that contains the communications file definition to be used with the program being compiled.

MRO or NOMRO specifies whether the compiler is to use memory resident overlays. This parameter can be used to override the option specified in the COBOL program's PROCESS statement. If no parameter is specified, NOMRO is assumed.

MRO specifies that the compiler is to use memory resident overlays.

NOMRO specifies that the compiler is not to use memory resident overlays.

Example

This example shows how to compile a COBOL program. The source member is named PAYROLL and it is in a library named TESTLIB. The compiled program is to be named PAYROLL and is to be placed in a library named PAYLIB. A diagnosed source member is to be created if errors are found during the compilation.

```
COBOLC PAYROLL,TESTLIB,PAYLIB,DSM,,XREF,,,,SOURCE,DEBUG
```

COBOLCG Procedure

This procedure is only supported for compatibility with the IBM System/34. See the "COBOLC Procedure" on page 4-99 for information about compiling COBOL programs. See the "LOAD OCL Statement" on page 5-69 and the "RUN OCL Statement" on page 5-105 for information about running programs.

COBOLG Procedure

This procedure is only supported for compatibility with the IBM System/34. See the "LOAD OCL Statement" on page 5-69 and the "RUN OCL Statement" on page 5-105 for information about running programs.

COBOLONL Procedure

The COBOLONL procedure allows you to develop a COBOL program. You can create a new program, or make a large number of changes to an existing program. The COBOLONL procedure shows a series of displays that allow you to enter, compile, and change COBOL programs.

For more information about COBOL, see the manual *Programming with COBOL*.

```
COBOLONL
```

S9020080-0

The COBOLONL procedure has no parameters.

Example

To develop a COBOL program, enter:

```
COBOLONL
```

COBOLP Procedure

The COBOLP procedure causes a menu to be displayed that allows you to select the option you want to perform. You can enter, compile, or change COBOL programs.

For more information about COBOL, see the manual *Programming with COBOL*.

```
COBOLP
```

S9020081-0

The COBOLP procedure has no parameters.

Example

This example shows how to display the COBOLP menu.

```
COBOLP
```

COBSAVE Procedure

The COBSAVE procedure copies the COBOL support to diskette. The COBOL support from the libraries #COBLIB and #LIBRARY is copied. You should use the "COBLOAD Procedure" on page 4-98 to load the COBOL support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPCOBL and be located in diskette slot S1.

```
COBSAVE
```

S9020082-0

The COBSAVE procedure has no parameters.

Example

Copy the COBOL support to diskette.

```
COBSAVE
```

COBSDA Procedure

The COBSDA procedure starts the screen design aid (SDA) procedure. See the manual *Creating Displays* for information about display formats and about how to use SDA.

See the manual *Programming with COBOL* for more information about this procedure and about COBOL.

```
COBSDA
```

S9020083-0

The COBSDA procedure has no parameters.

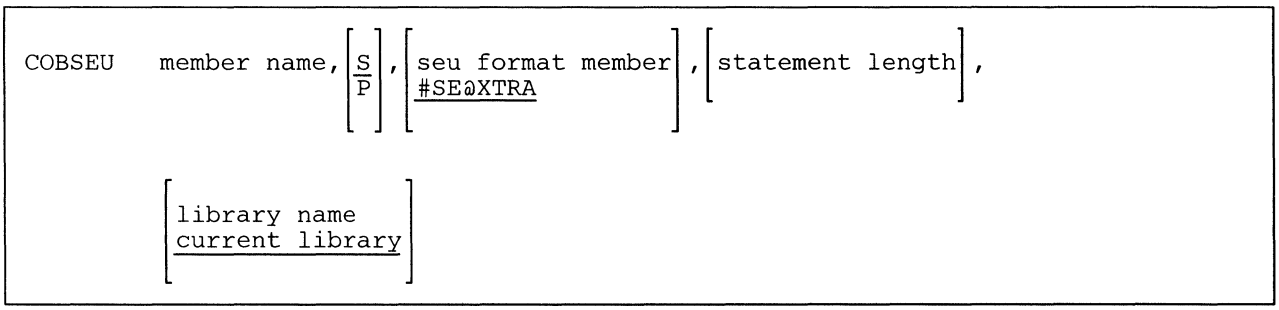
Example

To start the COBSDA procedure, enter:

```
COBSDA
```

COBSEU Procedure

The COBSEU procedure starts the source entry utility (SEU) procedure. This allows you to create or change a COBOL program or procedure. See the manual *Programming with COBOL* for more information about this procedure and about COBOL. See the *SEU Guide* for more information on SEU.



S9020084-0

member name specifies the source member or procedure that you want to create or change.

S specifies a COBOL source member. If no parameter is specified, S is assumed.

P specifies a library procedure member.

seu format member specifies the name of the SEU display format member to be used to edit source statements. If no format name is specified, #SE@XTRA is assumed.

statement length specifies the length for each source or procedure statement. This can be any decimal number from 40 to 120. If the member exists, the statement length of the member is assumed. If the member is being created, the values you can specify and the values that are assumed by COBSEU if no statement length is specified are as follows:

Member Type	Allowed Statement Length	Assumed Statement Length
S	80 to 96	96
P	40 to 120	120

library name specifies the library that contains or will contain the library member being changed or created. If no library name is specified, the current library is assumed.

Example

This example shows how to start COBSEU to create or change a procedure named PAYROLL. The procedure is contained in the current library.

```
COBSEU PAYROLL,P
```

COMPRESS Procedure

The COMPRESS procedure collects all unused disk space within the user area. This allows you to make room available on the disk for larger files, libraries, and folders by gathering the smaller unused areas together. This area can be located at either the high or low block numbers for each disk unit.

For systems with one disk drive, the A1 refers to that drive. For systems with multiple disk drives, A1 refers to the first drive; A2 refers to the second disk drive; A3 refers to the third disk drive; and A4 refers to the fourth disk drive.

The COMPRESS procedure can be run from any command-capable display station, from the job queue, from the Interactive Communications feature (SSP-ICF), or evoked using the EVOKE OCL statement.

When COMPRESS is running, the following tasks are allowed:

- Operators signed on to the system (but not running programs)
- Remote work stations
- SDLC
- SSP-ICF subsystems
- Autocall

Although the COMPRESS procedure does not require a dedicated system, COMPRESS does prevent you from doing the following:

- Entering data or commands
- Signing on to display stations
- Switching into console mode at the system console or at a subconsole

See the "#STRTUP1 Procedure" on page 4-4 for information about compressing the disk as part of initial program load (IPL).

COMPRESS

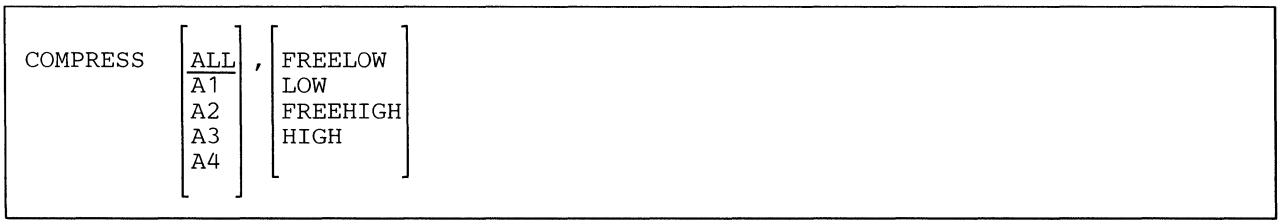
The COMPRESS procedure runs the $FREE utility program. The amount of time the system takes to run the COMPRESS procedure is affected by the size of main storage available for running programs. The COMPRESS procedure automatically uses as much of this space as possible.

Note:

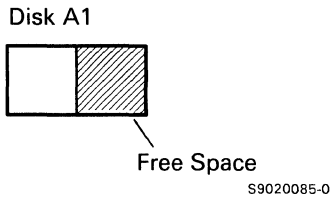
1. *If the LOCATION parameter was specified in the FILE statement for a file moved by the COMPRESS procedure, the location specified is no longer correct after the COMPRESS procedure moves the file. Use the CATALOG procedure to display the VTOC entries to determine the new locations of the files.*

2. *There can only be a maximum of 100 disk scratch files referenced in the procedure that contains the COMPRESS procedure.*

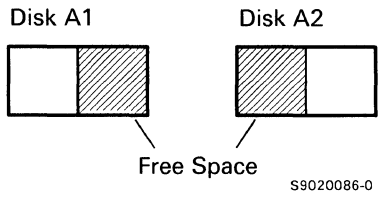
3. *If COMPRESS is run on a 5362 and there is a disk drive which is powered off, this may cause fragmentation of the disk. Library extents are not moved if the library is on a disk drive which is not available.*



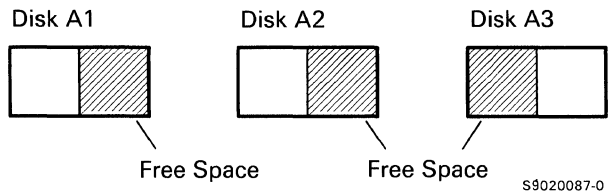
ALL specifies that all disk drives are to be compressed. If you have only one disk drive (A1), free space is to be collected at the highest block numbers (FREEHIGH).



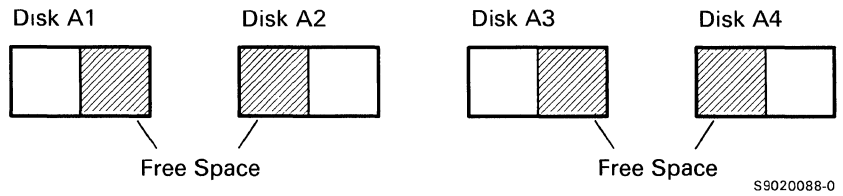
If you have two disk drives (A1, A2), free space is to be collected at the highest block numbers (FREEHIGH) for A1, and at the lowest block numbers (FREELOW) for A2.



If you have three disk drives (A1, A2, A3), free space is to be collected at the highest block numbers (FREEHIGH) for A1 and A2, and at the lowest block numbers (FREELOW) for A3.



If you have four disk drives (A1, A2, A3, A4), free space is to be collected at the highest block numbers (FREEHIGH) for A1 and A3, and at the lowest block numbers (FREELOW) for A2 and A4.



If no parameters are specified (COMPRESS is specified only), COMPRESS ALL is assumed.

A1 specifies that only the first disk drive is to be compressed. If no second parameter is specified, FREEHIGH is assumed.

A2 specifies that only the second disk drive is to be compressed. If no second parameter is specified, FREELOW is assumed. A2 is not valid unless the system has at least two disk drives.

A3 specifies that only the third disk drive is to be compressed. If no second parameter is specified, FREELOW is assumed. A3 is not valid unless the system has at least three disk drives.

A4 specifies that only the fourth disk drive is to be compressed. If no second parameter is specified, FREELOW is assumed. A4 is not valid unless the system has four disk drives.

FREELOW or LOW specifies that the free space on the specified disk drive is to be collected at the lowest block numbers (the beginning) of the drive. You could use this when you need to increase the size of the system library (#LIBRARY) and you need to move one or more files or libraries away from #LIBRARY.

FREEHIGH or HIGH specifies that the free space on the specified disk drive is to be collected at the highest block numbers (the end) of the drive.

Example 1

To compress the free space on disk:

```
COMPRESS
```

Example 2

To compress all disk units and cause the free space to be collected at the high block numbers of each unit, use the following:

```
COMPRESS ALL,FREEHIGH
```

Example 3

To compress disk unit A1 and cause the free space to be collected at the high block numbers:

```
COMPRESS A1,FREEHIGH
```

CONDENSE Procedure

The CONDENSE procedure collects all unused space within a library or folder into a single area. This allows you to make room available for more members by gathering the smaller unused areas together. The unused space is collected at the end of the library or folder. Also, if the library has an extent, the extent is combined with the library (if the library has enough space). A library extent is an area on disk that contains library members that would not fit into the original library.

When the CONDENSE procedure is running, no other jobs or display stations can use the specified library or folder. If an attempt is made to run the CONDENSE procedure for a library or folder that is being used by another job, an error message is displayed and the CONDENSE procedure is not run. Once CONDENSE begins running, no other jobs are allowed to use the specified library or folder until the CONDENSE procedure is complete.

The system library (#LIBRARY) cannot be condensed while:

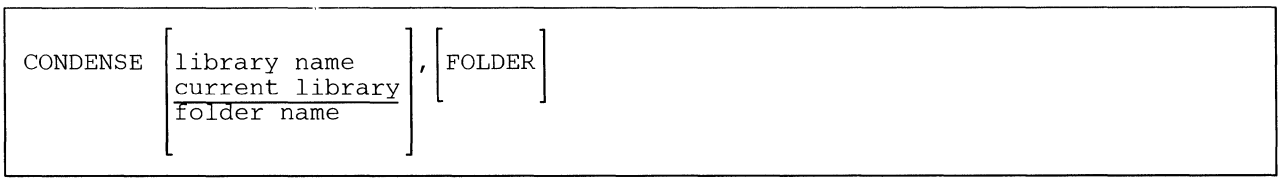
- Any other jobs are being run
- SSP-ICF subsystems are enabled
- A communications line is enabled

You can use the STATUS USERS and the STATUS SYSTASK control commands to determine whether any of these conditions are true.

See the "#STRTUP1 Procedure" on page 4-4 for information about condensing the system library as part of initial program load (IPL).

The CONDENSE procedure can also reorganize a folder. The folder members are moved together at the front of the folder to make the number of extents that the folder is made up of as small as possible. (This may increase performance because the folder makes better use of disk space.) The CONDENSE procedure also makes the folder itself as small as possible. This is the same as running the ALOCFLDR procedure with the MIN parameter specified.

The CONDENSE procedure runs the $MAINT utility program. If the FOLDER parameter is specified, the CONDENSE procedure runs the $TMSERV utility program.



S9020090-0

library name specifies the library to be condensed. If a library name is not specified, the current library is assumed.

folder name specifies the folder to be reorganized. If the FOLDER parameter is specified, this parameter is required.

FOLDER specifies that the folder specified by the first parameter is to be reorganized.

Example 1

To condense the members in the system library:

```
CONDENSE #LIBRARY
```

Example 2

To condense the members in the current library, which is named MYLIB:

```
CONDENSE
```

Example 3

To reorganize a folder name MYFLDR:

```
CONDENSE MYFLDR,FOLDER
```

COPYDATA Procedure

The COPYDATA procedure copies a data file on disk to another data file on disk. During the copy operation, the COPYDATA procedure can:

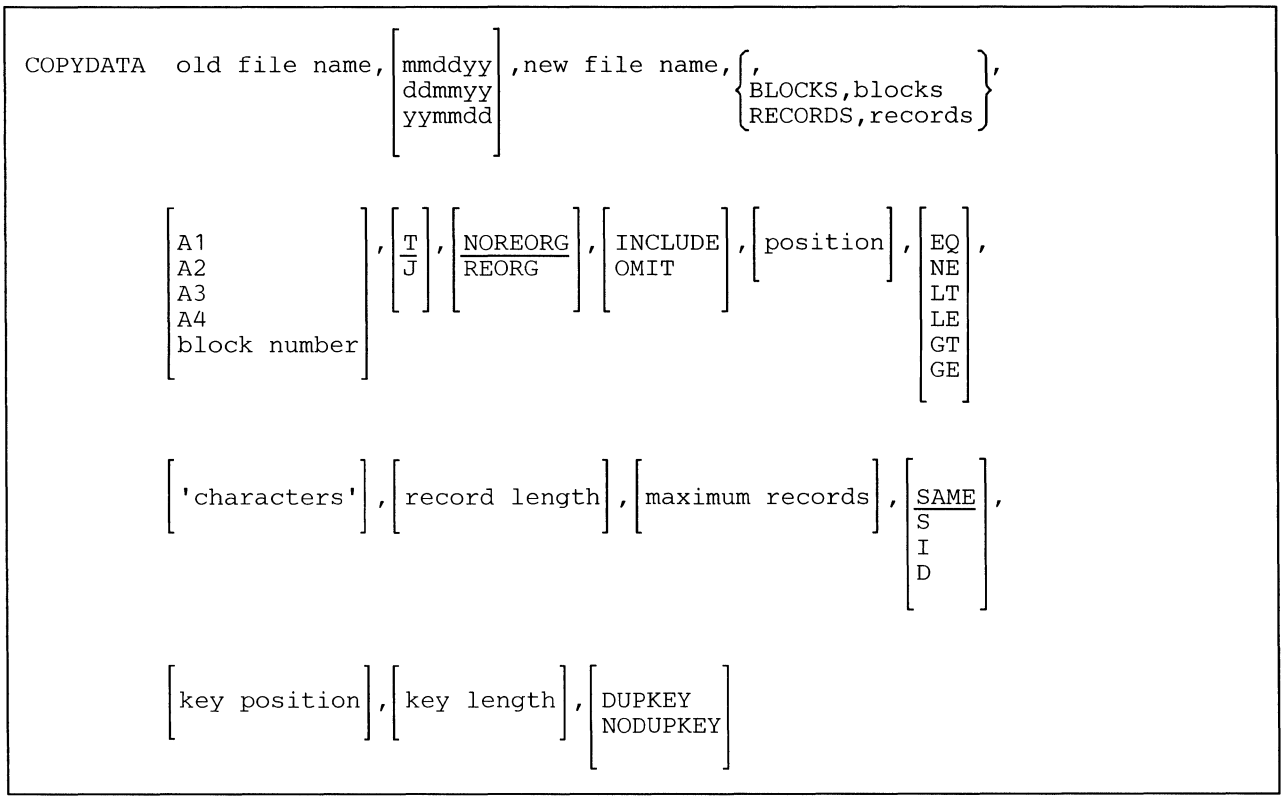
- Create the copied file in either the same file organization, or a different file organization. Files can be sequential, direct, or indexed.
- Omit or include specific records.
- Remove deleted records.
- Change the record length.
- Reorganize data records in an indexed file.
- Limit the number of output records to be copied.

The COPYDATA procedure can be used to copy files to or from a remote system. See the *Distributed Data Management Guide*.

The COPYDATA procedure can process a file that is being used by another job on the system, providing that job's FILE OCL statement specifies DISP-SHRRR or DISP-SHRRM; see the "FILE OCL Statement (for Disk Files)" on page 5-32 for more information about file sharing. The COPYDATA procedure cannot copy alternative index files.

The COPYDATA procedure runs the $COPY utility program.

COPYDATA



S9020091-1

old file name specifies the file to be copied. The file can have any file organization. However, alternative index files cannot be copied. See the "BLDINDEX Procedure" on page 4-59 for more information about alternative indexes.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be copied. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If a date is not specified and more than one file exists with the same file name, the most recent file created is copied.

new file name specifies the name of the new file containing copied records from the old file. The first character of the file name must be alphabetic (A through Z, #, @, or $). The file name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. The number of characters in a file name must not exceed eight.

BLOCKS or RECORDS specifies the size for the new file. If neither BLOCKS or RECORDS are specified, the size of the new file is the same as the size of the old file. If the file's organization and/or record length is being changed, the new file is made large enough to hold the total number of records in the old file.

BLOCKS specifies the number of disk file blocks to be allocated (that is, reserved) for the new file. One disk block contains 2560 bytes.

blocks specifies the number of blocks needed for the new file. It cannot be greater than the configured maximum.

You can use the CATALOG procedure to determine the number of disk blocks available for a file. A file cannot be created if it contains more than 16777200 records.

RECORDS specifies the number of records to be allocated (that is, reserved) for the new file. If you specify RECORDS, the SSP allocates the required number of blocks to contain those records. The total space reserved for the new file is rounded up to the next block, allowing the file to contain at least the number of records specified. The smallest disk file unit that can be reserved is one block.

records specifies the number of records to allocate for the new file. It cannot be greater than the value 8000000.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked and the file is placed on the least used disk unit. If no location is specified, the preferred location of the old file is assumed.

block number specifies the block number of the first block in the new file. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk. Use the block number parameter with caution because the COMPRESS and RESTORE procedures can cause a file to be moved from the block number specified when the file was created.

T specifies that the new file is to be a resident file. A resident file is usually used more than once. The disk area containing a resident file becomes available for another file after the DELETE procedure is used to remove the file from the disk. For more information about resident files, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

If no parameter is specified, T is assumed.

J specifies that the new file is to be a job file. A job file can only be used by following job steps in a job. The job file is defined only within the job and is deleted by the SSP when the job ends.

Once a job file is created, following steps in the job can refer to the file by using a FILE OCL statement that contains either the RETAIN-S or RETAIN-J parameter, or by not specifying a RETAIN parameter. If the RETAIN parameter is not specified, or if RETAIN-J is specified, the file still exists as a job file for following job steps. If RETAIN-S is specified for a job file, the job file becomes a scratch file, and is removed from the disk when that job step ends. For more information about job files, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

NOREORG specifies that any deleted records are not to be removed in the copy process. Also, the records in an indexed file are to be in the same position in the new file as they were in the file to be copied. Deleted records will, however, be removed when any additional parameters are specified for sequential or indexed files. If no parameter is specified, NOREORG is assumed.

REORG specifies that any deleted records in the file to be copied are to be removed when the file is copied to the new file. However, deleted records are always copied for direct files. If the file to be copied is an indexed file, the records in the old file are read sequentially by key.

INCLUDE or OMIT specifies whether or not specific records in the file to be copied are to be included in or omitted from the new file. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If only 'characters' or positions are specified, INCLUDE and EQ are assumed.

position specifies, for each record, the first character to be compared with the comparison characters. The position can be any number from 1 through 4096. If a position is not specified, then every position in the record is compared with the comparison characters until the condition is met.

EQ specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is to be included in or omitted from the new file.

NE specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record is to be included in or omitted from the new file.

LT specifies that if the characters in the record indicated by position are less than the comparison characters, the record is to be included in or omitted from the new file.

LE specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record is to be included in or omitted from the new file.

GT specifies that if the characters in the record indicated by position are greater than the comparison characters, the record is to be included in or omitted from the new file.

GE specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record is to be included in or omitted from the new file.

'characters' specifies the comparison characters. Up to 30 characters can be specified, and they should be enclosed by apostrophes ('). Blanks and commas (,) may be included but apostrophes cannot be specified as data.

record length specifies the record length of the new file, and can be any number from 1 through 4096. If this parameter is not entered, the record length of the file to be copied is used for the record length of the new file.

If the record length of the old file is less than the specified record length of the new file, the additional record positions in the new file are filled with blanks. If the record length of the old file is greater than the specified record length of the new file, the extra positions are truncated. If the new file is an indexed file and the key field would be truncated, an error message is displayed.

maximum records specifies the total number of records to be copied into the new file. Any number greater than 0 can be entered. Note that the SSP cannot create a file that contains more than 16777200 records.

SAME specifies that the new file is to have the same organization as the file to be copied. If a parameter is not specified, SAME is assumed.

S specifies that the new file is to be organized as a sequential file.

I specifies that the new file is to be organized as an indexed file.

D specifies that the new file is to be organized as a direct file.

key position specifies the starting position of the key for the new file. The key position must be specified if the new file is to be an indexed file, but the old file was not organized as an indexed file. The key position can be any number from 1 through 4096. If a value is not specified, the key position for the indexed file being copied is assumed. The entire key, defined by the key position and key length, must be within the record. If either the key position or key length is entered, the other must also be entered.

key length specifies the length of the key for the new file. The key length must be specified if the new file is to be an indexed file, but the old file was not organized as an indexed file. The key length can be any number from 1 through 120. If a value is not specified, the key length for the indexed file being copied is assumed. The entire key, defined by the key position and key length, must be within the record. If either the key position or key length is entered, the other must also be entered.

DUPKEY specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file will be the attribute of the output file.

NODUPKEY specifies that duplicate keys are not to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file will be the attribute of the output file. If an indexed file is created with records containing duplicate keys but the records are not supposed to contain duplicate keys, error message SYS-1365 is issued when the keys are sorted at the end of the job, and the duplicate keys are displayed. At that point, the user can choose a response that deletes the file or one that changes the file atttribute so that duplicate keys are allowed for the file.

Note: NODUPKEY will not remove duplicate key records.

Examples

The examples that follow are based on a file, named FILE1, that contains five records. The file is an indexed file, the file's record length is 60, and the index key is in positions 1 through 4.

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE1       10/24/82      INDEXED         4        1         60             F1   10/24/82   10.40.33

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - 0001
LINE NO -   1    0001SAMPLE DATA         4512 - 18TH AVE NW  NEW YORK, NY

 KEY - 0002
LINE NO -   1    0002MORE SAMPLE DATA    100 - MAIN ST N     CHICAGO, IL

 KEY - 0003
LINE NO -   1    0003CUSTOMER 1          3214 - 5TH ST NE    NEW YORK, NY

 KEY - 0004
LINE NO -   1    0004CUSTOMER 2          2014 - 4TH ST NE    NEW YORK, NY

 KEY - 0005
LINE NO -   1    0005CUSTOMER 3          3018 - 5TH ST NE    CHICAGO, IL

    TOTAL RECORDS PRINTED - 5
```

Example 1

This example shows how to copy FILE1 and create an exact copy of it, FILE2.

```
COPYDATA FILE1,,FILE2
```

FILE2 now contains the following records:

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE2       10/24/82      INDEXED         4        1         60             F1   10/24/82   10.40.53

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - 0001
LINE NO -   1    0001SAMPLE DATA         4512 - 18TH AVE NW  NEW YORK, NY

 KEY - 0002
LINE NO -   1    0002MORE SAMPLE DATA    100 - MAIN ST N     CHICAGO, IL

 KEY - 0003
LINE NO -   1    0003CUSTOMER 1          3214 - 5TH ST NE    NEW YORK, NY

 KEY - 0004
LINE NO -   1    0004CUSTOMER 2          2014 - 4TH ST NE    NEW YORK, NY

 KEY - 0005
LINE NO -   1    0005CUSTOMER 3          3018 - 5TH ST NE    CHICAGO, IL

    TOTAL RECORDS PRINTED - 5
```

Example 2

This example shows how to copy a file named FILE1, which is an indexed file, and create a new file named FILE3. The new file is to be an indexed file and the key is to be in positions 5 through 24.

```
COPYDATA FILE1,,FILE3,,,,,,,,,,,I,5,20
```

FILE3 now contains the following records:

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE3       10/26/82      INDEXED         20       5         60             F1   10/26/82   15.02.46

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - CUSTOMER 1
LINE NO -   1    0003CUSTOMER 1          3214 - 5TH ST NE    NEW YORK, NY

 KEY - CUSTOMER 2
LINE NO -   1    0004CUSTOMER 2          2014 - 4TH ST NE    NEW YORK, NY

 KEY - CUSTOMER 3
LINE NO -   1    0005CUSTOMER 3          3018 - 5TH ST NE    CHICAGO, IL

 KEY - MORE SAMPLE DATA
LINE NO -   1    0002MORE SAMPLE DATA    100 - MAIN ST N     CHICAGO, IL

 KEY - SAMPLE DATA
LINE NO -   1    0001SAMPLE DATA         4512 - 18TH AVE NW  NEW YORK, NY

    TOTAL RECORDS PRINTED - 5
```

Example 3

This example shows how to remove deleted records from a delete-capable sequential file named FILEA. The steps taken are as follows:

1. File FILEA is copied into a temporary file named TEMP and is reorganized using the COPYDATA procedure.

2. FILEA is removed from the disk using the DELETE procedure.

3. The temporary file is then renamed as FILEA using the RENAME procedure.

The following procedures are used:

```
COPYDATA FILEA,,TEMP,,,,,REORG
DELETE FILEA,F1
RENAME TEMP,FILEA
```

Example 4

This example shows how to copy only the records that contain the characters 'CHICAGO, IL' starting in position 45 into a new file.

```
COPYDATA FILE1,,FILE4,,,,,,INCLUDE,45,EQ,'CHICAGO, IL'
```

FILE4 now contains the following records:

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE4       10/24/82      INDEXED         4        1          60             F1   10/24/82   10.41.11

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - 0002
LINE NO -   1    0002MORE SAMPLE DATA    100 - MAIN ST N     CHICAGO, IL

 KEY - 0005
LINE NO -   1    0005CUSTOMER 3          3018 - 5TH ST NE    CHICAGO, IL

    TOTAL RECORDS PRINTED - 2
```

Example 5

This example shows how to copy only those records that have the words 'SAMPLE DATA' in them. Note that because no position is specified, the record is copied no matter where the phrase appears.

```
COPYDATA FILE1,,FILE5,,,,,,INCLUDE,,EQ,'SAMPLE DATA'
```

FILE5 now contains the following records:

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE5       10/24/82      INDEXED         4        1          60             F1   10/24/82   10.41.27

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - 0001
LINE NO -   1    0001SAMPLE DATA         4512 - 18TH AVE NW  NEW YORK, NY

 KEY - 0002
LINE NO -   1    0002MORE SAMPLE DATA    100 - MAIN ST N     CHICAGO, IL

    TOTAL RECORDS PRINTED - 2
```

Example 6

This example shows how to copy only those records that have a value greater than '0002' in positions 1 through 4.

```
COPYDATA FILE1,,FILE6,,,,,,INCLUDE,1,GT,'0002'
```

FILE6 now contains the following records:

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE6       10/24/82      INDEXED         4        1          60             F1   10/24/82   10.41.46

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - 0003
LINE NO -   1    0003CUSTOMER 1          3214 - 5TH ST NE    NEW YORK, NY

 KEY - 0004
LINE NO -   1    0004CUSTOMER 2          2014 - 4TH ST NE    NEW YORK, NY

 KEY - 0005
LINE NO -   1    0005CUSTOMER 3          3018 - 5TH ST NE    CHICAGO, IL

    TOTAL RECORDS PRINTED - 3
```

COPYDIAG Procedure

The COPYDIAG procedure copies a single diagnostic (microcode) diskette. Insert the input diskette in S1 and the output diskette in S2. When the input diskette is read, the data is stored in a disk file. INITDIAG must be run against the diskette(s) you are copying to before you copy diagnostic diskette(s) using COPYDIAG.

The COPYDIAG procedure makes multiple copies of a diskette. The system responds with

```
copy complete, insert diskette for next copy
```

When you have copied the desired number of copies, you must cancel the procedure with either options 2 or 3.

The COPYDIAG procedure runs the $DCOPY utility program.

```
COPYDIAG
```

S9020504-0

The COPYDIAG procedure has no parameters.

Example

To run the COPYDIAG procedure:

```
COPYDIAG
```

COPYI1 Procedure

The COPYI1 procedure copies the files from one or more diskettes onto one or more other diskettes. COPYI1 can be used to create an extra copy of a diskette file or to gather all unused space on one diskette into a single free area on another diskette.

If the system does not have a diskette magazine drive, the SSP reads the data to be copied from the diskette in the drive and then instructs the system operator to remove the original diskette and to insert a new diskette (to receive the copy) into the drive.

If a system has a diskette magazine drive, the COPYI1 procedure can:

- Copy a file on diskette from a diskette slot or a single slot of a magazine to a diskette in a single diskette slot or a single slot of a magazine
- Copy the contents of the diskettes in a magazine from one magazine slot to the other magazine slot
- Make multiple copies of a file from a diskette or an entire diskette to a magazine
- Make multiple copies of a file from a diskette or an entire diskette to a diskette in a single diskette slot
- Make multiple copies of one magazine to the other magazine

The operator must intervene when:

- The same slot location is specified for the diskette or magazine being copied and the diskette or magazine to receive the copy
- Multiple copies of a file from a diskette or an entire diskette are made to a single diskette slot
- Multiple copies of a magazine are made
- More than 10 copies are made to a magazine

A work space large enough to contain the one or more diskette files being copied must be available on the disk.

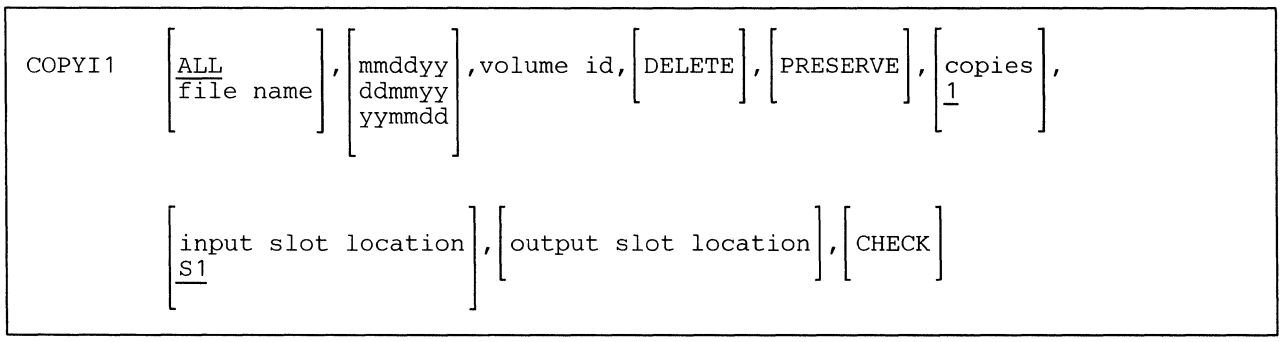
Note: The output diskette must be in the same format as the diskette being copied, either a diskette 1 or a diskette 2D, and initialized in the same format (either FORMAT or FORMAT2; parameters for the INIT procedure).

Diskettes with important permanent files are the diskettes normally copied. Because diskettes can wear out as they are used, it is a good idea to have more than one copy of important diskette files.

If COPYI1 encounters an unreadable sector on the diskette being copied, the unreadable sector is displayed. You can then correct the information in the sector or copy it as is to the diskette receiving the copy. You can use the following function and command keys:

Key	Function
Roll Up	Scans right.
Roll Down	Scans left.
Enter/Rec Adv	Updates the sector being displayed with the changes entered by the operator.
Command Key 1	Displays the next sector.
Command Key 2	Displays the preceding sector.
Command Key 3	If pressed while the unreadable sector is displayed, causes the sector to be copied to the output diskette. If pressed while a sector other than the unreadable sector is displayed, causes the unreadable sector to be displayed.

The COPYI1 procedure runs the $DUPRD utility program.



S9020092-0

ALL specifies that all files on a diskette are to be copied to another diskette or, if the seventh parameter is M1.01, M1, M2.01, or M2 and the eighth parameter is M1, M2, or not specified, then the contents of the diskettes in one magazine are to be copied onto the diskettes in the other magazine. If the first parameter is not specified, ALL is assumed.

file name specifies the file (or saved library) to be copied to another diskette. A name other than ALL is not allowed if the seventh parameter is M1 or M2, or if the seventh parameter is M1.01 and the eighth parameter is M2.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be copied. If ALL is specified as the first parameter, a creation date cannot be specified. The specified date is used to verify that the correct file is on the diskette to be copied. The specified date must be in the same format as the session date; use the STATUS SESSION control command to determine the date format. The output file created by COPYI1 has the same date as the file being copied.

volume id specifies the volume ID of the one or more diskettes to receive the copy; from 1 through 6 alphameric characters can be specified.

Note: If a magazine drive is specified, the volume ID must be the same for all the diskettes in the magazine or an error message is issued. A message option allows future mismatches to be ignored.

DELETE specifies that files that have expired are not to be copied. DELETE can be specified only when ALL is specified.

PRESERVE specifies that the entire file is copied; that is, all the sectors allocated to the file are preserved (even if some of the sectors contain no data).

If PRESERVE is not specified, only the diskette sectors that contain data are copied. That is, any sectors that contain no data are not copied, and the end of the file immediately follows that last copied sector.

copies specifies the number of diskette copies to be made, and can be any number from 1 through 99. If a copies value is not specified, 1 copy is assumed.

input slot location specifies the diskette slot, single slot of a magazine, or magazine slot containing the first diskette to be copied. If no parameter is specified, S1 is assumed. If a single diskette is being copied, the data is copied to diskette slot 2 (S2) if the eighth parameter is not specified. If a magazine is being copied, the data is copied to the opposite magazine (M1 to M2 or M2 to M1) if the eighth parameter is not specified. The system copies the entire magazine if M1, M1.01, or M2 is specified.

output slot location specifies the diskette slot, single slot of a magazine, or magazine slot containing the first diskette to receive the copy. Unless M1, M1.01, or M2 is specified as the seventh parameter, S2 is assumed.

CHECK specifies that the system should continue copying diskettes from a magazine if an empty slot is found or a diskette select failure occurs. In response to an error message, the operator has the option of either trying to read the same diskette again or skipping to the next diskette in the magazine. Skipping to the next diskette causes a corresponding skip of a diskette in the magazine receiving the copy. CHECK is valid only if a magazine is being copied; otherwise, the parameter is ignored. If CHECK is not specified, the system stops copying diskettes from the magazine when the empty slot is found or a diskette select failure occurs.

The valid combinations of these parameters are indicated in the following table. If a combination is not valid, an error code is listed, and an explanation of the error is provided at the bottom of the table.

			Output Slot Value		
Input Slot Value	**Number Copies**	**ALL or File Name**	**S1, S2, or S3**	**M1.01->M1.10 M2.01->M2.10**	**M1 or M2**
S1, S2, or S3	1	ALL	Valid	Valid	1650 error
S1, S2, or S3	1	File name	Valid	Valid	1650 error
S1, S2, or S3	>1	ALL	Valid	1657 error	Valid
S1, S2, or S3	>1	File name	Valid	1657 error	Valid
M1.01	1	ALL	Valid	Valid	Valid[1]
M1.01	1	File name	Valid	Valid	1645 error
M1.01	>1	ALL	Valid	1657 error	Valid[1]
M1.01	>1	File name	Valid	1657 error	Valid
M1 or M2	1	ALL	1655 error	1655 error	Valid[1]
M1 or M2	1	File name	1655 error	1655 error	1645 error
M1 or M2	>1	ALL	1655 error	1655 error	Valid[1]
M1 or M2	>1	File name	1655 error	1655 error	1645 error
M1.01->M1.10 M2.01->M2.10	1	ALL	Valid	Valid	1650 error
M1.01->M1.10 M2.01->M2.10	1	File name	Valid	Valid	1650 error
M1.01->M1.10 M2.01->M2.10	>1	ALL	Valid	1657 error	Valid
M1.01->M1.10 M2.01->M2.10	>1	File name	Valid	1657 error	Valid

[1] Valid if the magazine to be copied is not the same as the magazine to receive a copy.

Error Codes

1645 error indicates an attempt to copy a single file when a magazine is specified to be copied or to receive a copy.

1650 error indicates an attempt to copy a single diskette slot to an entire magazine.

1655 error indicates an attempt to copy an entire magazine to a diskette.

1657 error indicates an attempt to make multiple copies of a diskette to one magazine slot.

Example 1

Copy a diskette file named PAYROLL from the diskette in slot S1 onto a diskette in slot S2 with a volume ID of VOL001.

```
COPYI1 PAYROLL,,VOL001
```

Example 2

Copy files that occupy four diskettes to four other diskettes, if the diskettes that will receive the copy have the same volume ID of VOL002. (If the volume IDs do not agree, an error message is issued.) The four diskettes to be copied are in the first four locations of the magazine in slot M1, and the diskettes that will receive the copy are in the first four locations of the magazine in slot M2.

```
COPYI1 ALL,,VOL002,,,,M1
```

Example 3

Make two copies of all the files on the diskettes in magazine slot 2 to the diskettes in magazine slot 1, if the diskettes that will receive the copies have the same volume ID of VOL003. Expired files are not to be copied, but all of the sectors allocated to the file are to be copied or preserved, even if some of the sectors contain no data. If empty magazine slots are found or if a diskette select failure occurs, an error message is issued to try reading the same diskette again or to skip to the next diskette in the magazine.

```
COPYI1 ALL,VOL003,DELETE,PRESERVE,2,M2,M1,CHECK
```

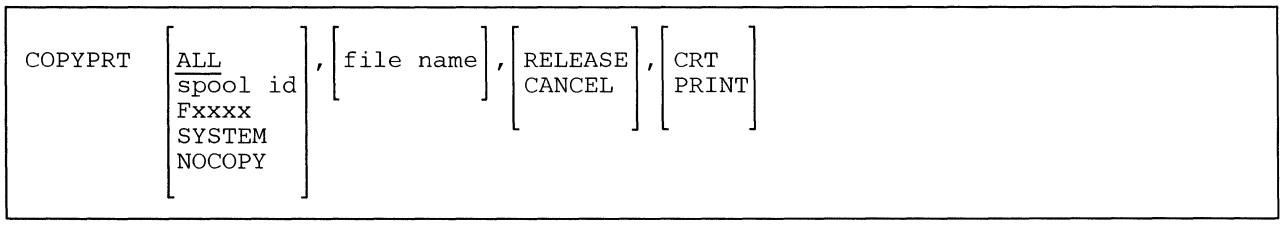
COPYPRT Procedure

The COPYPRT procedure creates a new disk file and copies one or more spool file entries to it. Up to 255 entries can be copied at one time. Selected spool file entries can also be displayed and printed.

| The COPYPRT procedure can also be used to display or print the contents of an existing disk file created by a
| previous COPYPRT procedure.

The COPYPRT procedure cannot copy spool file entries that were created by DW/36, or by PC Support/36 virtual printer facility, or any spool file records that were created in transparent mode. Such records are bypassed during a copy.

The COPYPRT procedure runs the $UASF and $UASC utility programs.



S9020093-1

| **ALL** specifies that all spool file entries for a user are to be copied. If the first parameter is not specified, ALL is assumed. Only the entries not being processed by spool and those entries having the same user ID as the operator that entered the procedure are copied.

spool id specifies the 6-character spool file ID of the spool file entry to be copied. Entries being processed by spool are not copied.

If password security is not active, operators can copy any spool file entries. If password security is active, the operator must have a security classification of system operator or higher to copy any spool ID belonging to another operator; otherwise, the operator can only copy entries that belong to him or her.

Fxxxx specifies the forms number of the one or more spool file entries to be copied. **xxxx** is the 4-character forms number of the entries to be copied. Entries being processed by spool are not copied.

If password security is not active, operators can copy any spool file entries with the specified forms number. If password security is active, the operator must have a security classification of system operator or higher to copy any spool entry belonging to another operator; otherwise, the operator can only copy entries that belong to him or her.

For more information about forms numbers, see the "PRINT Procedure" on page 4-350, "FORMS OCL Statement" on page 5-55, or "PRINTER OCL Statement" on page 5-83.

SYSTEM specifies that all spool file entries not being processed by spool are to be copied.

If password security is not active, all entries can be copied. If password security is active, the operator must have a security classification of system operator or higher to copy all entries.

NOCOPY specifies that a disk file that contains one or more spool file entries previously copied using COPYPRT are to be displayed or printed. You can also display or print history file information that was copied by using the HISTORY COPYPRT procedure; see the "HISTORY Procedure" on page 4-209 for more information. If NOCOPY is specified, RELEASE or CANCEL are invalid.

file name specifies the new disk file that is to be created and contain the copied spool file entries (or that is to be displayed or printed if NOCOPY is specified). A file name must be specified if ALL, SYSTEM, or NOCOPY is specified as the first parameter or if the first parameter is not specified.

If a file name is not specified but a spool ID or forms number is specified, the file name assumed is either the spool ID or Fxxxx and the 2-character ID of the display station. For example, if the operator at display station W3 entered the COPYPRT procedure specifying a spool ID of SP0032, the file name would be SP0032W3; if a forms number of 0001 were entered by the operator at display station W3, the file name would be F0001W3.

The format of the file is described in "File Format for COPYPRT Files" on page 4-128.

RELEASE specifies that when the COPYPRT procedure has copied the spool file entries to a disk file, the spool file entries specified by parameter 1 are to be released for printing.

CANCEL specifies that when the COPYPRT procedure has copied the spool file entries to a disk file, the spool file entries specified by parameter 1 are to be removed from the spool file.

CRT specifies that the spool file entries are to be displayed. You can also use this option to print the entries that were copied. Figure 4-2 shows how the entries are displayed and explains the fields on the display. CRT is assumed if the first parameter is NOCOPY and the fourth parameter is omitted.

PRINT specifies that the spool file entries are to be printed. PRINT is allowed only if NOCOPY is also specified.

| *Note: The file to be printed must have been created on Release 4 or a later release of the System/36. Using this PRINT parameter to print a file created by a previous release may cause errors to occur.*

If this parameter is specified, each copied spool file entry in the input file is printed. The values for FORMSNO, COPIES, ALIGN, LINES, LPI, CPI, FONT, and JUSTIFY with which the original spool entry were created are preserved.

If run from a work station, the printer allocated will be the session printer, otherwise the system printer will be used.

If this parameter is specified, there is no requirement for a work station. The file is printed and COPYPRT terminates. Print requests may be evoked or placed on the job queue.

Example 1

Copy spool file entry SP0001 into a file named SP0001W2. The COPYPRT procedure is entered from display station W2.

```
COPYPRT SP0001
```

Example 2

Copy all spool file entries not being processed by the spool into a file named SPFILE.

```
COPYPRT SYSTEM,SPFILE
```

Example 3

Copy all spool file entries with forms number 0017 into a file named F0017W3. The COPYPRT procedure is entered from display station W3.

```
COPYPRT F0017
```

Example 4

To copy all your spool file entries into a file named TEMP, cancel the entries from the spool file, and display the entries.

```
COPYPRT ,TEMP,CANCEL,CRT
```

Example 5

To print all of the copied spool file entries in an existing file named TEMP:

```
COPYPRT  NOCOPY,TEMP,,PRINT
```

File Format for COPYPRT Files

The disk file created by the COPYPRT procedure has a record length of 150, 215, or 248 bytes. The record length is 150 if all of the copied spool file entries have print lengths less than 133. The record length is 215 if the longest copied spool file entry has a print length between 132 and 199. The record length is 248 if the longest copied spool file entry has a print length greater than 198.

For each entry in the spool file copied to the disk file, one header record is created. The header record contains the following fields:

Beginning Column	Field Length	Contents or Description
1	1	The letter H (to indicate the header)
4	6	The spool ID of the entry
12	8	The procedure name
22	8	The job name

Beginning Column	Field Length	Contents or Description
32	8	The user ID of the operator
42	8	The printer file name
52	2	The printer ID
56	4	The forms number
61	2	The number of copies (in binary)
65	2	The number of pages (in binary)
69	4	The number of records (in binary)
74	2	The number of lines per page (in binary)
78	1	The letter I if this entry contains print records with IGC data (otherwise a blank)
81	1	The letter M if this entry contains print records with a length greater than 132 (otherwise a blank)
84	1	Lines per inch (in binary)
85	1	Characters per inch (in binary)
86	1	Font ID (in binary)
87	1	Justify value (in binary)
88	1	Align (Y or N)
89	1	Constant of zero (in binary)
90	1	Record length (in binary)

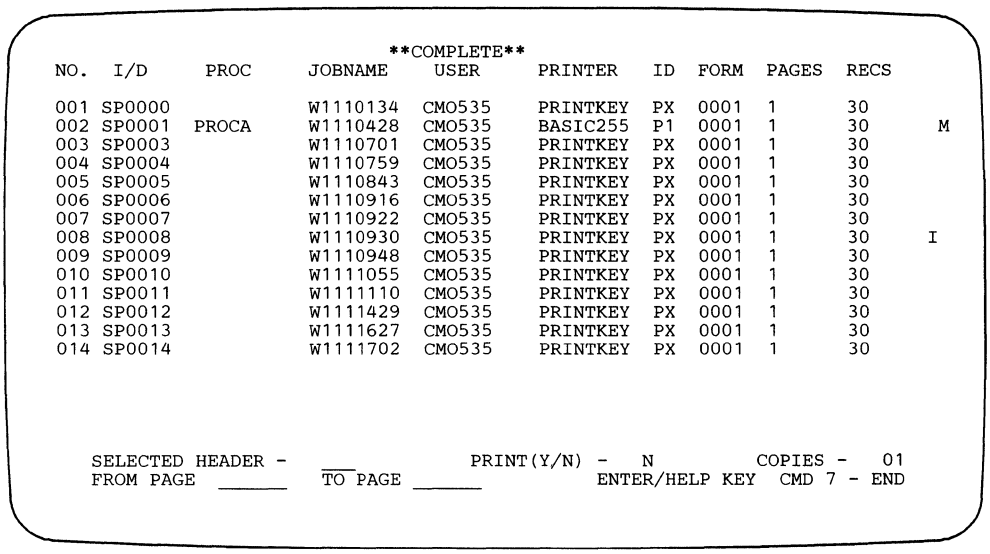
All positions not specified above are set to blanks.

Following the header record are the data character records. These records have a record length of 150, 215, or 248. The data character records contain the following fields:

Beginning Column	Field Length	Contents or Description
1	2	The page number (in binary)
3	2	The line number (in binary)
5	4	The record number (in binary)
9	1	The letter I if this print record contains IGC data
10	1	An IGC shift-out character (hex 0E) if this print record starts with IGC data
11	132, 198, or 225	The data to be printed

All positions not specified above are set to blanks.

You can press the Help key to get information about the following display.



```
                              **COMPLETE**
NO.  I/D      PROC     JOBNAME    USER      PRINTER    ID  FORM  PAGES  RECS

001 SP0000             W1110134  CMO535    PRINTKEY   PX  0001  1      30
002 SP0001   PROCA     W1110428  CMO535    BASIC255   P1  0001  1      30      M
003 SP0003             W1110701  CMO535    PRINTKEY   PX  0001  1      30
004 SP0004             W1110759  CMO535    PRINTKEY   PX  0001  1      30
005 SP0005             W1110843  CMO535    PRINTKEY   PX  0001  1      30
006 SP0006             W1110916  CMO535    PRINTKEY   PX  0001  1      30
007 SP0007             W1110922  CMO535    PRINTKEY   PX  0001  1      30
008 SP0008             W1110930  CMO535    PRINTKEY   PX  0001  1      30      I
009 SP0009             W1110948  CMO535    PRINTKEY   PX  0001  1      30
010 SP0010             W1111055  CMO535    PRINTKEY   PX  0001  1      30
011 SP0011             W1111110  CMO535    PRINTKEY   PX  0001  1      30
012 SP0012             W1111429  CMO535    PRINTKEY   PX  0001  1      30
013 SP0013             W1111627  CMO535    PRINTKEY   PX  0001  1      30
014 SP0014             W1111702  CMO535    PRINTKEY   PX  0001  1      30




   SELECTED HEADER - ___          PRINT(Y/N) -  N        COPIES -  01
   FROM PAGE ______  TO PAGE ______          ENTER/HELP KEY  CMD 7 - END
```

Figure 4-3. Sample COPYPRT Display Showing the Spool Header Entries

Description of Fields

NO. specifies the number of the entry. You enter this number into the SELECTED HEADER field to display (or print) the entry.

I/D specifies the spool ID of the entry.

PROC specifies the name of the procedure that created the entry.

JOB NAME specifies the system-assigned name of the job that created the entry.

USER specifies the user ID of the operator that created the entry.

PRINTER specifies that name of the printer file assigned to the entry.

ID specifies the printer ID.

FORM specifies the forms number associated with the entry.

PAGES specifies the number of pages for the entry.

RECS specifies the number of print records associated with the entry. This number does not include any transparent mode print records that may have been in the spool entry prior to copying.

I specifies that the print entry contains ideographic data.

M specifies that the print entry contains records with a length greater than 132.

SELECTED HEADER specifies the header number you want to display or print.

PRINT Y/N specifies whether the entry is to be printed. Enter Y to print the entry; enter N to display the entry. If Y is specified, the entry will be printed on the printer used during that display station session. The forms number, number of lines per page, number of lines per inch, and number of characters per inch parameters will also assume the session values. N is assumed.

COPIES specifies the number of copies to be printed. You can enter a number from 1 through 99. If no entry is specified, 1 is assumed. This parameter is valid only if PRINT Y is specified.

FROM PAGE specifies the beginning page to be printed. If no entry is specified, the printing begins from the first page. This parameter is valid only if PRINT Y is specified.

TO PAGE specifies the ending page to be printed. If no entry is specified, the printing ends with the last page. This parameter is valid only if PRINT Y is specified.

For example, Figure 4-3 shows the display that appears when you select header 1:

```
001 SP0000            W1110134  CMO535    PRINTKEY  PX  0001  1      30

       1...*...10....*...20....*...30....*...40....*...50....*...60....*.
     PAGE   1
6           0    +    1    +    2    +    3    +    4    +    5    +    6
7           1    +    0    +    0    +    0    +    0    +    0    +    0
8          ************************************************************
9       01 *
10      02 *                                  SCRNPRT PROGRAM
11      03 *
12      04 *     This program lists the Print key display images contained
13      05 *
14      06 *
15      07 *     Name of the disk file that contains the display images . .
16      08 *
17      09 *     ID of printer to be used . . . . . .
18      10 *
19      11 *     Number of copies to print  . . . . . .
20      12 *


  PAGE NO     ______          LINE NO    ___       DISP START POS    ___
RECORD NO     ______                               ENTER/ROLL DN/ROLL UP/HELP KEY
      CMD 1 - REDISPLAY HEADERS      CMD 2 - RECORD MODE     CMD 7 - END
```

Figure 4-4. Sample COPYPRT Display Showing a Page from a Copied Display

Description of Fields

PAGE NO specifies the page to be displayed. Enter a number into this field to display a specific page.

LINE NO specifies the print line to be shown at the top of the display. Enter the number of a print line to cause that line of the current or specified page to be shown at the top of the display.

DISP START POS specifies the leftmost starting position of the columns to be displayed. Enter a number into this field to shift the columns displayed to the right or left.

RECORD NO specifies the print record to be shown at the top of the display. Enter the number of a record to cause that record to be shown at the top of the display.

CREATE Procedure

The CREATE procedure generates a message load member from a message source member. A message member contains messages that can be displayed or printed by a program, on display formats, or by the "ERR Procedure" on page 4-176. You can also use the "// * (Informational Message) Statement" on page 3-57 and the "// ** (System Console Message) Statement" on page 3-59 to display messages. The "?Mmic? or ?M'mic,position,length'? (Message Member) Expression" on page 3-21 can also be used to get a message from a message load member.

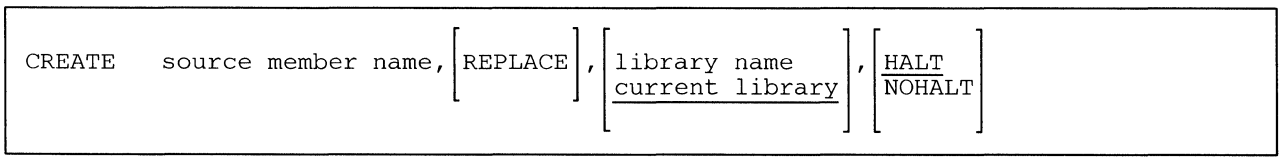
The input to the CREATE procedure is a message source member. The message source member contains three types of statements:

- The message control statement
- One or more message text statements
- One or more comment statements (optional)

These statements are described in "Message Member Statements" on page 4-133. If you have the ideographic version of the SSP, refer to "Considerations for the Ideographic Version of the SSP" on page 4-135.

You can create the source member using the source entry utility (SEU) or development support utility (DSU). SEU is described in detail in the *SEU Guide*. DSU is described in detail in the *Development Support Utility Guide*. You can also use the $MAINT utility program to create source members. See "Create Source or Procedure Members" on page A-57 for more information.

The CREATE procedure runs the $MGBLD utility program.



S9020094-0

source member name specifies the existing message source member that contains the message source statements.

REPLACE specifies that the created message load member is to replace an existing load member with the same name. The name of the load member is specified by the message control statement in the source member.

If REPLACE is not specified, a load member with the same name is not to be replaced. If a load member with the same name does exist, a message is displayed and the operator can either replace the load member or cancel the job.

library name specifies the library that contains the source member and into which the load member is to be placed. If a library name is not specified, the current library is assumed.

HALT specifies that if errors are encountered in the message source member, a message should be displayed. If no parameter is specified, HALT is assumed.

NOHALT specifies that no message should be displayed if errors are encountered in the message source member.

Example

To generate a message load member from a source member named MESSAGES that is contained in the library MYLIB, you would enter the CREATE procedure as follows:

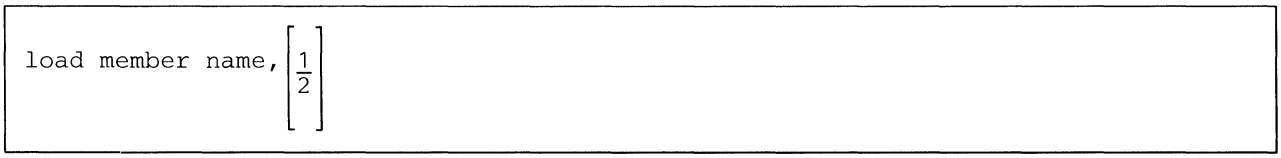
```
CREATE MESSAGES,,MYLIB
```

Message Member Statements

The Message Control Statement

The message control statement specifies the name of the message load member to be created by the CREATE procedure, and whether the message member is to be a first- or second-level message member. The message control statement must be the first line in the source member, and only one is allowed.

The format of the message control statement is:



S9020095-0

load member name specifies the message load member to be created. A member name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because these have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), and equal signs (=). Do not use DIR, LIBRARY, or ALL as a member name.

1 specifies that the member is a first-level message member. First-level message members can have up to 75 characters of message text. If the message level parameter is not specified, 1 is assumed.

2 specifies that the member is a second-level message member. Second-level message members can have up to 225 characters of message text.

This example shows the format of a message control statement. The name to be assigned to the load member is MESSAGES; the load member is to be a first-level message member.

```
MESSAGES,1
```

Message Text Statement

The message text statement specifies a message identification code (MIC) and the message text for a message.

The format of the message control statement is:

```
mic  message
```

S9020096-0

mic specifies the message identification code (MIC) of the message. The MIC must be a 4-digit number from 0000 through 9999; it must be placed in positions 1 through 4 of the message text statement. The MICs in a message member must be in ascending order. The same MIC can be specified more than once for statements that have a large amount of text. The number of statements that can specify the same MIC is limited to the number of statements required to specify up to 75 characters for first-level messages and up to 225 characters for second-level messages.

message specifies the text of the message. The text area begins at position 6 in the text statement and continues through the end of the source member line. The text for the message is the characters starting in position 6 to the last nonblank character in the line.

This example shows the format of a message text statement. The mic number is 0014; the text of the message follows the number.

```
0014 This is a message
```

If the message is continued on another source line, the text area on the following line is added to the end of the previous line, including one or more blanks from the previous line. For example:

```
         Column Indicators
         1         2         3         4
12345678901234567890123456789012345678901

MESSAGE,2
0001 This is the first message and it
0001 is continued on this line.
```

Statements are read up to their statement length, which is set when the member is created (either by SEU or $MAINT). In a source member having a statement length of 40, the preceding lines would produce a second-level message load member named MESSAGE. Message 0001 would contain the following message (note the spaces in the combined message caused by the spaces following that first line of message text):

```
This is the first message and it is continued on this line.
```

The message text may contain replacement text fields. These fields allow a single message to be created and data within the message to be specified; for example, a file name, a user ID, or the time of day when the message is issued. Multiple fields may exist and can be placed anywhere within the message. The total of the field lengths is limited to 75 characters. Pound symbol (#) characters must be used as place holders in the message text to indicate where the replacement text fields are located. The text used to replace the # characters must be specified as the text parameter on the "ERR Procedure" on page 4-176. See the *Concepts and Programmers Guide* for further information and examples of using replacement text in displayed messages.

Comment Statement

The comment statement is optional. The format of the comment statement is as follows:

```
* comment
```

S9020097-0

The asterisk (*) must be in the first position in the statement. This statement does not become part of the message load member. The comment statement cannot be the first statement in the message source member.

Example Message Source Member

Assume a message source member contains the following statements. Note how the messages for the two programs are grouped.

```
MESSAGES,1
* The following messages are for the program PAYROLL (00xx)
0001 Enter yesterday's date:
0002 Enter today's date:
0003 Enter tomorrow's date:
* The following messages are for the program ACCTS   (01xx)
0101 Accounts Payable program is running.
0102 Make sure printer P3 is powered on.
0103 Accounts Payable program is complete.
```

Considerations for the Ideographic Version of the SSP

For systems with the ideographic version of the SSP, you can create a message member that has either ideographic or nonideographic messages, or a message member that contains both ideographic and nonideographic messages.

If the message member contains only nonideographic characters, you create the source member the same way as for a system without the ideographic version of the SSP.

If the message member contains only ideographic messages, the source member should have a MIC of A000 preceding the message text statements. For example:

```
MESSAGE,2
* Ideographic Message Member
A000
0001    (ideographic text)
0002    (ideographic text)
.
.
.
9999    (ideographic text)
```

If the message member contains both ideographic and nonideographic messages, the source member should have a MIC of A000 following the nonideographic messages and preceding ideographic messages. The MICs for each type of message can be the same. For example:

```
MESSAGE,2
* Nonideographic Message Portion
0001    (nonideographic text)
0002    (nonideographic text)
.
.
.
9999    (nonideographic text)
* Ideographic Message Portion
A000
0001    (ideographic text)
0002    (ideographic text)
.
.
.
9999    (ideographic text)
```

If an operator is signed on to an ideographic session, messages are retrieved from the ideographic portion of the message member. If the MIC is not found in the ideographic portion, the message is retrieved from the nonideographic portion of the message member.

If a job is evoked or placed on the input job queue, or the operator is not signed on to an ideographic session, the messages are retrieved from the nonideographic portion of the message member.

For the ideographic version of the SSP, the following special conditions exist when you continue message text from one message text statement to another:

- If a shift-in character is in one of the last two positions in a line and if a shift-out character is in the sixth position of the continued line, the shift-in/shift-out pair is removed along with any blanks between the pair when the lines are joined in the CREATE procedure.

- If the last two positions in the first line contain a shift-out character followed by a shift-in character, the shift-out/shift-in pair is removed when the lines are continued during the CREATE procedure.

DATE Procedure

The DATE procedure changes the session date or the program date.

If you enter the DATE procedure and it is not between a LOAD OCL statement and a RUN OCL statement, the session date is changed. A session begins when an operator signs on and ends when the operator enters the OFF command. If the DATE procedure is not used to establish a session date, the system date specified during initial program load (IPL) is used as the session date.

If you enter the DATE procedure between a LOAD OCL statement and a RUN OCL statement in a procedure, it specifies the program date; that is, the date the program uses. The program date is also called the job step date. When the program ends, the session date is used as the program date. If the DATE procedure is not entered between a LOAD statement and a RUN statement, the session date is used as the program date. If you enter two or more DATE procedures between a LOAD statement and a RUN statement, the last DATE procedure is used.

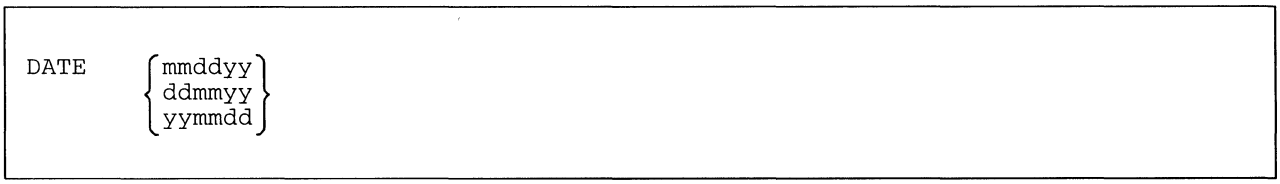
The DATE procedure processes a DATE OCL statement.

Notes:

1. *The program date is used to determine the file retention period for diskette files used by the program and can be printed on the printed output. The program date is also used as the creation date of any disk or diskette files created by the program.*

2. *If a job is placed on the input job queue, the program date when the job was placed on the queue is assigned to the job.*

3. *If 2400 hours (midnight) occurs, the system date is automatically updated, but the session date and the program date are not.*



S9020098-0

The date specified on the DATE procedure must be in the current session date format. The session date can be in any of three formats: month-day-year (mmddyy), day-month-year (ddmmyy), or year-month-day (yymmdd). You can use the STATUS SESSION command to show the current session date format, and the SET procedure can change the current session date format.

mm, dd, and **yy** must each be a 2-digit number, but leading zeros in month and day can be omitted when punctuation is used. A date of all zeros (000000) is not allowed. The date can be entered with or without punctuation. For example, July 24, 1984 can be specified in any of the following ways:

```
7-24-84         mm-dd-yy
24-7-84         dd-mm-yy
84-7-24         yy-mm-dd
072484          mmddyy
240784          ddmmyy
840724          yymmdd
```

In the punctuated form, any characters except commas (,), quotes ('), numbers, and blanks can be used as punctuation. However, if the DATE procedure is within a procedure, question marks (?), slashes (/), equal signs (=), plus signs (+), and hyphens (-) should be used with caution because these characters have special meanings within procedures.

Example 1

To specify a session date for July 1, 1984, the operator would enter the following from the keyboard:

```
DATE 7-1-84
```

or:

```
DATE 070184
```

Example 2

To specify a job date for the PAYROLL and PAYPRNT programs, you could place the DATE procedure before the first LOAD OCL statement in the procedure.

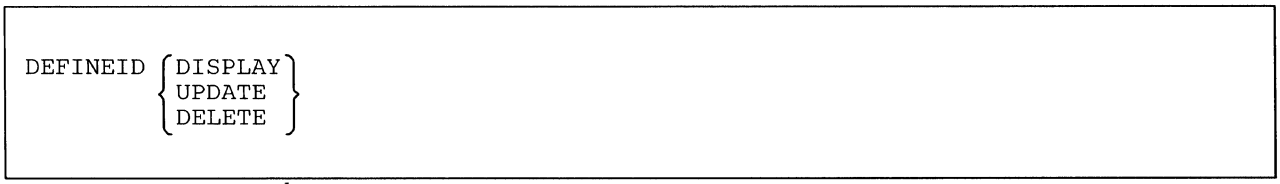
```
// * 'ENTER PAYROLL DATE'
DATE ?1R?
// LOAD PAYROLL
// FILE NAME-EMPLOYES
// FILE NAME-CHECKS,RETAIN-J,BLOCKS-100
// RUN
// LOAD PAYPRNT
// FILE NAME-CHECKS,RETAIN-J
// PRINTER NAME-PRINT,SPOOL-NO,LINES-25,FORMSNO-CHCK,
//         ALIGN-YES,DEVICE-P3
// RUN
```

DEFINEID Procedure

The DEFINEID procedure is used to specify a list of the remote IDs for a switched communications line using the SSP-ICF BSCEL subsystem. When the System/36 using BSCEL is connected to a remote system by a switched communications line, this list is examined for a comparison with the ID received from the remote system. This list is only examined when multiple remote switched line IDs were specified when the subsystem was configured. If the ID is found, line initialization completes successfully. If it is not found, the remote system is disconnected.

The DEFINEID procedure runs the $IDSET utility program. The DEFINEID procedure creates a system file named #IBSRID, which is four sectors long and can contain up to 55 remote IDs.

The DEFINEID procedure is described in the manual *Using System/36 Communications*.



S9020099-0

DISPLAY specifies that the remote IDs are to be displayed. The remote IDs cannot be modified. This parameter is valid only if remote IDs are defined.

UPDATE specifies either that the list of remote IDs is to be created, or that the existing list of remote IDs is to be changed.

DELETE specifies that the entire list of remote IDs (file #IBSRID) is to be deleted from disk. This parameter is valid only if remote IDs are defined.

Example

To create or update the remote ID list:

```
DEFINEID UPDATE
```

DEFINEPN Procedure

The DEFINEPN procedure provides a way to create or update a phone list for the autocall feature. Each phone list can contain up to 105 phone numbers. The phone lists are created by the DEFINEPN procedure and are used by the autocall feature. The DEFINEPN procedure is described in the manual *Using System/36 Communications*.

The DEFINEPN procedure runs the $PNLM utility program.

```
DEFINEPN
```

S9020100-0

The DEFINEPN procedure has no parameters.

Example

To run the DEFINEPN procedure:

```
DEFINEPN
```

DEFINLOC Procedure

The DEFINLOC procedure allows you to set up a list of names and location IDs of remote locations that you allow to call your subsystem. (This procedure is valid only for an asynchronous subsystem using X.25 support).

For more information about DEFINLOC, see the manual *Using System/36 Communications*, SC21-9143.

```
DEFINLOC
```

S9020516-0

The DEFINLOC procedure has no parameters.

Example

This example shows how to set up a list of names that you allow to call your subsystem:

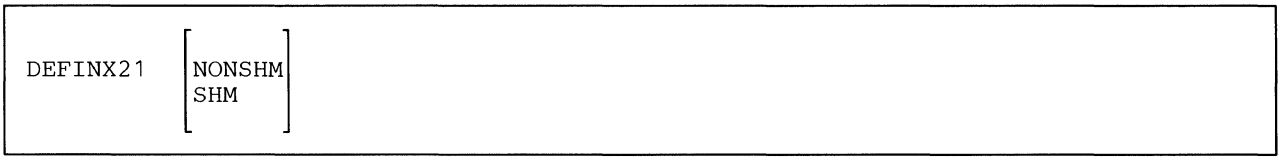
```
DEFINLOC
```

DEFINX21 Procedure

The DEFINX21 procedure provides a way to create or update a list of public data network connection numbers for the X.21 feature or a short hold mode line configuration. The DEFINX21 procedure also builds the library #X21LIB for short hold mode if it does not already exist.

The DEFINX21 procedure is described in the manual *Using System/36 Communications*.

The DEFINX21 procedure runs the $XNLM and $XNSH utility programs.



S9020101-1

NONSHM specifies that you want to create or update an X.21 phone list that will not use short hold mode support. Each list can contain up to 75 connection numbers. The lists created by the DEFINX21 procedure are stored as library load members.

SHM specifies that you want to create, update, print, or remove a short hold mode line configuration. Each line configuration consists of local line information. This information is stored as a load member in library #X21LIB. SHM helps reduce line usage by disconnecting the circuit switched line when there is no line activity.

Example 1

To create or update a phone list that will not use short hold mode support:

```
DEFINX21 NONSHM
```

Example 2

To update a short hold mode line configuration:

```
DEFINX21 SHM
```

DEFINX25 Procedure

The DEFINX25 procedure allows you to create and edit a list of remote network addresses used for the X.25 feature. The DEFINX25 procedure also allows you to create or edit a list of phone numbers used with the rotary dial function when using the asynchronous subsystem. The DEFINX25 procedure is described in the manual *Using System/36 Communications*.

```
DEFINX25
```

S9020102-0

The DEFINX25 procedure has no parameters.

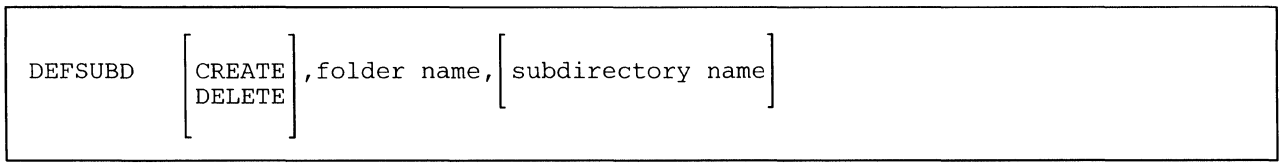
Example

To run the DEFINX25 procedure:

```
DEFINX25
```

DEFSUBD Procedure

The DEFSUBD procedure creates or deletes a subdirectory or presents a subdirectory display. Folder name is always required whether using either the CREATE or DELETE option, or no option. The subdirectory name is required with only the CREATE and DELETE options.



S9020596-0

CREATE specifies that you want to create a subdirectory.

DELETE specifies that you want to delete a subdirectory.

folder name specifies the name of the folder.

subdirectory name specifies the name of a subdirectory.

Example

This example indicates how to show the Work with Subdirectory display.

```
DEFSUBD
```

DELETE Procedure

The DELETE procedure removes one or more files, libraries, or folders from a diskette or disk. The DELETE procedure can be used to delete files on a remote system if they are defined in the network resource directory (NRD).

The DELETE procedure cannot be used to delete a file, library, or folder while it is being used by another job or if a file statement was specified within the job step when the file name and file label for that statement are different. It also cannot be used to delete a file that has one or more alternative index files on disk. The system library (#LIBRARY), the system files (such as the history file, #SYSHIST), job files, and scratch files cannot be deleted with this procedure. IDDU dictionaries with externally-described files linked to them cannot be deleted.

To remove one or more members from a library, see the "REMOVE Procedure" on page 4-370.

The DELETE procedure runs the $DELET utility program. For information on how to delete **all** files, libraries, and folders from disk, refer to "$DELET Utility" on page A-40.

To delete one or all files from **diskette**:

```
DELETE   {file name}, [I1], [SCRATCH], [mmddyy],, [starting location], [ending location  ], [volume id]
         {ALL      }        [REMOVE ]  [ddmmyy]   [S1              ]  [starting location]
                            [ERASE  ]  [yymmdd]   [S2              ]  [S1              ]
                                                  [S3              ]  [S2              ]
                                                  [M1.nn           ]  [S3              ]
                                                  [M2.nn           ]  [M1.nn           ]
                                                                      [M2.nn           ]
```

S9020103-0

To delete a file, library, folder, or a group of files, libraries, or folders from **disk**:

```
DELETE   {file name   },F1, [SCRATCH], [mmddyy], [LIBR  ],,,, [group name]
         {library name}     [REMOVE ]  [ddmmyy]  [FOLDER]
         {folder name }     [ERASE  ]  [yymmdd]  [ALL   ]
         {ALL         }
```

S9020104-0

file name, library name, or folder name specifies the file to be deleted from diskette or the file, library, or folder to be deleted from disk.

For systems with a diskette magazine drive: If a file is contained on more than one diskette and no ending location is specified, processing continues until the file is deleted from the last diskette; however, processing will not go from slots to magazines or from magazines to slots.

If a slot (S1, S2, or S3) is specified for the starting location, processing remains within the slots. Processing will begin at the specified starting location and continue through S3. Then, if the file is contained on additional diskettes, they will be processed beginning in the specified starting location and continue through S3.

If a position in a magazine (M1.nn or M2.nn) is specified for the starting location and no ending location is specified, processing remains within the specified magazine (M1 or M2). Processing will begin at the specified starting location and continue through M1.10 or M2.10. Then, if the file is contained on additional diskettes, they will be processed beginning in the specified starting location and continue through M1.10 or M2.10.

For example, file FILEA is contained on three diskettes and S2 is specified as the starting location but no ending location is specified. The file is deleted from the diskettes in S2 and S3. A message is then displayed asking for the next diskette to be placed into S2. The file is then deleted from S2, and processing stops.

ALL specifies the following:

- If I1 is specified, all the files on diskette are to be deleted.
- If F1 is specified, a group name must be specified. All the members of the file group are to be deleted.
- If F1 and LIBR are both specified, and a group name is also specified in the ninth parameter position, all files and libraries that are members of the file group are to be deleted.
- If F1 and FOLDER are both specified, and a group name is also specified in the ninth parameter position, all files and folders that are members of the file group are to be deleted.
- If F1 and ALL are both specified, and a group name is also specified in the ninth parameter position, all files, libraries, and folders that are members of the file group are to be deleted.

I1 specifies the file is to be deleted from diskette. If a second parameter is not specified, I1 is assumed. If the file is contained on more than one diskette, a message may be displayed and the operator may have to insert the next diskette. If I1 is specified and ALL is specified as the file name, a message will be displayed that says that the end of the diskette has been reached. You can delete all files from another diskette with the same volume ID by inserting it in the slot and taking option 0.

F1 specifies the file, library, or folder to be deleted from the disk.

SCRATCH specifies the following:

- If the file is on **diskette**, the expiration date is to be set to the current job step date. The file remains on diskette and can be accessed only for reading. If you want to reuse free space on a diskette, this space must follow the last active file. **If you write any new information** onto the diskette, the file no longer exists.

- If the file, library, or folder is on the **disk**, the VTOC entry for it is to be removed. This allows the space in the VTOC and on the disk to be used for other data. For the disk, SCRATCH and REMOVE do the same function.

If no third parameter is specified, SCRATCH is assumed.

Note: Although the file no longer exists on disk or diskette, the data has not been erased and can be read using the "PATCH Procedure" on page C-17.

REMOVE specifies, for either diskette or disk, that the VTOC entry for the file, library, or folder is to be removed. This allows the space in the VTOC and on disk or diskette to be used for other data. If you want to reuse free space on a diskette, this space must follow the last active file. For the disk, SCRATCH and REMOVE do the same function.

Note: Although the file no longer exists on disk or diskette, the data has not been erased and can be read using the "PATCH Procedure" on page C-17.

ERASE specifies that the VTOC entry for the file, library, or folder is to be removed. This allows the space in the VTOC and on disk or diskette to be used for other data. If you want to reuse free space on a diskette, this space must follow the last active file. Also, the data that was contained in the deleted file, library, or folder is removed. That is, all characters in the file, library, or folder are replaced with zeros. (ERASE prevents anyone from reading the data using a system procedure, such as PATCH, that can display anything on disk or diskette.)

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be deleted. For a disk file, the specified date must be in the same format as the session date. For a file on diskette, the specified date must be in the same format as the creation date of the file on diskette.

If a library or folder is being deleted from disk (LIBR or FOLDER is specified), a date cannot be specified. If ALL is specified, a date cannot be specified.

Notes:

1. *If no date is specified and more than one file with the specified name exists on the disk, the operator can delete all the files with that name or cancel the job.*

2. *If no date is specified and more than one file with the specified name exists on a diskette, only the first file (in terms of location) with the specified name is deleted. The first file is determined by the physical placement of the file on the diskette. You can determine the placement of the files on diskette by using the CATALOG procedure to list the entries by location.*

LIBR specifies that a library is to be deleted from disk. If LIBR or FOLDER is not specified, a file is to be deleted from disk. LIBR can be specified only if F1 is specified.

FOLDER specifies that a folder is to be deleted from disk. If LIBR or FOLDER is not specified, a file is to be deleted from disk. FOLDER can be specified only if F1 is specified.

ALL specifies that files, libraries, and folders that are members of the specified file group are to be deleted from disk. ALL can be specified only if F1 is specified.

starting location specifies the starting location of a diskette slot to be used by DELETE. If this parameter is not specified, S1 is assumed. This parameter is ignored if the system does not have a diskette magazine drive. The starting location must be less than the ending location.

ending location specifies the ending location of a diskette slot to be used by DELETE. If this parameter is not specified, the starting location is assumed. This parameter is ignored if the system does not have a diskette magazine drive.

The ending location (if specified) must be greater than or the same as the starting location and must not cause a jump from the the diskette slots (S1, S2, and S3) to the magazine slots (M1 and M2). That is, specifying a starting location of S3 and an ending location of M1 is not correct. You can specify a starting location in magazine 1 and an ending location in magazine 2.

The starting and ending locations can be any of the following:

S1, S2, or S3 specifies one of the three diskette slots.

M1 or M2 specifies one of the two diskette magazine slots. If M1 or M2 is specified for the starting location, and no ending location is specified, the entire magazine (all 10 slots) is processed. If M1 or M2 is specified for the ending location, it is the same as specifying M1.10 or M2.10, respectively.

M1.nn or M2.nn specifies a specific position within a magazine. **nn** is a number from 01 through 10, specifying the position within the magazine.

volume id specifies the volume ID of the one or more diskettes to be processed. The volume ID is used to ensure that the correct diskettes are being processed. A volume ID must be entered if the first parameter is ALL and the second parameter is I1.

If the specified volume ID matches the volume ID of the diskettes, the files are deleted. If the specified volume ID does not match the volume ID of the diskettes, a message is displayed and the operator can either insert the correct diskettes or cancel the procedure.

group name specifies the file group to be deleted from disk. For example, to delete files AB.F1 and AB.F2, which make up file group AB, you would enter AB without the period. The first parameter must be ALL; the second parameter must be F1.

Example 1

Delete the file named PAYROLL from the diskette in slot 1.

```
DELETE PAYROLL,,REMOVE
```

Example 2

Delete the library named MYLIB from disk.

```
DELETE MYLIB,F1,,,LIBR
```

Example 3

Delete the folder named MYFLDR from disk.

```
DELETE MYFLDR,F1,,,FOLDER
```

Example 4

Delete all the files on a diskette with a volume ID of VOL001. The diskette is in slot 2.

```
DELETE ALL,I1,,,,S2,,VOL001
```

Example 5

Delete all the files in the file group named FIL from disk. The files in this group are named FIL.A, FIL.B, and FIL.C.

```
DELETE ALL,F1,,,,,,,FIL
```

Example 6

Delete all the files from the diskettes in magazine slot 1 from position 1 through position 5. The diskettes have a volume ID of VOL001.

```
DELETE ALL,I1,REMOVE,,,M1.01,M1.05,VOL001
```

Example 7

Remove the PAYROLL file from diskette, and erase all the data that was contained in the file. The diskette is in slot 1.

```
DELETE PAYROLL,,ERASE
```

DELNRD Procedure

The DELNRD procedure removes the network resource directory (#NRD.FLE) from disk. The directory cannot be removed while it is being used by another job.

For more information about the network resource directory, see the *Distributed Data Management Guide.*

The DELNRD procedure runs the $SINDL utility program.

```
DELNRD
```

S9020105-0

The DELNRD procedure has no parameters.

DFU Procedure

The DFU procedure displays a menu from which you can select an option to run an existing data file utility (DFU) program, or to create or change a DFU program. Using the DFU program, you can create, update, list, or display data in a disk file. For more information about DFU, see the *DFU Guide.*

```
DFU
```

S9020106-0

Example

This example shows how to display the DFU menu.

```
DFU
```

DFULOAD Procedure

The DFULOAD procedure creates a library named #DFULIB and copies the data file utility (DFU) support from diskette into that library. DFULOAD copies additional support into the system library (#LIBRARY). The DFULOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the DFUSAVE procedure. See the "DFUSAVE Procedure" on page 4-152 for information about how to save the DFU support on diskette.

The DFULOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the DFULOAD procedure to restore support that has been saved by DFUSAVE.

If DFU is not currently on the system, you must use the TOLIBR procedure to copy the diskette file DFU into #LIBRARY before running DFULOAD.

If #LIBRARY was backed up with DFU on the system and then replaced before DFULOAD is run, you do not have to copy the diskette file DFU using the TOLIBR procedure.

```
DFULOAD  [A1 ]  [S1 ]
         [A2 ]  [S2 ]
         [A3 ]  [S3 ]
         [A4 ]  [M1.nn]
                [M2.nn]
```

S9020107-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #DFULIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #DFULIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #DFULIB on disk and copy the DFU support from diskette.

```
DFULOAD
```

DFUSAVE Procedure

The DFUSAVE procedure copies the data file utility (DFU) support to diskette. The DFU support from the libraries #DFULIB and #LIBRARY is copied. You should use the "DFULOAD Procedure" on page 4-151 to load the DFU support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPUTIL and be located in diskette slot S1.

```
DFUSAVE
```

S9020108-0

The DFUSAVE procedure has no parameters.

Example

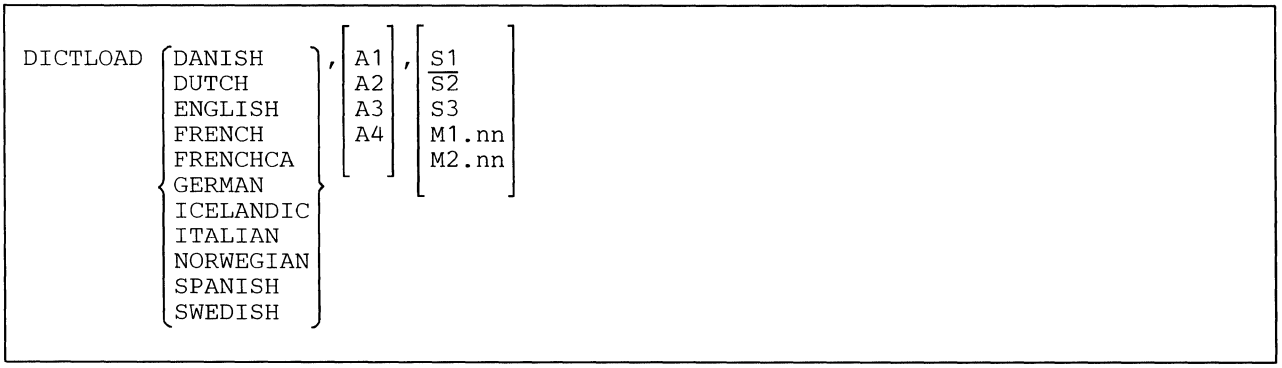
Copy the DFU support to diskette.

```
DFUSAVE
```

DICTLOAD Procedure

The DICTLOAD procedure creates a library for a specified language and copies the dictionary support for DW/36 from diskette into that library. The DICTLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the DICTSAVE procedure. See the "DICTSAVE Procedure" on page 4-155 for information about how to save the dictionary support on diskette.

The DICTLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the DICTLOAD procedure to restore support that has been saved by DICTSAVE.



S9020477-3

DANISH specifies that the Danish dictionary is to be copied. If DANISH is specified, the DICTLOAD procedure creates a library named #TUDAN and copies the support from diskette into that library.

DUTCH specifies that the Dutch dictionary is to be copied. If DUTCH is specified, the DICTLOAD procedure creates a library named #TUDUT and copies the support from diskette into that library.

ENGLISH specifies that the English dictionary is to be copied. If ENGLISH is specified, the DICTLOAD procedure creates a library named #TUENG and copies the support from diskette into that library.

FRENCH specifies that the French dictionary is to be copied. If FRENCH is specified, the DICTLOAD procedure creates a library named #TUFRN and copies the support from diskette into that library.

FRENCHCA specifies that the Canadian French dictionary is to be copied. If FRENCHCA is specified, the DICTLOAD procedure creates a library named #TUFRN and copies the support from diskette into that library.

GERMAN specifies that the German dictionary is to be copied. If GERMAN is specified, the DICTLOAD procedure creates a library named #TUGER and copies the support from diskette into that library.

ICELANDIC specifies that the Icelandic dictionary is to be copied. If ICELANDIC is specified, the DICTLOAD procedure creates a library named #TUICE and copies the support from diskette into that library.

ITALIAN specifies that the Italian dictionary is to be copied. If ITALIAN is specified, the DICTLOAD procedure creates a library named #TUITN and copies the support from diskette into that library.

NORWEGIAN specifies that the Norwegian dictionary is to be copied. If NORWEGIAN is specified, the DICTLOAD procedure creates a library named #TUNOR and copies the support from diskette into that library.

SPANISH specifies that the Spanish dictionary is to be copied. If SPANISH is specified, the DICTLOAD procedure creates a library named #TUSPN and copies the support from diskette into that library.

SWEDISH specifies that the Swedish dictionary is to be copied. If SWEDISH is specified, the DICTLOAD procedure creates a library named #TUSWE and copies the support from diskette into that library.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and the library is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, the library is placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

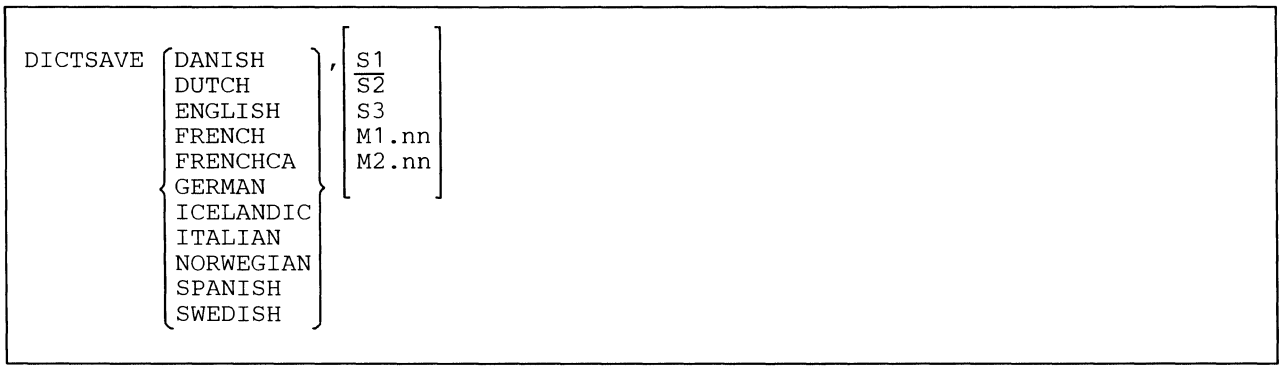
Example

Create #TUENG on disk and copy the English dictionary from diskette.

```
DICTLOAD ENGLISH
```

DICTSAVE Procedure

The DICTSAVE procedure copies the specified dictionary support for DW/36 to diskette. See the "DICTLOAD Procedure" on page 4-153 for the library associated with the specified dictionary, from which the support is copied. You should use the DICTLOAD procedure to load the dictionary support from the backup diskette.



S9020478-3

DANISH specifies that the Danish dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPDAN.

DUTCH specifies that the Dutch dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPDUT.

ENGLISH specifies that the English dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPENG.

FRENCH specifies that the French dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPFRN.

FRENCHCA specifies that the Canadian French dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPFRC.

GERMAN specifies that the German dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPGER.

ICELANDIC specifies that the Icelandic dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPICE.

ITALIAN specifies that the Italian dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPITN.

NORWEGIAN specifies that the Norwegian dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPNOR.

SPANISH specifies that the Spanish dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPSPN.

SWEDISH specifies that the Swedish dictionary is to be saved. The diskette to contain the saved copy must have a volume ID of PPSWD.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

DISABLE Procedure

The DISABLE procedure stops or disables an enabled Interactive Communications Feature (SSP-ICF) subsystem, MSRJE, 3270 device emulation, or PC Support/36. See one of the following manuals for more information:

- For information about the DISABLE procedure, see the manual *Using System/36 Communications.*
- For information about SSP-ICF, see the manuals *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference*, *Interactive Communications Feature: Upline Subsystems Reference*, *Interactive Communications Feature: Finance Subsystem Reference*, and *Interactive Communications Feature: Base Subsystems Reference.*
- For information about MSRJE, see the *MSRJE Guide.*
- For information about 3270 device emulation, see the manual *3270 Guide.*
- For information about asynchronous communications support, see the manual *Using Asynchronous Communications Support.*
- For information about APPN, see the manual *Advanced Peer-to-Peer Networking Guide.*
- For information about PC Support/36, see the manual *PC Support/36 Technical Reference.*

The DISABLE procedure can be run from any display station, but from only one display station at a time.

If there are active communication sessions, DISABLE displays a message with the following options:

0 Hold the disable; no new sessions are to start. The system will continue the disable when the current communication sessions end.

1 Retry the disable and check again for active sessions.

2 Immediately end the active sessions and continue the disable.

3 End the DISABLE procedure and ignore the disable request. The DISABLE procedure is canceled and the subsystem remains enabled.

The DISABLE procedure runs the $IEDS utility program.

```
DISABLE  subsystem configuration name,[location name],[line number]
```

S9020109-1

subsystem configuration name specifies the subsystem configuration to be disabled.

location name specifies the remote location to be disabled. The subsystem remains enabled as long as there are other active locations.

line number specifies the number of the line to be disabled. This parameter is valid only when disabling an APPN subsystem member.

Example

Disable the subsystem configuration SUB1.

```
DISABLE SUB1
```

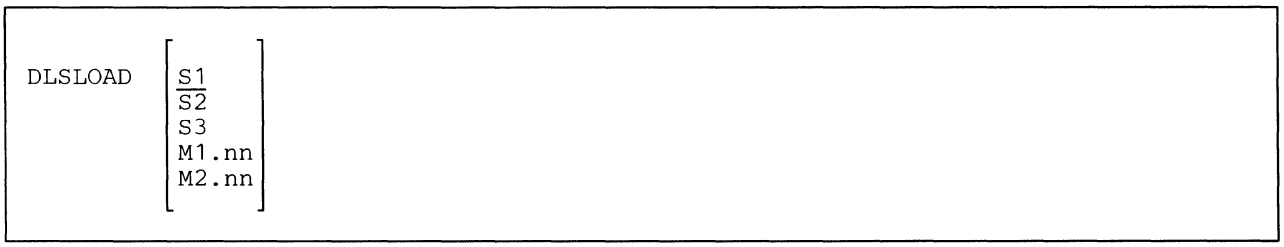
DISPLAY Procedure

The DISPLAY procedure is supported only for compatibility with the IBM System/34. See the "LISTDATA Procedure" on page 4-261 for information on how to display or print a disk or diskette file.

DLSLOAD Procedure

The DLSLOAD procedure reallocates #OFCLIB to 2400 blocks and copies document library services (DLS) support from diskette into the library. DLSLOAD copies additional support into the system library. The DLSLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the DLSSAVE procedure. See the "DLSSAVE Procedure" for more information about how to save the document library services support on diskette.

Note: Document library services require Personal Services/36 to be on the system to load document library services. If Personal Services/36 is not on the system you will be prompted to load Personal Services/36 before the document library services can be loaded.



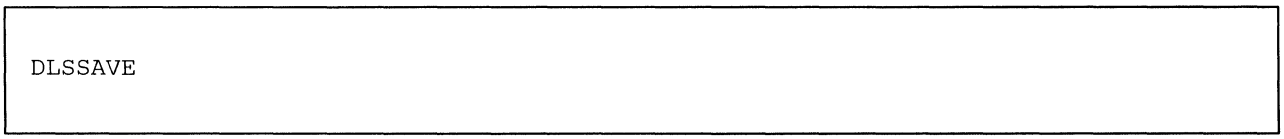
S9020588-0

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

DLSSAVE Procedure

The DLSSAVE procedure copies the document library services (DLS) support from the libraries #OFCLIB and #LIBRARY to diskette. The DLSLOAD procedure should be used to load the document library services support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPOFC.

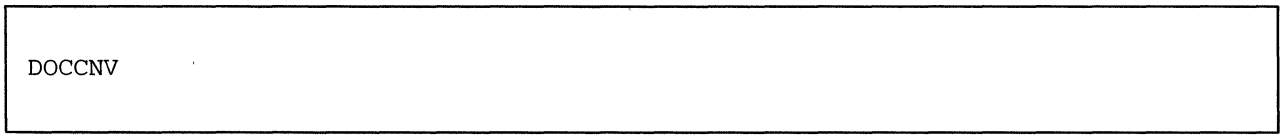


S9020586-0

The DLSSAVE procedure has no parameters.

DOCCNV Procedure

The DOCCNV procedure converts all document folders and mail folders to a new internal format.



S9020597-0

The DOCCNV procedure has no parameters.

DOCPLOAD Procedure

The DOCPLOAD procedure creates a library named #TULIB and copies the print online support from diskette into that library. DOCPLOAD copies additional support into the system library (#LIBRARY). The DOCPLOAD procedure can copy diskettes created by the DOCPSAVE procedure. See the "DOCPSAVE Procedure" on page 4-162 for information about how to save the print online support on diskette.

The DOCPLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the DOCPLOAD procedure to restore support that has been saved by DOCPSAVE.

```
DOCPLOAD [A1 ] , [S1   ]
         [A2 ]   [S2   ]
         [A3 ]   [S3   ]
         [A4 ]   [M1.nn]
                 [M2.nn]
```

S9020604-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #TULIB is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, #TULIB is placed on the least-used disk unit.

A1 specifies that #TULIB is to be placed on the first disk if space is available.

A2 specifies that #TULIB is to be placed on the second disk if space is available.

A3 specifies that #TULIB is to be placed on the third disk if space is available.

A4 specifies that #TULIB is to be placed on the fourth disk if space is available.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #TULIB on disk and copy the print online support from diskette. #TULIB is to be placed on the second disk if space is available.

```
DOCPLOAD A2
```

DOCPSAVE Procedure

The DOCPSAVE procedure copies the print online support to diskette. The print online support from the libraries #TULIB and #LIBRARY is copied. You should use the "DOCPLOAD Procedure" on page 4-161 to load the print online support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPSSP and be located in slot S1.

```
DOCPSAVE
```

S9020605-0

The DOCPSAVE procedure has no parameters.

DSU Procedure

The DSU procedure allows you to create, edit, remove, view, or print library procedure members and library source members using the development support utility (DSU). DSU is a full-screen editor that allows you to edit an entire screen of text or data at a time.

For more information about the DSU procedure, see the *IBM System/36 Development Support Utility Guide.*

To create or edit a procedure or source member using DSU:

```
DSU  [EDIT],[member name],[type],[format member],[statement length],

     [library name],[diagnosed source file],[display size]
```

S9020505-1

To create, edit, view, print, or remove a library member using DSU:

```
DSU  [LIBRARY],[library name],[type],[format member],,,,[display size]
```

S9020506-1

EDIT specifies that an edit session is initialized for the member specified. If the member does not exist, it will be created. If no member name is specified, the *Work With Library Members* display is shown.

LIBRARY takes you to the *Work With Library Members* display which lists the members in the specified or default library. From there you can create, edit, view, print, or remove a library member.

member name specifies the source member or procedure that you want to create or edit. A member name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except blanks. You should avoid using the following characters because these have special meanings in procedures: commas (,), hyphens (-), apostrophes ('), question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), or periods (.). If a name is not specified, a list of the members in the library will be displayed.

library name specifies the library that contains or will contain the library member being changed or created. If no library member is specified, the session library is assumed.

type specifies the type of member to be listed, created or edited. If LIBRARY was specified as the first parameter, P and S are valid entries. Entering a P in this field indicates library procedure members are to be listed. Entering an S in this field indicates library source members are to be listed. If this parameter is not specified, you will see both procedure members and source members.

If EDIT was specified as the first parameter, the following entries are valid:

A specifies an RPG II program that contains auto report specifications.

F specifies a library source member that contains display format specifications.

P specifies a library procedure member.

R specifies a library source member that contains an RPG II program.

S specifies a library source member.

T specifies a message source member.

W specifies a source member containing work station utility statements.

If EDIT is specified as the first parameter and type is not specified, S is assumed.

format member specifies the name of the display format load member that you created to be used by DSU to edit source or procedure statements.

statement length specifies the length for each source or procedure statement. Member types of A or R must be from 80 to 96; T member types must be from 40 to 80; all other member types can be from 40 to 120. If the member is being created, the values you can specify and the values that are assumed by DSU if no statement length is specified are as follows:

Member Type	Allowed Statement Length	Assumed Statement Length
SOURCE, S	40 to 120	96
PROC, P	40 to 120	120
A	80 to 96	96
F	40 to 120	96
R	80 to 96	96
T	40 to 80	80
W	40 to 120	96

diagnosed source file specifies the name of the file that contains the diagnosed source statements and any resulting messages. This parameter is used by the RPGONL, COBOLONL, and FORTONL procedures to allow a diagnosed source file to be built, and then edited. The file must have been created by a compiler, such as the RPG compiler as invoked by RPGONL. If a diagnosed source file is specified, DSU uses that file as input and allows you to make changes. When DSU ends, the file, together with any changes you have made, replaces the source member specified in the first parameter.

display size specifies the screen size. If this parameter is blank, the display size will default to 24x80 for a 5250-type display station or 27x132 for a 3180 display station. If a 1 is specified for this parameter, the display size will be set to 24x80. If a 2 is specified, the display size will be set to 27x132, which is valid only for a 3180 Display Station.

Example 1

This example allows you to create or edit source member ACCTRECV in library ACCTLIB.

```
DSU EDIT,ACCTRECV,S,,,ACCTLIB
```

Example 2

This example lists the names of all procedure members in library ACCTLIB on your display station. You can then choose to edit, view, print, or remove them.

```
DSU LIBRARY,ACCTLIB,P
```

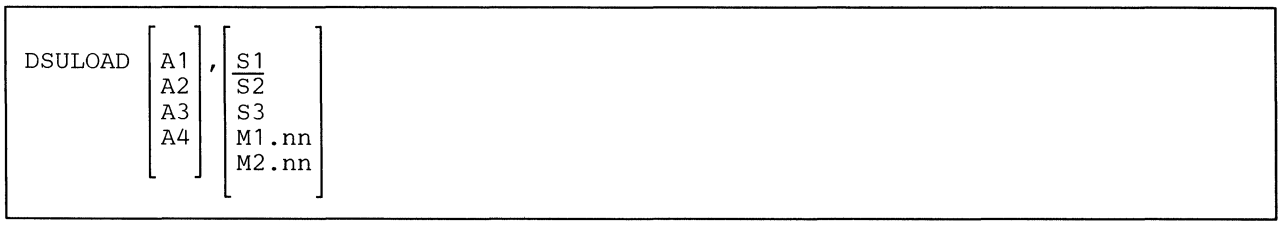
DSULOAD Procedure

The DSULOAD procedure creates a library named #DSULIB and copies the development support utility (DSU) support from diskette into that library. The DSULOAD procedure also creates the libraries #DSULB1 and #DSULB2 if they do not already exist. These two libraries contain the user profiles. DSULOAD copies additional support into the system library (#LIBRARY). The DSULOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the DSUSAVE procedure.

The DSULOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the DSULOAD procedure to restore support that has been saved by DSUSAVE.

If DSU is not currently on the system, you must use the TOLIBR procedure to copy the diskette file DSU into #LIBRARY before running DSULOAD.

If #LIBRARY was backed up with DSU on the system and then replaced before DSULOAD is run, you do not have to copy the diskette file DSU using the TOLIBR procedure.



S9020533-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #DSULIB is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, #DSULIB is placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #DSULIB on disk and copy the DSU support from diskette.

```
DSULOAD
```

DSUSAVE Procedure

The DSUSAVE procedure copies the development support utility (DSU) support to diskette. The DSU support from #LIBRARY and #DSULIB is copied. You should use the DSULOAD procedure to load the DSU support from the backup diskettes. The diskette to contain the saved copy must have a volume ID of PPDSU.

```
DSUSAVE
```

S9020535-0

The DSUSAVE procedure has no parameters.

Example

Copy DSU support from #LIBRARY and #DSULIB to diskette.

```
DSUSAVE
```

EDITNRD Procedure

The EDITNRD procedure creates and edits the network resource directory (#NRD.FLE). When editing the directory, you can add, change, scan, delete, and print the entries that make up the directory.

The EDITNRD procedure is interactive: the system prompts you to enter the appropriate information. For more information about this procedure and about the network resource directory, see the *Distributed Data Management Guide.*

The EDITNRD procedure runs the $SINR utility program.

```
EDITNRD
```

S9020110-0

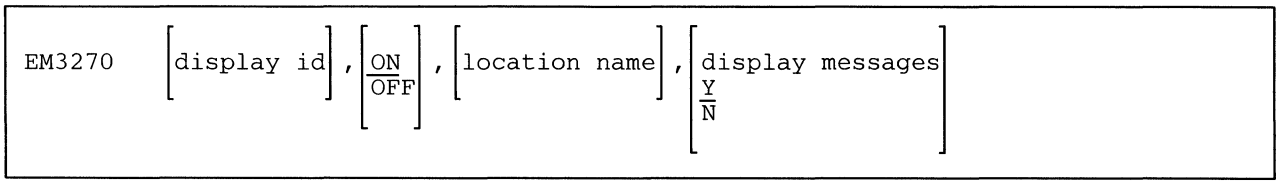
The EDITNRD procedure has no parameters.

EM3270 Procedure

The EM3270 procedure signs a display station or printer on or off the BSC 3270 device emulation subsystem. Before you can enter this procedure, the BSC 3270 subsystem must be enabled.

See the *3270 Guide* for more information about the EM3270 procedure and 3270 device emulation.

For signing on or off a display station:



S9020111-1

If you are signing on your own display station, you need to enter only EM3270.

display id specifies the work station ID of the display station to be signed on or off BSC 3270 device emulation. If the display station being signed on is the one from which the procedure is entered, this parameter is not required. If the display ID is that of another display station, that display station must be a data display station that is not currently in use, or a command display station that is in standby mode.

ON specifies that you want to sign a display station on to BSC 3270 device emulation. If no parameter is entered, ON is assumed.

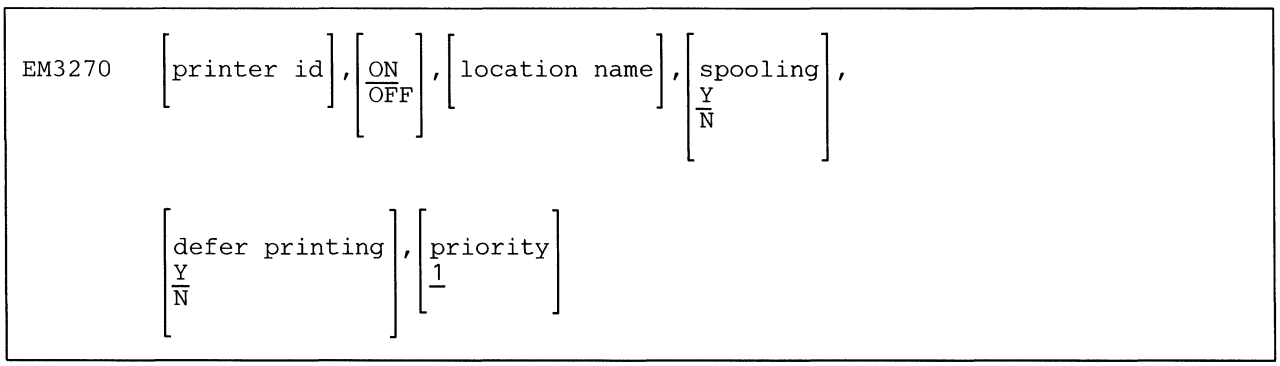
OFF specifies that you want to sign a display station off BSC 3270 device emulation.

location name specifies the location name associated with this session. The location name is defined during subsystem configuration, and refers to the name of the remote location with which communication is to take place.

YES or Y specifies that messages are to be displayed. If no parameter is entered, YES is assumed.

NO or N specifies that messages are not to be displayed.

For signing on or off a printer:



S9020112-1

printer id specifies the work station ID of the printer to be signed on or off. If the printer output for this printer is not being spooled, the printer must not be in use.

ON specifies that you want to sign a printer on to BSC 3270 device emulation. If no parameter is entered, ON is assumed.

OFF specifies that you want to sign a printer off BSC 3270 device emulation.

location name specifies the location name associated with this session. The location name is defined during subsystem configuration, and refers to the remote location with which communication is to take place.

spooling specifies whether output to the printer is to be spooled. Either YES or NO can be specified. If spooling is not active, this parameter is ignored.

YES or Y specifies that output is to be spooled. If no parameter is specified, YES is assumed.

NO or N specifies that output is not to be spooled.

defer printing specifies when spooled output should be printed. Either YES or NO can be entered. If spooling is not active, this parameter is ignored.

YES or Y indicates that the printing of spooled output is to wait until the printer is signed off BSC 3270 device emulation. If no parameter is entered, YES is assumed.

NO or N indicates that printing of the spooled output should not wait until the printer is signed off 3270 device emulation. That is, output should be printed as soon as output can be printed.

priority specifies the priority for spooled output. The priority is a number from 0 through 5, with 5 being highest priority (printed first), 1 being lowest priority, and 0 indicating that the output should be held until a RELEASE command is entered to allow the output to be printed. If this parameter is not specified, 1 is assumed. If spooling is not active, this parameter is ignored.

Example 1

This example shows how to sign your display station on to 3270 device emulation.

```
EM3270
```

Example 2

This example shows how to sign printer P3 on to 3270 device emulation. The printed output is to be spooled, and any output should be printed as soon as possible.

```
EM3270 P3,ON,,,NO
```

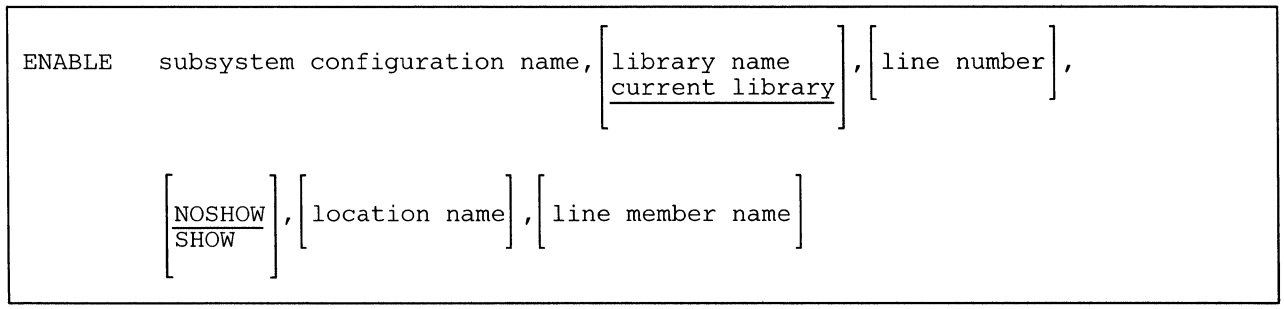
ENABLE Procedure

The ENABLE procedure starts or enables a communications subsystem. The ENABLE procedure must be run before a subsystem can be used. See one of the following manuals for more information:

- For information about the ENABLE procedure, see the manual *Using System/36 Communications.*
- For information about SSP-ICF, see the manuals *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference, Interactive Communications Feature: Upline Subsystems Reference, Interactive Communications Feature: Finance Subsystem Reference,* and *Interactive Communications Feature: Base Subsystems Reference.*
- For information about MSRJE, see the *MSRJE Guide.*
- For information about 3270 device emulation, see the manual *3270 Guide.*
- For information about asynchronous communications support, see the manual *Using Asynchronous Communications Support.*
- For information about APPN, see the *Advanced Peer-to-Peer Networking Guide.*
- For information about PC Support/36, see the manual *PC Support/36 Technical Reference.*

This procedure can be run at any display station, but only one ENABLE procedure can be run at a time.

The ENABLE procedure runs the $IENBL utility program.



S9020113-2

subsystem configuration name specifies the subsystem configuration to be enabled. This is the name that was specified when the CNFIGICF procedure was run to configure the subsystem member.

library name specifies the library containing the subsystem configuration. If no library name is specified, the current library is assumed.

line number specifies the number of the communications line to use for the subsystem. If you are enabling the Intra subsystem, you must omit this parameter.

NOSHOW specifies that the parameters configured for a particular subsystem are not to be displayed. If no parameter is specified, NOSHOW is assumed.

SHOW specifies that all the parameters configured for a particular subsystem are to be displayed. You cannot change any of these parameters. For a description of the parameters that are displayed, see one of the manuals mentioned above.

location name specifies the location to be activated. The location is activated if it is not already enabled.

line member name specifies the name of the line member to be enabled. This parameter is valid only when you enable an APPN subsystem member.

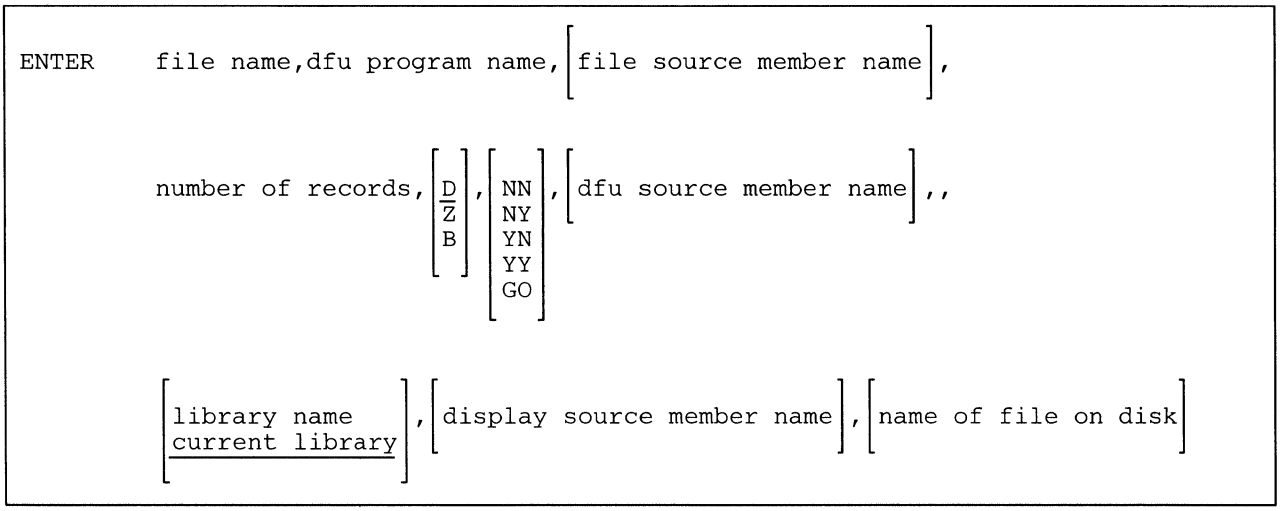
Example

In the following example, the ENABLE procedure enables the subsystem specified by the configuration member SUB1 found in the library named MYLIB. SUB1 is to use communications line 1, and the parameters for this configuration are to be displayed.

```
ENABLE SUB1,MYLIB,1,SHOW
```

ENTER Procedure

The ENTER procedure allows you to create a disk file using the data file utility (DFU). File specifications are used to define the format of the records in the file. For more information about DFU, see the *DFU Guide*.



S9020114-0

file name specifies the file to be created. The file name can be from 1 through 8 alphameric characters.

dfu program name specifies the DFU program to be created to process the file. If the program does not exist in the library, DFU starts the setup procedures. If the program does exist in the library, DFU goes directly to creating the file.

file source member name specifies the source member that contains the file description (F-specification) and record input descriptions (I-specifications) that describe the file to be processed. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are taken as the data description.

This parameter is prompted for if it is not specified. This parameter is required if the specified program does not exist.

number of records specify the maximum number of records you want to enter into the file. This parameter is required. If the parameter is not specified, it is prompted for. DFU allows a maximum of 8000000 records within a file.

D, Z, or B indicates whether unkeyed numeric fields are to be filled with zeros (hexadecimal F0) or blanks. The only allowed entries are **D**, **Z**, or **B**. If no parameter is specified, D is assumed.

D or B specifies a data file with blank fill of unkeyed numeric fields.

Z specifies a data file with zero fill of unkeyed numeric fields.

NN, NY, YN, YY, or GO specifies how the DFU source specifications are to be processed. This parameter is used if the program does not exist and the program setup must be performed. It indicates whether the DFU specifications for this program are already stored in the library, and whether they are to be stored in the library when the program setup is complete. This source member of DFU specifications is stored or looked for in the current library, unless a library name is specified.

NN specifies stored DFU specifications are not used for this program setup. You are prompted to create the specifications but they are not saved in the library.

NY specifies stored DFU specifications are not used for this program setup. You are prompted to create the specifications; once created, they are stored in the library.

YN specifies stored DFU specifications are used for this program setup. You are not prompted, but you can update the retrieved specifications before the program is created. If you change the specifications at the display station, only this program is affected; the changes are not stored in the library.

YY specifies stored DFU specifications are used for this program setup. You are not prompted, but you can update the retrieved specifications before the format description is created. The specifications in the library are replaced by the updated specifications.

GO specifies stored DFU specifications are used for this program setup. You are not prompted and cannot update the specifications before the program is created unless errors are found. If errors are found in the specifications, the operator must correct the specification before the program is created. However, the stored DFU specifications are not changed.

dfu source member name specifies the source member that contains, or will contain, saved DFU specifications. This parameter is required if the DFU source processing parameter is not NN.

library name specifies the library containing or to contain the DFU specifications. All library members associated with the DFU job are looked for, or stored, in this library. If this parameter is not specified, the current library is assumed.

display source member name specifies the source member in which DFU is to store the display format source specifications that were created when a DFU job was set up. If this parameter is specified, you can change the generated display format source specifications at another time, compile them to replace the existing display format load member, and then run the same DFU program with the data displayed in the redefined fields.

name of file on disk specifies the name to be listed in the disk VTOC for the file to be created, if it is different from the name specified in the DFU program. If this parameter is specified, you can have several programs create different, logical files that all refer to the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file to be created by the DFU program, you are prompted for that parameter.

Example

This example shows how to create a disk file named FILE1 using a file source member named RPGPROG. The file is to contain 100 records, and the DFU specifications named DFUSETUP are to be saved after they are created. The library MYLIB contains the file source member and will contain the DFU specifications.

```
ENTER FILE1,DFILFMT,RPGPROG,100,,NY,DFUSETUP,,MYLIB,DFILFMT1
```

EPDOWNL Procedure

The EPDOWNL procedure copies the personal computer portion of the 3278 Emulation via IBM Personal Computer from the System/36 to the IBM Personal Computer.

For more information about the EPDOWNL procedure, see the manual *3278 Emulation via IBM Personal Computer User's Guide*, SC09-1086.

```
EPDOWNL
```

S9020529-0

The EPDOWNL procedure has no parameters.

EPLMRG Procedure

The EPLMRG procedure merges the personal computer machine-readable instruction (MRI) files and translated tables in #LIBRARY into the virtual diskette #EPPCLD1.

For more information about the EPLMRG procedure, see the *3278 Emulation via IBM Personal Computer User's Guide*.

```
EPLMRG
```

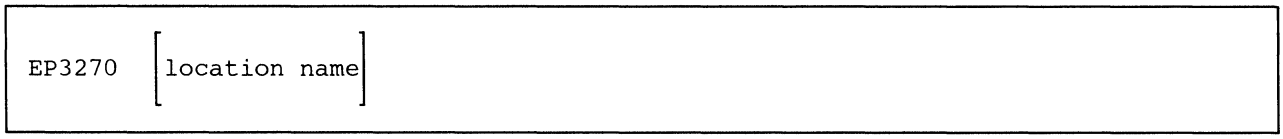
S9020621-0

The EPLMRG procedure has no parameters.

EP3270 Procedure

The EP3270 procedure signs an IBM Personal Computer on to SNA 3270 device emulation. Before you can enter this procedure, the SNA 3270 subsystem must be enabled.

See the *3278 Emulation via IBM Personal Computer User's Guide* for more information about the EP3270 procedure and about 3270 device emulation.



S9020507-0

location name specifies the location associated with this session. The location name is defined during subsystem configuration and refers to the name of the remote location with which communications is to take place. If only one remote location of SNA 3270 device emulation is enabled, this parameter can be omitted.

Example

This example shows how to sign your IBM Personal Computer on to 3270 device emulation.

```
EP3270
```

ERR Procedure

The ERR procedure causes the specified error message to be displayed, and allows an operator to select an option: 0, 1, 2, 3, or D. The message specified is retrieved from the current user level one message member. This allows your procedures to display messages and issue options in the same way that the system displays error messages.

For information on assigning the USER1 message member, see the "MEMBER OCL Statement" on page 5-73.

The ERR procedure runs the $CPPE utility program.

```
ERR      message id,[options ],['text']
                    [3      ]  [     ]
```

S9020115-0

message id code specifies the message identification code of the message to be displayed. The parameter is a 1- to 4-digit number in the range 0 through 9999.

options specifies the options to be displayed with the message. The options can be any combination of 0, 1, 2, or 3; but the numbers must be entered in ascending order and with no blanks or other characters. For example, to cause options 1 and 3 to be displayed, enter 13. If no options are specified, option 3 is assumed.

If option 3 is selected by an operator, the procedure is immediately canceled; that is, control is not returned to the procedure. If option 0, 1, or 2 is selected by an operator, a return code is set. You can use the ?CD? expression in a procedure to check for the option selected by an operator. The value set is as follows:

Option	?CD? Return Code Value
0	1010
1	1011
2	1012

See "Substitution Expressions" on page 3-10 for more information about substitution expressions.

If option D is selected by an operator, the procedure is immediately canceled and a task dump is taken.

text specifies the characters that will be used as replacement text in the displayed message. The characters will be used to sequentially replace each pound symbol character (#) used as a place holder in the stored message. Up to 75 characters can be specified, and they must be enclosed in apostrophes ('). Replacement text fields are discussed under "Message Member Statements" on page 4-133.

Example

The following example shows a procedure that will display an error message from the MESSAGES message member if the operator did not enter SOURCE or PROC for the second parameter. That is, parameter 2 does not equal SOURCE or PROC. The message displayed is number 0020 from the level one message member named MESSAGES. The options that will be allowed are 0, 1, and 3.

- If option 0 is entered, parameter 2 (P2) will contain SOURCE.
- If option 1 is entered, parameter 2 (P2) will contain PROC.
- If option 3 is entered, the procedure is automatically canceled.

```
// MEMBER USER1-MESSAGES
// IFF ?2?=SOURCE IFF ?2?=PROC   ERR 0020,013
// IF ?CD?=1010 EVALUATE P2='SOURCE'
// IF ?CD?=1011 EVALUATE P2='PROC'
LISTLIBR ?1?,?2?,MYLIB
```

If the operator did not enter a second parameter, message 0020 could be displayed as follows:

```
USER-0020 (01 3)
SOURCE or PROC must be entered.  0=SOURCE, 1=PROC, 3=Cancel
```

The source statements contained in the MESSAGES member are:

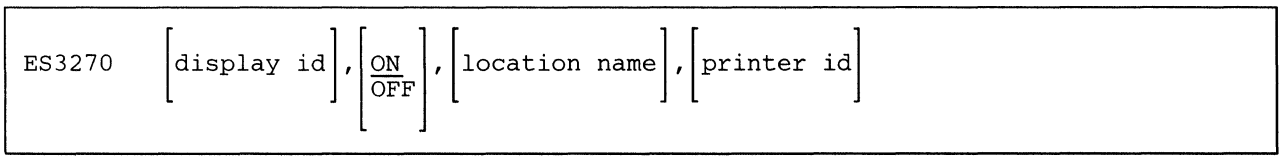
```
MESSAGES,1
0020 SOURCE or PROC must be entered.  0=SOURCE, 1=PROC, 3=Cancel
```

ES3270 Procedure

The ES3270 procedure signs a display station or printer on or off the SNA 3270 device emulation subsystem. Before you can enter this procedure, the SNA 3270 subsystem must be enabled.

See the *3270 Guide* for more information about the ES3270 procedure and about 3270 device emulation.

To sign on or off a display station:



S9020116-0

display id specifies the work station ID of the display station to be signed on or off SNA 3270 device emulation. If the display station being signed on is the one from which the procedure is entered, this parameter is not required. If the display ID is that of another display station, that display station must be a data display station that is not currently in use, or a command display station that is in standby mode.

ON specifies that you want to sign a display station on to SNA 3270 device emulation. If no parameter is entered, ON is assumed.

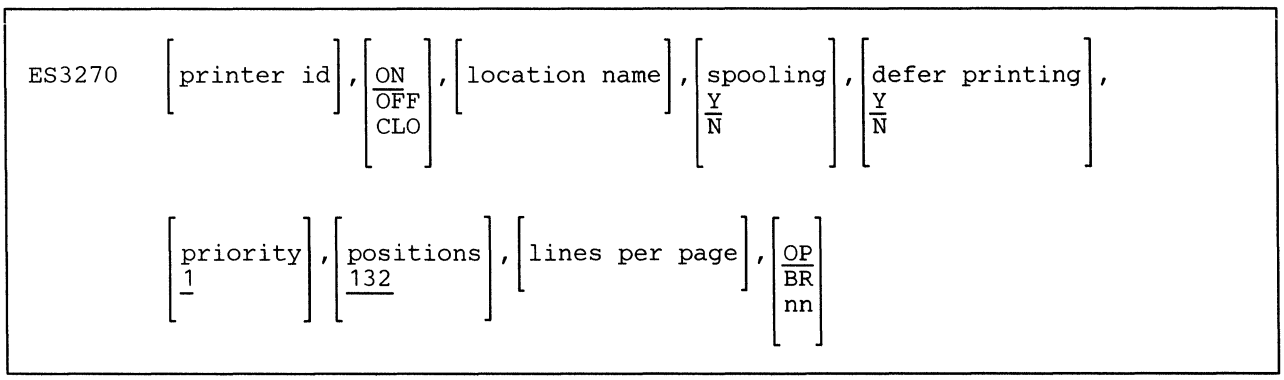
OFF specifies that you want to sign a display station off SNA 3270 device emulation.

location name specifies the location name associated with this session.

The location name is defined during subsystem configuration and refers to the name of the remote location with which communication is to take place. If only one SNA 3270 device emulation subsystem is active, you need not specify this parameter.

printer id specifies the work station ID of a printer that is to be used for a host-initiated copy function. This parameter corresponds to the print authorization matrix used by the 3274. If no parameter is entered, the session printer associated with the display station is assumed.

To sign on or off a printer or close a spool file:



S9020117-2

printer id specifies the work station ID of the printer to be signed on or off SNA 3270 device emulation. If the printer output for this printer is not being spooled, the printer must not be in use.

ON specifies that you want to sign a printer on to SNA 3270 device emulation. If no parameter is entered, ON is assumed.

OFF specifies that you want to sign a printer off SNA 3270 device emulation.

| **CLO** specifies that the current spooled file is to be closed and another one opened in order to continue with the SNA 3270 printer emulation operation.

location name specifies the location name associated with this session. The location name is defined during subsystem configuration and refers to the name of the location with which communication is to take place. If only one SNA 3270 device emulation subsystem is active, you need not specify this parameter.

spooling specifies whether output to the printer is to be spooled. Either YES (Y) or NO (N) can be specified. If spooling is not active, this parameter is ignored. YES specifies that output is to be spooled; NO specifies that output is not to be spooled. If no parameter is specified, YES is assumed.

defer printing specifies when spooled output should be printed. Either YES (Y) or NO (N) can be entered. If spooling is not active, this parameter is ignored. YES indicates that the printing of spooled output is to wait until the printer is signed off SNA 3270. NO indicates that printing of the spooled output should not wait until the printer is signed off SNA 3270. That is, output should be printed as soon as output can be printed. If no parameter is entered, YES is assumed.

priority specifies the priority for spooled output. The priority is a number from 0 through 5, with 5 being highest priority (printed first), 1 being lowest priority, and 0 indicating that the output should be held until a RELEASE command is entered to allow the output to be printed. If this parameter is not specified, 1 is assumed. If spooling is not active, this parameter is ignored.

positions specifies the number of print positions in a line. Any decimal number from 1 through 132 can be specified. If no parameter is specified, 132 is assumed. This parameter applies only when you are signing on a printer that will use SNA character strings (a feature of the 3287 printer).

lines per page specifies the number of lines per page. Any decimal number from 1 through 255 can be specified. If no parameter is entered, the current number of lines per page is assumed. This parameter applies only when you are signing on a printer that will use SNA character strings (a feature of the 3287 printer).

NO specifies that the spool file close will be controlled by the operator.

BR specifies that spooled files are created for each group of data received in brackets from the host.

nn specifies an inactive time value in minutes. Values can be from 01 to 99. When the host system has not sent data for the specified time, the spooled file is closed. If a spooled file is closed, another spooled file is opened with the same inactive conditions.

Example 1

This example shows how to sign your display station on to 3270 device emulation.

```
ES3270
```

Example 2

This example shows how to sign printer P3 on to 3270 device emulation. The printed output is to be spooled, and any output should be printed as soon as possible.

```
ES3270 P3,ON,,,NO
```

EXTRACT Procedure

The EXTRACT procedure is supported only for compatibility with the IBM System/34. (System/36 does not allow you to specify hexadecimal data in a procedure, however, so the seventh parameter in the EXTRACT procedure cannot contain hexadecimal data.) See the "COPYDATA Procedure" on page 4-111 for information about copying data files. See the "LISTDATA Procedure" on page 4-261 for information about listing data files.

FORMAT Procedure

The FORMAT procedure processes source statements you have created called **display format** specifications. Display format specifications define how information is to be displayed at a display station. You can use the source entry utility (SEU) or the screen design aid (SDA) to create the display format specifications. When you use SDA, you need not use the FORMAT procedure.

For information about display format specifications and for sample listings of the FORMAT procedure, see the manual *Creating Displays*.

The FORMAT procedure does the following:

- Creates a new display format load member containing the formats defined by the source specifications. All display formats used by a program can be placed in one or more display format load members. Up to 255 display formats can be placed in one member.
- Adds one or more display formats to an existing display format load member.
- Replaces one or more display formats in an existing display format load member.
- Deletes a display format from an existing display format load member.

When you generate a display format load member, the reference number assigned to the source member is assigned to the generated load member. This allows you to determine whether an old level of a load member is being used by comparing the two reference numbers.

The FORMAT procedure runs the $SFGR utility program.

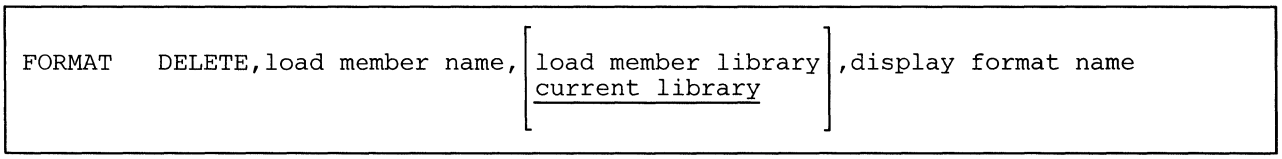
To create, add to, or update a display format load member:

```
FORMAT   [CREATE ]                                [load member library  ]
         [ADD    ],load member name,              [current library      ],source member name,
         [UPDATE ]

         [source member library ],[number of formats],[REPLACE],[HALT  ],
         [current library       ] [1                 ]         [NOHALT]

         [PRINT  ]
         [NOPRINT]
         [PARTIAL]
```

S9020118-0

FORMAT

To delete a format from a display format load member:



S9020119-0

CREATE specifies that a new display format load member is to be created. If no parameter is specified, CREATE is assumed.

ADD specifies that one or more display formats are to be added to an existing display format load member. Only the display format(s) that are to be added are contained in the source member.

UPDATE specifies that one or more display formats are to be replaced in an existing display format load member. Only the display format(s) that are to be updated are contained in the source member.

DELETE specifies that a single display format is to be removed from an existing display format load member. If the removed format was the only one in the member, the entire display format load member is removed from the library.

load member name specifies the display format load member to be created, added to, updated, or deleted from.

load member library specifies the name of the library that is to contain (or does contain) the display format load member. If a library name is not specified, the current library is assumed.

source member name specifies the name of the library source member that contains the display format specifications.

source member library specifies the name of the library that contains the source member. If a library name is not specified, the current library is assumed.

number of formats specifies the number of display formats contained in the display format source member. This number is used to determine the amount of temporary work file space needed by the FORMAT procedure. The larger the number entered, the more temporary disk file space will be used. If the number of display formats is not specified, 1 is assumed. The number of formats needs only to be specified if CREATE or ADD is specified.

REPLACE specifies that if a load member already exists with the same name as the display format load member being created, the existing load member is to be replaced by the FORMAT procedure. REPLACE is only used if CREATE is specified; if UPDATE or ADD is specified, REPLACE is ignored.

If REPLACE is not specified and a duplicate load member exists, a message is displayed. The operator can then either replace the existing load member with the created display format load member or end the job.

HALT specifies that a message is to be displayed that indicates that warning and terminating errors were encountered during the processing of the display format source member. If a message is displayed indicating a warning, the operator can either continue the job or cancel the job. If a message is displayed indicating a terminating error, the operator can only cancel the job. If a parameter is not entered, HALT is assumed.

NOHALT specifies that no message is to be displayed that indicates that warning or terminating errors were encountered during the processing of a display format source member. For warning errors, the job step is completed and the display formats are generated. For terminating errors, the job step is ended and no display formats are generated.

Note: For terminating errors, the ?CD? substitution expression is set to 1008; see "Substitution Expressions" on page 3-10 for more information.

PRINT specifies that the following is to be printed:

- The display format source member name
- The display format S- and D-specifications
- Any informational, warning, or terminating error messages
- The input and output field descriptions
- A list of the display format indicators used
- The input and output library names
- The display format load member name
- The storage requirements for each format
- The data stream length for each format
- The number of bytes of help text storage

If a parameter is not entered, PRINT is assumed.

NOPRINT specifies that when a terminating error is encountered in the source specifications, only the statement in error and the terminating error message are to be printed. If no errors are found, nothing is printed.

PARTIAL specifies that the following is to be printed.

- The display format source member name
- Any warning or terminating messages together with the statement causing the message, or any informational messages
- The input and output library names
- The display format load member name

display format name specifies the name of the display format to be deleted.

Example 1

Create a new display format load member named FORMAT2 in the library named MYLIB. The display format specifications define three display formats and are in a source member named INPUT. The source member is in the library named MYLIB.

```
FORMAT CREATE,FORMAT2,MYLIB,INPUT,MYLIB,3
```

Example 2

Update an existing display format load member named FORMAT2 in the library MYLIB. The source member containing the display format specifications is named SCREEN and is also contained in the library MYLIB.

```
FORMAT UPDATE,FORMAT2,MYLIB,SCREEN,MYLIB
```

Example 3

The following example shows how to remove a display format named SCREEN1 from the display format load member named FORMAT2. The load member is in the library named MYLIB.

```
FORMAT DELETE,FORMAT2,MYLIB,SCREEN1
```

Example 4

Add to an existing display format load member named FORMAT2 in the library MYLIB. The source member containing the display format specifications is named SCREEN and is also contained in the library MYLIB.

```
FORMAT ADD,FORMAT2,MYLIB,SCREEN,MYLIB
```

FORTC Procedure

This procedure is supported only for compatibility with the IBM System/34. The procedure you should use to compile a FORTRAN program on the System/36 is the "FORTRANC Procedure" on page 4-188.

FORTCG Procedure

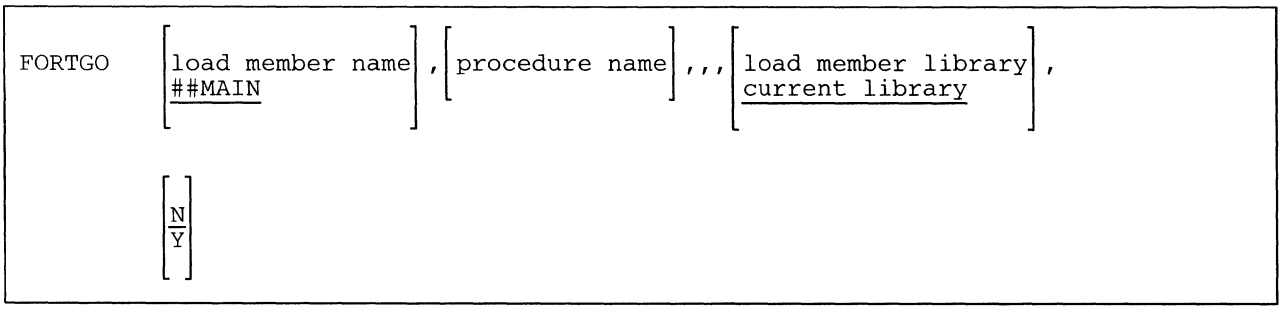
This procedure is supported only for compatibility with the IBM System/34. The procedure you should use to compile a FORTRAN program on the System/36 is the "FORTRANC Procedure" on page 4-188. The procedure you should use to run a FORTRAN program on the System/36 is the "FORTGO Procedure" on page 4-185.

FORTG Procedure

This procedure is supported only for compatibility with the IBM System/34. The procedure you should use to run a FORTRAN program on the System/36 is the "FORTGO Procedure" on page 4-185.

FORTGO Procedure

The FORTGO procedure runs a FORTRAN program. For information on FORTRAN, see the manual *Programming with FORTRAN IV.*



S9020120-0

load member name specifies the library load member that contains the FORTRAN program to be run.

procedure name specifies the library procedure member that contains one or more OCL statements for the program. This procedure can contain any OCL statements that can be placed between a LOAD and RUN OCL statement pair.

library name specifies the library that contains the FORTRAN load member and procedure member. If this parameter is not specified, the current library and then the system library (#LIBRARY) is searched.

N specifies that the job is not to be placed on the job queue. If no parameter is entered, N is assumed.

Y specifies that the job is to be placed on the job queue.

Example

This example shows how to run a FORTRAN program that is named PAYROLL and is contained in the current library. The procedure member named PAYPROC contains the OCL statements used by the program.

```
FORTGO PAYROLL,PAYPROC
```

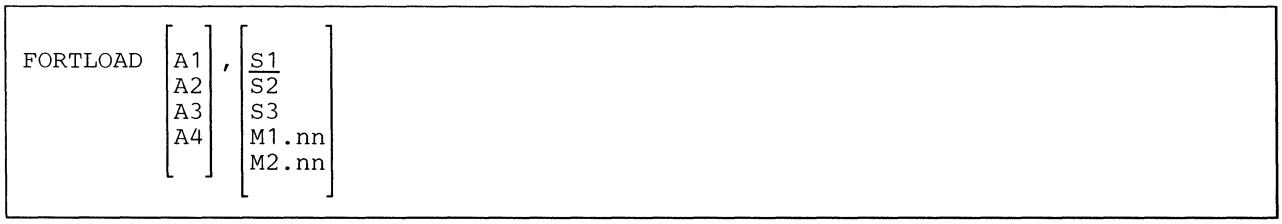
FORTLOAD Procedure

The FORTLOAD procedure creates a library named #FORTLIB and copies the FORTRAN support from diskette into that library. FORTLOAD copies additional support into the system library (#LIBRARY). The FORTLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the FORTSAVE procedure. See the "FORTSAVE Procedure" on page 4-191 for information about how to save the FORTRAN support on diskette.

The FORTLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the FORTLOAD procedure to restore support that has been saved by FORTSAVE.

If FORTRAN is not currently on the system, you must use the TOLIBR procedure to copy the diskette file FORTRAN into #LIBRARY before running FORTLOAD.

If #LIBRARY was backed up with FORTRAN on the system and then replaced before FORTLOAD is run, you do not have to copy the diskette file FORTRAN using the TOLIBR procedure.



S9020121-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on the unit, the other units (if they exist) are checked, and #FORTLIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #FORTLIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #FORTLIB on disk and copy the FORTRAN support from diskette.

```
FORTLOAD
```

FORTONL Procedure

The FORTONL procedure allows you to develop a FORTRAN program. You can create a new program, or make many changes to an existing program. The FORTONL procedure causes a series of displays to appear, which allow you to enter, compile, and change FORTRAN programs.

For more information about FORTRAN, see the manual *Programming with FORTRAN IV.*

```
FORTONL
```

S9020122-0

The FORTONL procedure has no parameters.

Example

To develop a FORTRAN program, enter:

```
FORTONL
```

FORTP Procedure

The FORTP procedure causes a menu to be displayed that allows you to select the FORTRAN task you want to perform. You can enter, compile, run, or change FORTRAN programs.

```
FORTP
```

S9020123-0

The FORTP procedure has no parameters.

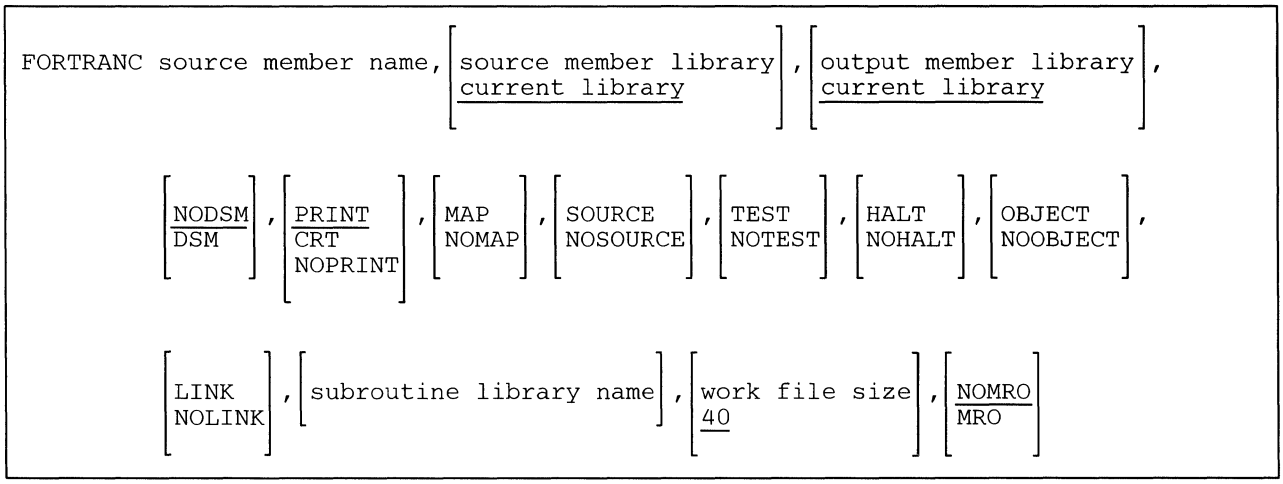
Example

To display the FORTP menu, enter:

```
FORTP
```

FORTRANC Procedure

The FORTRANC procedure compiles a FORTRAN program. For information on FORTRAN, see the manual *Programming with FORTRAN IV.*



S9020124-1

source member name specifies the library source member to be compiled.

source member library specifies the name of the library that contains the source member to be compiled. If this parameter is not specified, the current library is assumed.

output member library specifies the name of the library that contains or is to contain the compiled subroutine or load member. If this parameter is not specified, the current library is assumed. Note that if a library name is specified on the *PROCESS statement in the source program, that library will be used.

NODSM specifies that no diagnosed source member is to be produced. If no parameter is specified, NODSM is assumed.

DSM specifies that the FORTRANC procedure should produce a diagnosed source member of the FORTRAN program. The $MAINT utility program is used to replace the original source member with the diagnosed source member. Statements that have errors are preceded by a statement with two question marks (??) in positions 1 and 2 in the source member. See the manual *Programming with FORTRAN IV* for more information about the diagnosed source member.

PRINT specifies that the compiler listing is to be printed. If no parameter is specified, PRINT is assumed.

NOPRINT specifies that no output is to be printed.

CRT specifies that the compiler listing is to be displayed at the display station that is running the FORTRANC procedure.

MAP or NOMAP specifies whether the variables and statements in the FORTRAN program are to be mapped. If neither MAP or NOMAP is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used. MAP specifies a map of the variables and statements is to be produced. NOMAP specifies that no map of the variables and statements is to be produced.

SOURCE or NOSOURCE specifies whether the FORTRAN source statements are to be included in the printed listing produced by the compiler. If neither SOURCE or NOSOURCE is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used. SOURCE specifies the FORTRAN statements are to be included in the listing. NOSOURCE specifies that no FORTRAN statements are to be included.

TEST or NOTEST specifies whether any DEBUG statements in the program are to be processed. If neither TEST or NOTEST is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used. TEST specifies that any DEBUG statements are to be compiled. NOTEST specifies that no DEBUG statements are to be compiled.

HALT or NOHALT specifies whether the message that is displayed if errors are found in the program should cause the compilation to stop or halt. If neither HALT or NOHALT is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used. HALT specifies the FORTRANC procedure is to display the message and halt if any errors are found in the program being compiled. NOHALT specifies the FORTRANC procedure is to display the message but not halt.

OBJECT or NOOBJECT specifies whether the FORTRANC procedure should produce a subroutine member from the compiled program. If no parameter is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used.

OBJECT specifies that an object module is to be created. If the source member is a FORTRAN main program, the object module must be link-edited using the overlay linkage editor (the OLINK procedure) before it can be run. This allows you to make changes to subroutines the main program may call without having to recompile the main program; that is, you will only need to compile the subroutines and link-edit the main program.

If the source member is a FORTRAN subroutine, this is the only type of module that can be created. This module can be link-edited with a main program by either compiling the main program (and specifying LINK) or by using the overlay linkage editor.

The object module is cataloged as a library subroutine member. If a subroutine member exists in the specified output library, the existing member is replaced with the newly compiled program.

NOOBJECT specifies that an object module is not to be created.

LINK or NOLINK specifies whether the FORTRANC procedure should run the overlay linkage editor as part of the compile. If no parameter is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used. This parameter is valid only for FORTRAN main programs.

LINK specifies that the overlay linkage editor should be run. The compiled object module is link-edited to produce a load member that can be run. If a load member exists in the specified output library, it is replaced with the newly compiled and link-edited program.

NOLINK specifies that the overlay linkage editor is not to be run.

subroutine library name specifies the library that contains one or more subroutines to be link-edited with the program being compiled. If no parameter is specified, the parameter specified in the FORTRAN program's *PROCESS statement is used.

work file size specifies the size of the FORTRANC work file in blocks. Any number from 1 through 9999 can be entered. If no parameter is specified, 40 blocks are assumed.

MRO or NOMRO specifies whether the compiler is to use memory resident overlays. If no parameter is specified, NOMRO is assumed.

MRO specifies that the compiler is to use memory resident overlays.

NOMRO specifies that the compiler is not to use memory resident overlays.

Example

This example shows how to compile a FORTRAN program named PAYROLL. The example shows the following:

- The program is in the current library.
- A diagnosed source member is to be produced.
- No compiler listing is to be produced (the diagnosed source member will be used to correct any errors).
- No variable or statement map is to be produced.
- Any DEBUG statements are to be compiled.
- The FORTRANC procedure is to continue if any errors are found.
- No subroutine or load members are to be created.

```
FORTRANC PAYROLL,,,DSM,NOPRINT,NOMAP,,TEST,NOHALT,NOOBJECT,NOLINK
```

FORTSAVE Procedure

The FORTSAVE procedure copies the FORTRAN IV support to diskette. The FORTRAN IV support from the libraries #FORTLIB and #LIBRARY is copied. You should use the "FORTLOAD Procedure" on page 4-186 to load the FORTRAN IV support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPFORT and be located in diskette slot S1.

```
FORTSAVE
```

S9020125-0

The FORTSAVE procedure has no parameters.

Example

Copy the FORTRAN IV support to diskette.

```
FORTSAVE
```

FORTSDA Procedure

The FORTSDA procedure starts the screen design aid (SDA) procedure. See the manual *Creating Displays* for more information about how to use SDA and for information about display formats.

See the manual *Programming with FORTRAN IV* for more information about this procedure and about FORTRAN.

```
FORTSDA
```

S9020126-0

The FORTSDA procedure has no parameters.

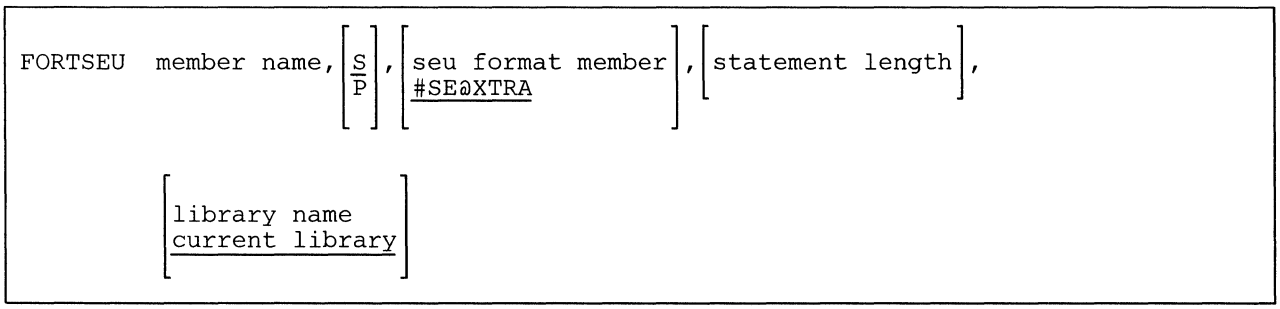
Example

To start the FORTSDA procedure, enter:

```
FORTSDA
```

FORTSEU Procedure

The FORTSEU procedure allows you to create or change a FORTRAN program using the source entry utility (SEU). For information on FORTRAN, see the manual *Programming with FORTRAN IV.* For more information on SEU, see the *SEU Guide.*



S9020127-0

member name specifies the name of the library member to be created or changed.

S specifies a FORTRAN source member. If no parameter is specified, S is assumed.

P specifies a procedure member.

format member name specifies the name of the load member that contains SEU formats. If no parameter is specified, #SE@XTRA is assumed.

statement length specifies the length for each source or procedure statement. This can be any number from 40 to 120. If the member exists, the statement length of the member is assumed. If the member is being created, the values you can specify and the values that are assumed by SEU if no statement length is specified are as follows:

Member Type	Allowed Statement Length	Assumed Statement Length
S	80 to 96	96
P	40 to 120	120

library name specifies the name of the library that is to contain or contains the member being created or changed. If no library name is specified, the current library is assumed.

Example

This example shows how to use the FORTSEU procedure to change a FORTRAN source member named PAYROLL. The current library contains the source member.

```
FORTSEU PAYROLL
```

FROMLIBR Procedure

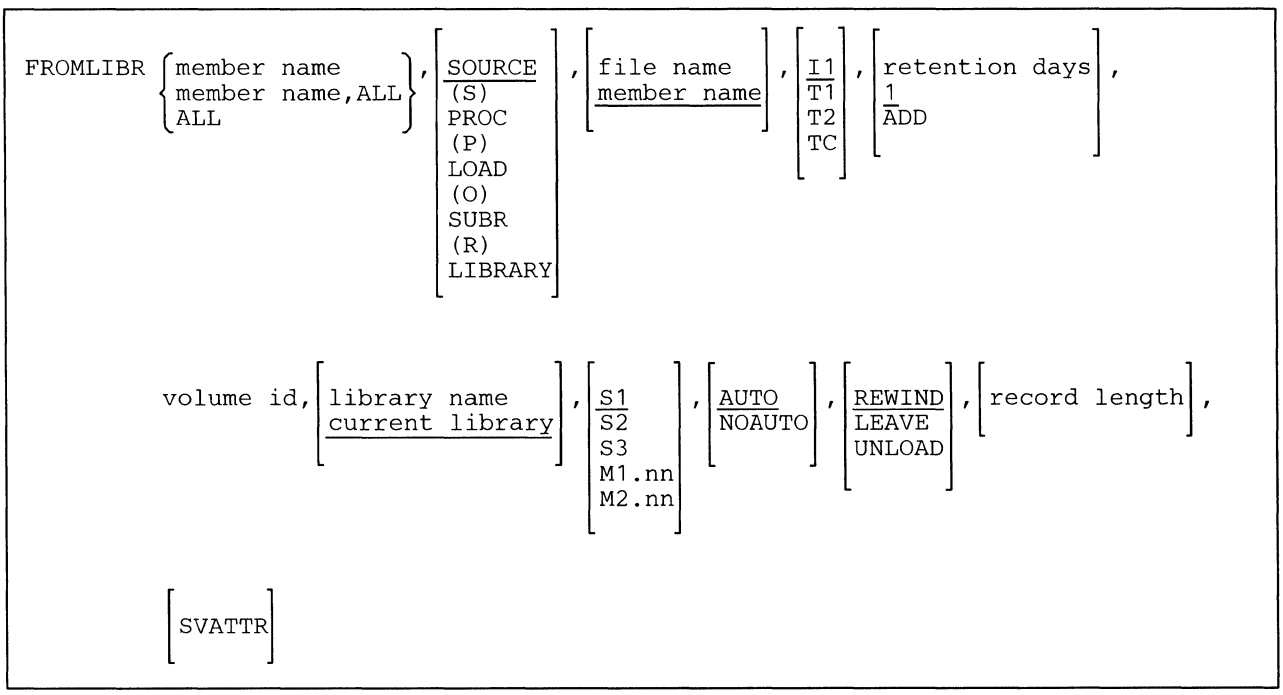
The FROMLIBR procedure copies one or more library members to a disk, diskette, tape, or tape cartridge file. These files can be copied back to the library by using the TOLIBR procedure.

The FROMLIBR procedure cannot copy IBM-supplied library members. These members are the IBM-supplied procedures, source members, and load members that come with the SSP and other licensed programs (such as RPG or BASIC). To copy IBM-supplied library members, see "Copy Members from a Library (FROMLIBR Procedure)" on page A-66.

To save an entire library on diskette, tape, or tape cartridge (even the libraries that contain IBM-supplied members), see the "SAVELIBR Procedure" on page 4-431.

The FROMLIBR procedure runs the $MAINT utility program.

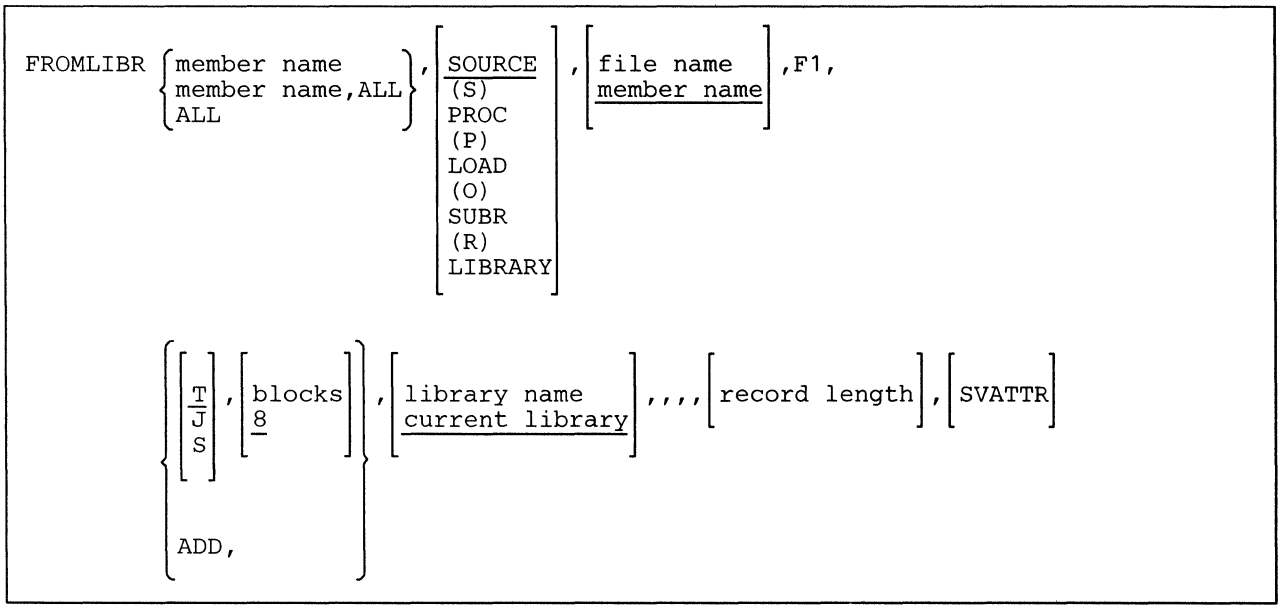
For copying or adding one or more library members to diskette, tape, or tape cartridge:



S9020128-1

FROMLIBR

For copying or adding one or more library members to a sequential **disk** file:



S9020129-0

member name specifies the library member to be copied from the library.

member name,ALL specifies that one or more library members with names beginning with **member name** are to be copied. The name can be up to 7 characters long. For example: PAY,ALL specifies all library members having names that begin with PAY are to be copied, such as: PAYROLL, PAYCHECK, or PAYRUN.

ALL specifies that all library members are to be copied from the library.

SOURCE or S specifies that only source members are to be copied. If a parameter is not specified, SOURCE is assumed.

PROC or P specifies that only procedure members are to be copied.

LOAD or O specifies that only load members are to be copied.

SUBR or R specifies that only subroutine members are to be copied.

LIBRARY specifies that all types of library members (source, procedure, load, and subroutine) are to be copied. If the ***record length*** parameter is specified, only source and procedure members are copied.

file name specifies the file to be created. A file name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, @, or $). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. Do not use ALL as a file name. If a file name is not specified, the name specified for the member name or partial member name is assumed. If ALL is specified but no file name is specified, a message is displayed and the operator must enter the name of the file to be created.

I1 specifies that one or more library members are to be placed in a new diskette file or added to an existing diskette file. If no parameter is specified, I1 is assumed.

F1 specifies that one or more library members are to be placed in a new disk file or added to an existing disk file.

T1 specifies that the tape to be used for this procedure is on tape drive 1.

T2 specifies that the tape to be used for this procedure is on tape drive 2.

TC specifies that a tape cartridge is to be used for this procedure.

retention days specifies how long the diskette, tape, or tape cartridge file is to be retained, and can be from 0 (zero) through 999 days. If I1, T1, T2, or TC is specified and retention days or ADD is not specified, 1 day is assumed. If a retention period of 999 days is specified, the diskette, tape, or tape cartridge file is a permanent file. For more information on diskette, tape, and tape cartridge file retention, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

ADD specifies that one or more library members are to be added to an existing file that contains library members.

Note: When members are being added to a disk file, the file must contain enough unused space to hold the members being added. When members are being added to a diskette or tape, the file must be the last active file on the diskette, tape, or tape cartridge.

volume id specifies the volume ID of the diskette, tape reel, or tape cartridge. From 1 through 6 alphameric characters can be specified. If the volume ID is not specified, a message is displayed and the operator must enter the volume ID.

T specifies that the disk file containing the library members is to be a resident file. If no parameter is entered, T is assumed and a new disk file is created. (The file will remain on disk after the FROMLIBR procedure ends.)

J specifies that the disk file containing the library members is to be a job file. The file does not exist after the job containing the FROMLIBR procedure ends.

S specifies that the disk file containing the library members is to be a scratch file. The file does not exist after the FROMLIBR procedure ends.

Note: For more information on disk file retention (the S, J, and T parameters), see the "FILE OCL Statement (for Disk Files)" on page 5-32.

blocks specifies the size of the new disk file to create. One disk block contains 2560 bytes; one byte contains one character. The blocks value is ignored if ADD is specified. If blocks is not specified, a size of 8 blocks is assumed.

library name specifies the library containing the one or more library members to be copied. If a library name is not specified, the current library is assumed.

S1, S2, or S3 specifies the diskette slot containing the first diskette from which members are to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette from which members are to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the FROMLIBR procedure has run. REWIND is assumed if this parameter is not specified. This parameter is not allowed if F1 or I1 is specified.

LEAVE specifies that the tape should be left where it is after the FROMLIBR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is not allowed if F1 or I1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the FROMLIBR procedure has run. This parameter is not allowed if F1 or I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

record length specifies the record length, in bytes, of the source or procedure members being copied. Load and subroutine members will not be copied. The record length can be from 40 through 120 bytes. This parameter should be specified only if you want to create a record mode file.

The record length parameter can be specified only with a member type of SOURCE (S), PROC (P), or LIBRARY. The members are saved in record mode format. Any members added to a record mode file will assume the record length that was specified for the file. Any other record length specified with the ADD parameter will be ignored. Likewise, a record length must not be specified when adding to a file that has no record length specified. Members with a subtype of TXT (text) will not be copied. If you attempt to copy a TXT member, a message will be issued allowing you to change the subtype to UNS (unspecified) in order to copy the member.

If no record length is specified, the members are copied in sector mode format.

SVATTR specifies that the library member attributes or indicators (SSP, MRT, PDATA, DATE, TIME, REF, SUB, and HIST) should be saved when the member is copied in record mode format. (SSP indicates that the library member is part of the SSP. MRT indicates that the library member is a multiple requester terminal procedure. PDATA indicates that the library procedure member passes parameters or program data when the procedure command is processed. DATE indicates the date the member was created or last changed. TIME indicates the time the member was created or last changed. REF indicates the reference number of the member. SUB indicates the subtype of the member. HIST indicates that the library procedure member OCL statements are logged to the history file.)

SVATTR is valid only if a record length is also specified. You must specify a record length of at least 73 bytes if SVATTR is specified, or an error message will be issued.

This parameter is not necessary when you copy members in sector mode format. The attributes are always saved in sector mode.

Example 1

This example shows how to save a library procedure member named TEST on a diskette having a volume ID of VOL003. The procedure is in the library named MYLIB and the diskette is in magazine 2, position 5.

```
FROMLIBR TEST,PROC,,I1,,VOL003,MYLIB,M2.05,NOAUTO
```

Example 2

Assume that all payroll application source programs are in a library named MYLIB and begin with the characters PAY. All the library members are to be copied to a permanent diskette file. The file's label will be PAY, and the diskette's volume ID is VOL001. The diskette is in slot 1.

```
FROMLIBR PAY,ALL,,,I1,999,VOL001,MYLIB,S1,AUTO
```

Example 3

This example shows how to copy a library source member named SAMPLE to a new disk file named FILE1. The source member is in the library named MYLIB, and the disk file is to be 4 blocks in size.

```
FROMLIBR SAMPLE,,FILE1,F1,,4,MYLIB
```

Example 4

This example shows how to add a library source member named SAMPLE1 to an existing disk file named FILE1. The source member is in the library named MYLIB.

```
FROMLIBR SAMPLE1,,FILE1,F1,ADD,,MYLIB
```

Example 5

This example shows how to save a library procedure member named TEST on a tape with a volume ID of VOL001. The procedure is in the library named MYLIB and the tape reel is mounted on tape drive 1. After saving the procedure on tape, unload the tape.

```
FROMLIBR TEST,PROC,,T1,,VOL001,MYLIB,,,UNLOAD
```

Example 6

This example shows how to copy all source and procedure members in record mode format with a record length of 80, saving the member attributes.

```
FROMLIBR ALL,LIBRARY,FILE1,I1,999,VOL001,MYLIB,,,,80,SVATTR
```

HELP Procedure

The HELP procedure allows you to:

- Display menus from which you can select options to do a desired task.
- Enter and run procedures. The procedures you can run using help are:
  - SSP procedures (such as LISTLIBR or CATALOG).
  - Procedures that are part of the Utilities Program Product. This includes the data file utility (DFU), screen design aid (SDA), source entry utility (SEU), and work station utility (WSU).
  - Procedures that run the language program products. This includes the Assembler, BASIC, COBOL, FORTRAN, and RPG.
  - Data communications procedures (such as MSRJE or DEFINEPN).
  - Service aid procedures (such as APAR or DUMP).
- Enter control commands (such as STATUS or CHANGE).
- Display information for many System/36 functions. HELP can be used to display quick-reference information about:
  - How to use the help support
  - OCL statements
  - Procedure control expressions

The main help menu is displayed when a new operator signs onto the system (without having specified a menu on the sign on display). You can also start the help support by entering the HELP procedure at the keyboard or by pressing command key 5 on any menu display or a command display. You can also enter the name of the procedure and press the Help key. This causes a prompt display to be shown that allows you to fill in the parameters.

The help support is made up of menus that are organized by the task you want to perform. For example, if you want to create a new library, you would select an option to work with libraries and a series of menus related to libraries would appear. You can then progress through the menus until the procedure or command that does the function is displayed.

When the HELP procedure is used to specify parameters for a system procedure, and a user procedure with the same name exists in the current user library, the system procedure help display will be shown, but the user procedure will be run. This results in the system procedure parameters being used by the user procedure. Unless the parameters match, the user procedure fails to run properly.

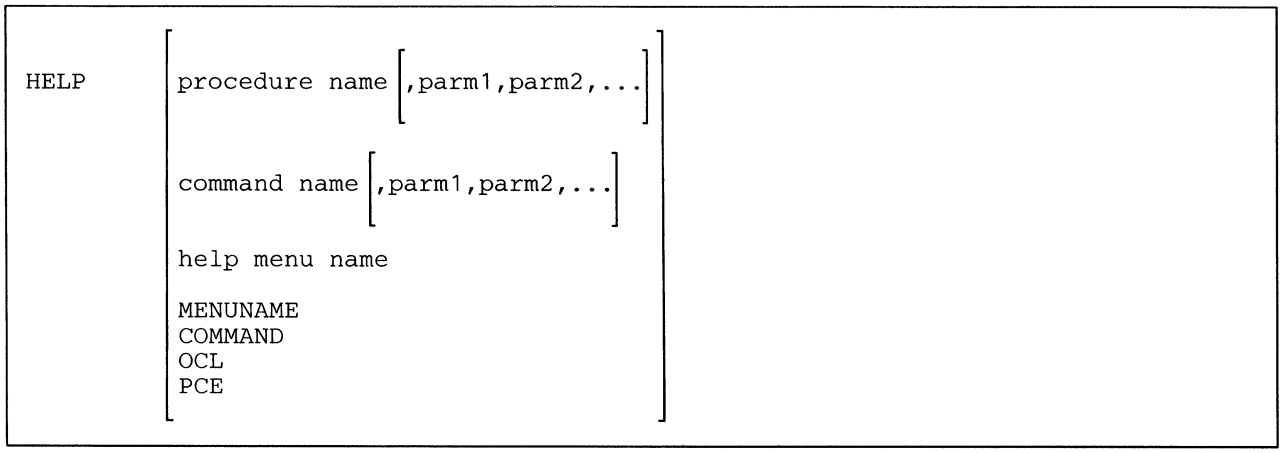
During help, the following function keys and command keys can be used. The term **command key** means that you press the Cmd key and then one of the keys in the top row of the keyboard. For example, to press command key 3, you first press the Cmd key and then the 3 key in the top row; to press command key 15, you first press the Cmd key and then hold down the Shift key while you press the 3 key in the top row. Templates are available to help you identify the command keys on the IBM display stations. See "About This Manual" for more information about the templates.

Key	Function
Help	Displays a help menu or prompt display when you have typed the name of a help menu, procedure, or control command. The Help key also displays additional information or help text about the display or menu you are currently viewing.
Command Key 2	For procedure prompts that require more than one display, command key 2 allows you to reshow the previous procedure prompt display (for example, to change a parameter you entered incorrectly).
Command Key 3	Can be used at any time to display the previous help menu or user menu. You can press command key 3 and back up through at most 20 menus or displays (maximum of 10 help menus; maximum of 10 user menus).
Command Key 4	Can be used only when a procedure prompt is being displayed. Pressing command key 4 causes the procedure to be placed on the job queue, instead of being run from the display station. The job queue function must be active.
Command Key 5	Displays the main help menu (named MAIN).
Command Key 6	Displays your beginning help menu. You can use this to display a help menu you use often, such as the RPG programming menu. To define a menu to be displayed when you press command key 6, see the description of command key 24.
Command Key 7	Can be used at any time to end help and return to the display you were viewing when you started help. When a procedure or control command prompt is shown, command key 7 reshows the display you were viewing before the prompt display was shown. This could be a help menu, your menu, a blank command display, or the console display (at the system console or at a subconsole). When a help menu is shown, command key 7 reshows the display you were viewing before the help menu was shown. This could be your menu, a blank command display, or the console display. When the help tutorial is shown, command key 7 reshows the menu you were viewing before the tutorial was started. This could be a help menu, your menu, a blank command display, or the console display.

Key	Function
Command Key 11	Can be used when a Help menu is displayed to cause the menu name or command names to appear to the right of the display.
Command Key 12	Can be used to start the help tutorial when a menu is being displayed. The tutorial explains how you can use the system help support.
Command Key 14	Can be used on a procedure prompt display to cause a second display of parameters to be prompted for.
Command Key 23	Can be used to change the default user menu that is displayed when you sign on to the system. If you did not enter a menu name as part of the sign on process, this menu will be displayed. The names of your menu and your current library are saved in the user identification file. Command key 23 can only be used when a user menu (not a help menu) is being displayed. The user identification file must be defined for this key to work. For information about how to define the user identification file, see the *System Security Guide*.
Command Key 24	Can be used to change the help menu that is displayed when you start help. This menu then becomes your beginning help menu. This menu is displayed when you press command key 6, or when you enter the HELP procedure with no parameters. The names of the help menu and your current library are saved in the user identification file. Command key 24 can only be used when a help menu is being displayed. The user identification file must be defined for this key to work. For information about how to define the user identification file, see the *System Security Guide*.
Home	Displays the user menu you entered when you signed on to the system, or (if you did not enter a menu name) your default sign-on menu. To define a sign-on menu, see the description of command key 23.
Roll Up (Roll ↑)	Shows the next help display when a series of help displays are used to explain a topic (for example, the help text that explains the parameters for a procedure or command).
Roll Down (Roll ↓)	Shows the previous help display in a series of help displays.

HELP

The HELP procedure runs the $HELP utility program.



S9020130-0

If no parameters are specified, your beginning help menu is displayed. If you have no beginning help menu, the main help menu (named MAIN) is displayed.

procedure name specifies the procedure whose parameters are to be prompted for. The HELP procedure does not support the library parameter specified on the INCLUDE OCL statement. Refer to the "INCLUDE OCL Statement" on page 5-63 for information on the library parameter for a procedure command.

command name specifies the control command to be processed.

parm1,parm2,... represent parameters for the procedure or control command, which help will place on the prompt displays. Either one or more blanks, or a comma, can be entered between the procedure or control command name and the first parameter. For example, you could enter either of the following and the effect is the same:

```
HELP BLDLIBR MYLIB
```

or:

```
HELP BLDLIBR,MYLIB
```

You could also have typed:

```
BLDLIBR MYLIB
```

and pressed the Help key.

All these cause the BLDLIBR procedure prompt display to be shown. The first parameter on the display will be MYLIB.

help menu name specifies one of the system help menus. If no name is specified, your beginning help menu is displayed. If no beginning help menu is defined, the main system help menu (named MAIN) is displayed. You can use the HELP MENUNAME procedure to display the system help menu names.

MENUNAME causes a list of the system help menu names to be displayed.

COMMAND causes a list of the IBM-supplied procedures and control commands to be displayed.

OCL specifies that a list of System/36 OCL statements is to be displayed. From that list, you can select a statement for which reference information will be displayed.

PCE specifies that a list of System/36 procedure control expressions is to be displayed. From that list, you can select the expressions for which reference information will be displayed.

Example 1

You want to create a new library, but you do not know the name of the procedure that creates a library. When you enter HELP, the following display is shown.

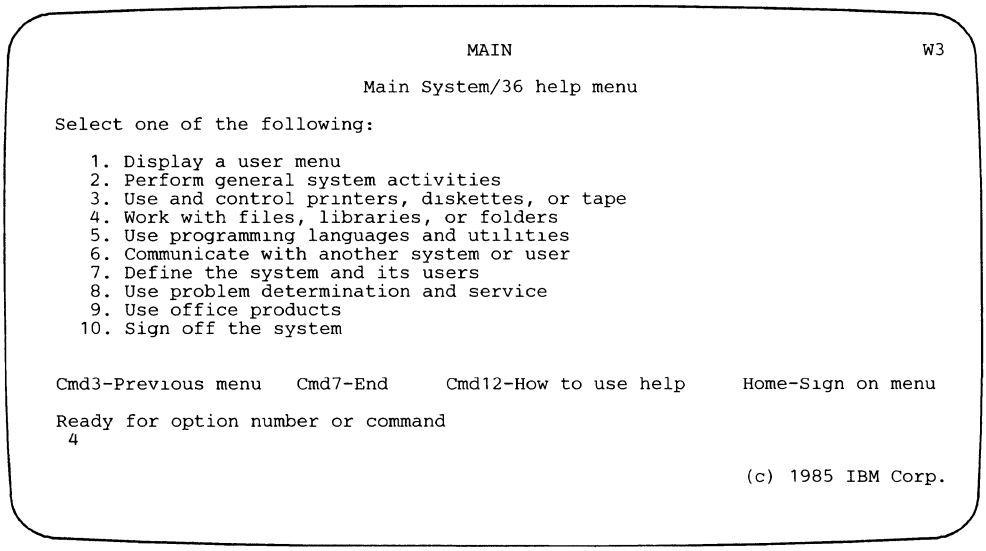


Figure 4-5. Main Help Menu Display

Because you want to create a library, you select option 4:

```
4. Work with files, libraries, or folders
```

After you type the 4 and press the Enter key, the following display is shown:

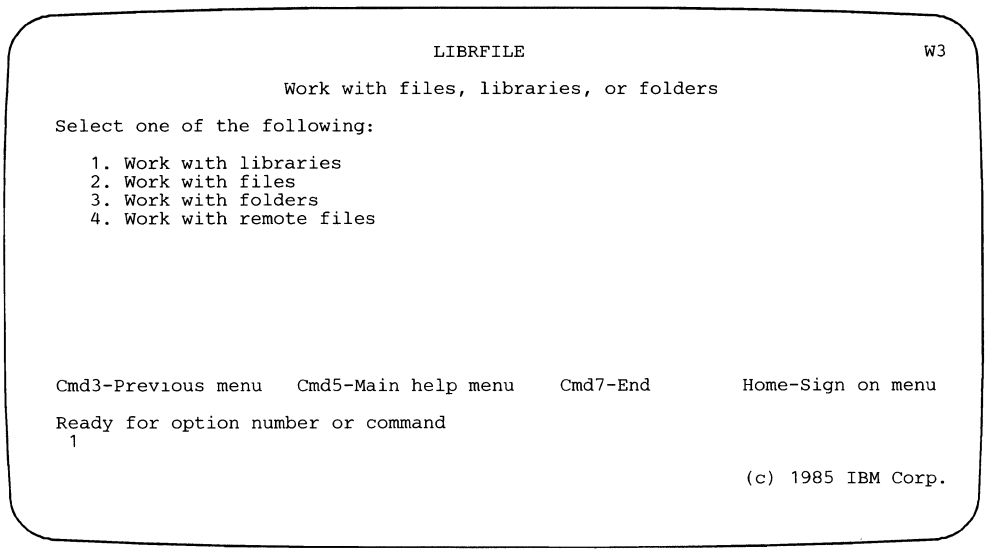


Figure 4-6. Library or Data File Maintenance Menu

Here you would take option 1 because you want to create a library. After you enter the 1, the following display is shown.

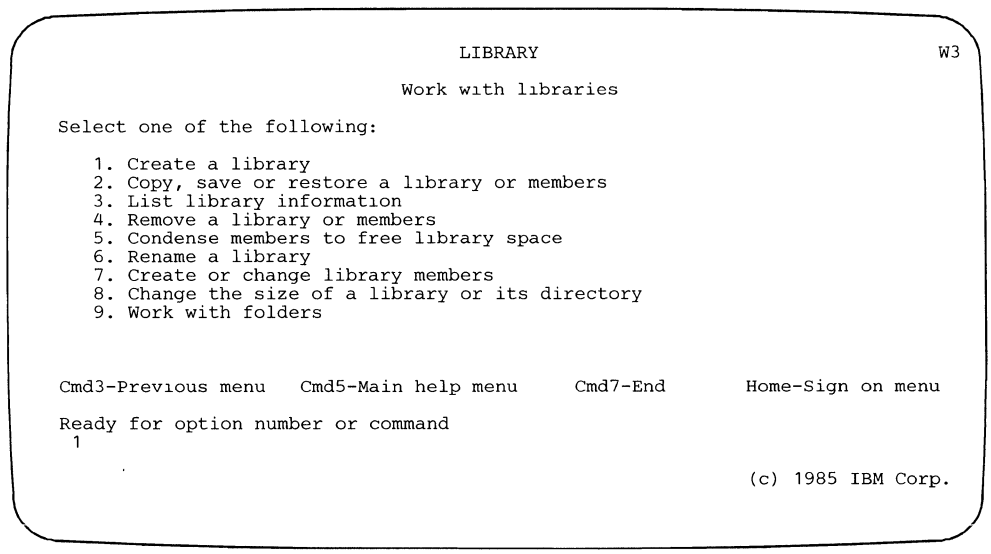


Figure 4-7. Work with Libraries Menu

Here you would take option 1 because you want to create a library. After you enter the 1, the following display is shown:

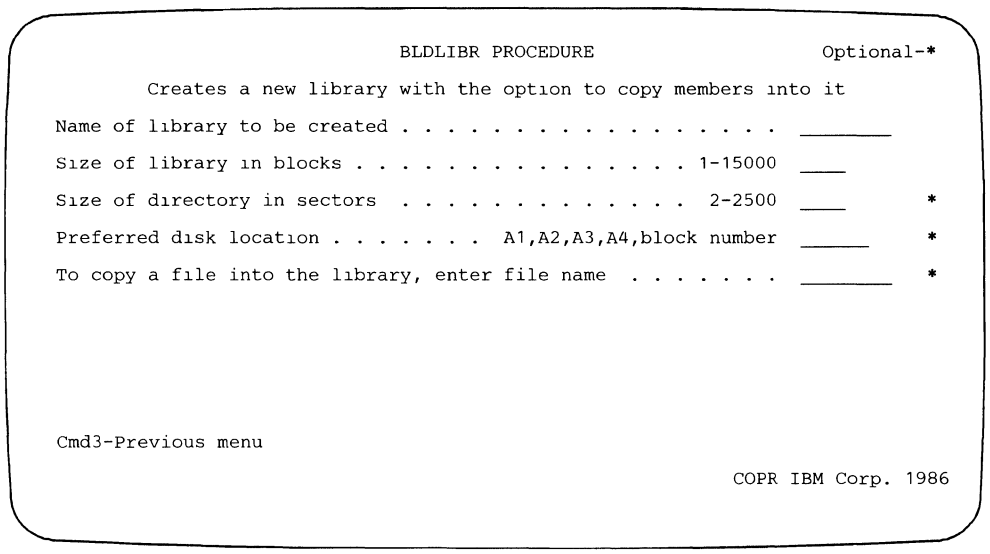


Figure 4-8. BLDLIBR Procedure Display

Here you would enter the name of the library and the size of the library. You could also enter the size of the directory, the preferred disk location, and a file name. These fields are optional fields that are indicated by an asterisk (*).

Example 2

Suppose that you want to create a library and you know that the BLDLIBR procedure creates libraries, but you do not know what parameters the BLDLIBR procedure requires. You could enter the following to cause the BLDLIBR procedure prompt display to appear, and then you could fill in the parameters:

```
HELP BLDLIBR
```

You could also enter the following:

```
BLDLIBR
```

and press the Help key.

Example 3

Suppose that you want to create a library named MYLIB and you know the following:

- The BLDLIBR procedure creates libraries.
- The first parameter of the BLDLIBR procedure is the name of the library.

But you do not know the other parameters.

You can enter the following to cause the BLDLIBR prompt display to appear with the library name filled in, and then you could fill in the other parameters:

```
HELP BLDLIBR MYLIB
```

or:

```
HELP BLDLIBR,MYLIB
```

You could also enter:

```
BLDLIBR MYLIB
```

and press the Help key.

HISTCRT Procedure

The HISTCRT procedure is supported only for compatibility with the IBM System/34. See the "HISTORY Procedure" on page 4-209 for information on how to display history file entries.

HISTCOPY Procedure

The HISTCOPY procedure is run after the history file has been automatically copied to a disk file named HISTCOPY. When the SSP is first installed, the HISTCOPY procedure contains only comments. You can edit or change the HISTCOPY procedure to run your own job after the history file has been copied. A sample procedure is shown on the next page. No parameters can be passed to the procedure. See the "HISTORY Procedure" on page 4-209 for more information about the automatic history file copy process.

To make your own procedures, see Chapter 2, "Making Your Own Procedures." You must place the HISTCOPY procedure in the system library (#LIBRARY) if you want the procedure to run automatically when the history file is filled.

The following figure shows the relationship between the automatic history file copy process and the procedure HISTCOPY:

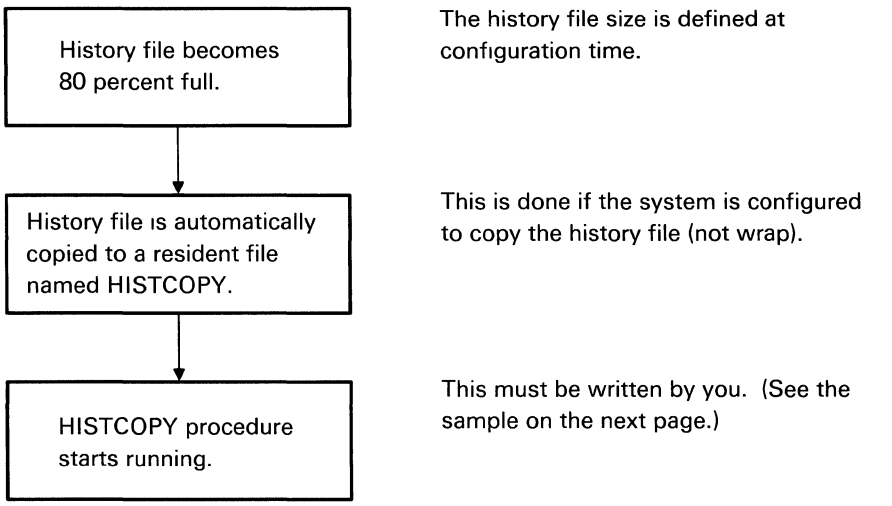


Figure 4-9. Relationship between HISTCOPY and History File

Considerations for Password Security

When password security is active, the user ID of the operator that caused the history file to become 80% full is used by the system to check the security classification of jobs run by the HISTCOPY procedure.

For example, you have a procedure contained in the HISTCOPY procedure that can only be run by a system operator. If a job run by a display station operator causes the HISTCOPY procedure to start, an error message will be sent (to the system console) when the security checking is done for your procedure. This is because the display station operator has a lower security classification than a system operator.

Therefore, when password security is active, you should not include procedures in the HISTCOPY procedure that are restricted to certain operators.

Example

This example shows procedures you could include in the HISTCOPY procedure, which you would then store in the system library (#LIBRARY).

```
* HISTCOPY PROCEDURE
* When your system is configured to periodically save the
* History File (see Changing Your System Configuration),
* two actions occur each time the History file becomes
* about 80% full:
* 1) The System History file is copied to a user file named "HISTCOPY".
* 2) Then control is passed to this HISTCOPY PROCEDURE.
*
* The purpose of the sample HISTCOPY procedure (given below)
* is to periodically rename the current "HISTCOPY" file
* to a new file name.  The new file names are HIST.01, HIST.02,
* up to HIST.99.
*
* Parameter 1 is used as a counter.
// EVALUATE P1=0
*
* Check the file names HIST.nn
*
// TAG LOOP
// EVALUATE P1,2=?1?+1
// IF DATAF1-HIST.?1? GOTO LOOP
*
* File HIST.nn does not exist, rename HISTCOPY to HIST.nn
*
RENAME HISTCOPY,HIST.?1?
*
* Send a message to the system operator
*
// MSG ,File HISTCOPY was renamed to file HIST.?1? on ?DATE? at ?TIME?.
```

HISTORY Procedure

The HISTORY procedure allows you to:

- Display or print all or part of the history file or files created by the HISTCOPY procedure
- Copy the history file to a disk file for later use
- Erase the history file

The following information is recorded in the history file:

- All procedures, control commands, OCL statements, and utility control statements processed by the SSP. OCL statements and utility control statements will be logged only when you specify LOG ON. See the "LOG Procedure" on page 4-294 or the "LOG OCL Statement" on page 5-72 for information about logging.
- If you specify LOG ON, the menu options taken from the user and the help menus are also recorded in the history file.
- All messages displayed at the display stations. Messages that are displayed at the system console or subconsole are given a 4-digit ID and logged to the history file (the 4-digit ID and the message are logged). When the operator responds to the message, which may be some time later, the 4-digit ID, the reply, and the message 'REPLY command successful' are logged. By using this 4-digit ID, you can match the messages to their responses. The 4-digit IDs are used only for the history file, they are not displayed with the message.
- All operator responses to messages.
- End of job and end of print job entries.
- The user ID of the operator associated with the entry. If the entry comes from a job on the job queue, no user ID is associated with the entry.
- The display ID or printer ID associated with the entry. If the entry comes from a job on the job queue, no display station ID is associated with the entry.
- The name of the job associated with the entry. If the entry is a control command, no job name is associated with the entry.
- The time when the entry was placed in the history file. (The time is based upon the time specified by the system operator during IPL.)
- The date (system date) when the entry was placed in the history file. (The date is based upon the date specified by the system operator during IPL.)

Because the history file is limited in size (the size is specified during system configuration), the number of events that can be reflected in the history file at a particular time varies with the length of the file and the size of the entries in the file. Also, the history file is constantly changing. Therefore, a request to list or view the entries may not result in all of the requested entries. If this should occur, run the procedure again.

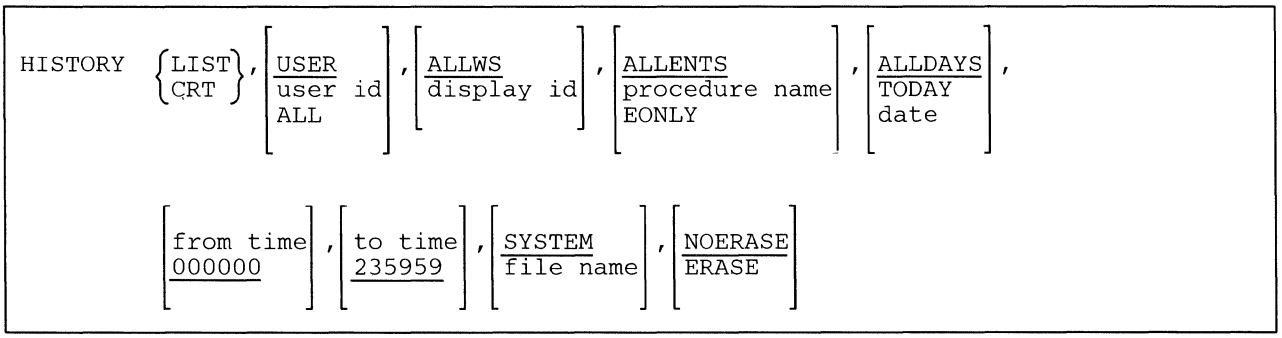
HISTORY

To avoid losing history file entries after the history file has been filled, you can request that when the history file is approximately 80 percent full, it is copied to a disk file named HISTCOPY. You specify this option during system configuration; see the manual *Changing Your System Configuration* for more information. This can cause a job to be run after the copy is performed; see the "HISTCOPY Procedure" on page 4-207 for more information.

The HISTORY procedure runs the $HIST utility program.

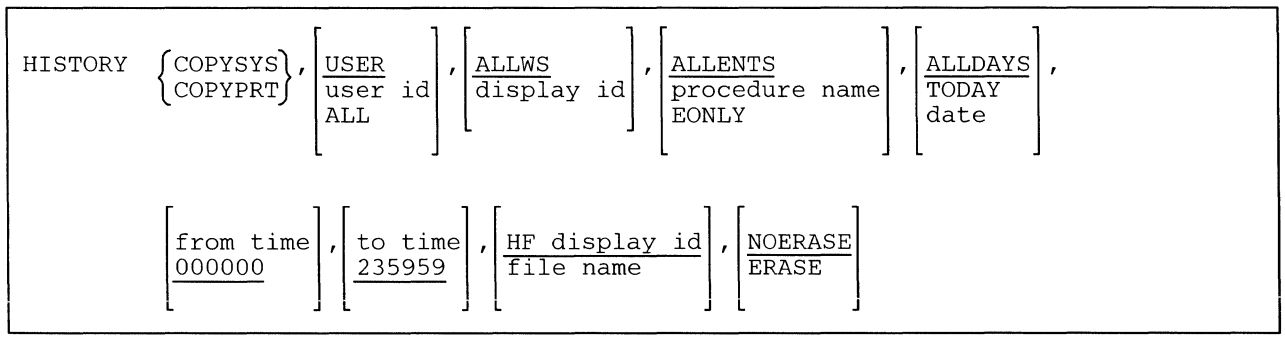
Note: If you try to print with ideographic characters at a nonideographic printer, and print spooling is not active, the ideographic characters are not printed. If print spooling is active, a message is displayed and you have the option of continuing without printing the ideographic characters, or holding the spool file entry, or canceling the spool file entry.

To print or display history file entries:



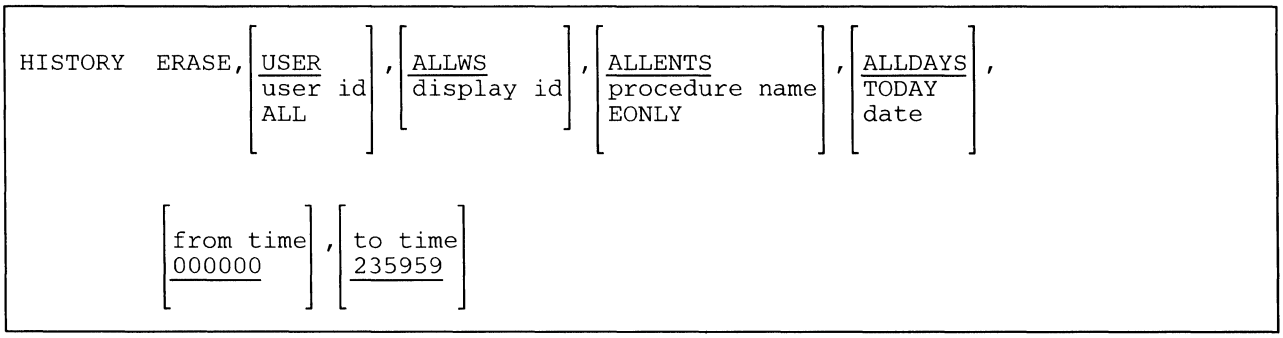
S9020131-0

To copy history file entries to a disk file:



S9020132-0

To erase history file entries:



S9020133-0

If you enter the HISTORY procedure with no parameters, a prompt display is shown to allow you to enter the parameters.

LIST specifies that history file entries are to be listed on the system list device. This can be either a printer or your display station. To determine your system list device, enter the STATUS SESSION command. To change the system list device, see the "SYSLIST Procedure" on page 4-498 or the "PRINT Procedure" on page 4-350.

See "Sample HISTORY Listing" on page 4-218 for a description of the listing that is printed.

CRT specifies that history file entries are to be displayed. You can:

- Page backward and forward through your history file entries for your display station, or (if your security level allows) you can page through all entries in the history file.
- Search the history file for entries containing specific data, such as a character string, a work station ID, a user ID, a job name, a start or stop time, and a date.
- Change the amount of control information displayed.
- Change the information used to select what entries will be displayed.
- Display (if your security level allows) all history file entries as they are added to the history file.

*Note: When you use the CRT parameter, entries longer than 76 characters (such as *EP entries) will be truncated.*

If password security is active, you must have a security classification of system operator or higher to be able to view the entries for another operator.

See "Sample HISTORY Display" on page 4-214 for a description of the displays shown and for information on how to use this option.

COPYSYS specifies that history file entries are to be copied to a new disk file. The format of the disk file is in the system format, which is the same format as the system history file. This allows you to use the copied file at a later time to print or display the entries, or to save the entries on diskette.

COPYPRT specifies that history file entries are to be copied to a new disk file. The format of the disk file is in the same format as if the file had been printed and copied (by the COPYPRT procedure) to a disk file. See the "COPYPRT Procedure" on page 4-126 for the format of the records in this file. This allows you to use the file as input to another program. The file can also be used as input to the COPYPRT procedure to display or print the entries.

ERASE (in the first parameter position) specifies the history file entries are to be erased.

USER specifies that the history file entries that match the operator's user ID are to be processed. If no parameter is specified, USER is assumed.

user id specifies an operator's user ID. Only those entries that match the specified user ID are to be processed.

If password security is active, the operator running the HISTORY procedure must be classified as system operator or higher to process another operator's entries.

ALL specifies that all user ID entries are to be processed.

If password security is active, the operator running the HISTORY procedure must be classified as system operator or higher to process another operator's entries.

ALLWS specifies that history file entries from all display stations are to be processed. If no parameter is specified, ALLWS is assumed.

display id specifies the work station ID of a display station. Only those entries that match the specified display ID are processed.

ALLENTS specifies that all entries in the history file are to be processed. If no parameter is specified, ALLENTS is assumed.

procedure name specifies that only those entries associated with the specified procedure name are to be processed.

EONLY specifies that only the entries beginning with *E are to be processed. The *EJ indicates end-of-job entries; the *EP indicates end-of-print entries. This provides you with a summary of when jobs started and ended, together with the display stations the jobs were run from and the operators that ran the jobs.

ALLDAYS specifies that entries are to be processed regardless of their date. If no parameter is specified, ALLDAYS is assumed.

TODAY specifies that only those entries with today's system date are to be processed.

date specifies a date. Only those entries that match the specified date are to be processed. The date must be entered in the system format; either mmddyy (month, day, year), ddmmyy (day, month, year), or yymmdd (year, month, day). Use the STATUS SESSION command to determine the system date format.

from time specifies a beginning time. Only those entries that occur at or after the specified time are to be processed. The time must be entered in the format HHMMSS (hours, minutes, seconds). For example:

```
8:32:53 AM is entered 083253
1:12:00 PM is entered 131200
```

Any 6-digit number from 000000 through 235959 can be specified. If no parameter is specified, 000000 (the earliest time) is assumed.

to time specifies an ending time. Only those entries that occur at or before the specified time are to be processed. The time must be entered in the format HHMMSS (hours, minutes, seconds). Any 6-digit number from 000000 through 235959 can be specified. If no parameter is specified, 235959 (the latest time) is assumed.

SYSTEM specifies that the system history file is to be displayed or printed. If no parameter is specified, SYSTEM is assumed.

file name specifies the following:

- If LIST or CRT is specified, the file name specifies the disk file to be printed or displayed. The file must have been created using the COPYSYS parameter of the HISTORY procedure, or by the history file automatic copy function. For more information about the history file automatic copy function, see the "HISTCOPY Procedure" on page 4-207.

  ERASE cannot be specified if a file name other than SYSTEM is specified.

- If COPYSYS or COPYPRT is specified, the file name specifies the disk file to create. This disk file will contain the processed history file entries.

  If no file name is specified, and COPYSYS or COPYPRT is specified, the name assigned to the file is made up of the letters HF and the display ID. For example, if the display ID is W3, the name of the file would be HFW3.

NOERASE specifies that the entries processed are not to be erased. If no parameter is specified, NOERASE is assumed.

ERASE specifies that the entries processed are to be erased after they are processed.

Example 1

To display the current user's history file entries, an operator would enter:

```
HISTORY CRT
```

Example 2

To list the entire system history, and clear all the entries in the history file, the system operator would enter:

```
HISTORY LIST,ALL,,,,,,,ERASE
```

Example 3

To list all history file entries that were created as a result of the operator running jobs at a display station, the operator would enter:

```
HISTORY LIST
```

Example 4

To list the history file entries associated with the running of a procedure named TEST, an operator would enter:

```
HISTORY LIST,,,TEST
```

Example 5

To copy today's end-of-job entries to a disk file (in COPYPRT format) named HISTFILE, a system operator would enter:

```
HISTORY COPYPRT,,,EONLY,TODAY,,,HISTFILE
```

Sample HISTORY Display

When HISTORY CRT is run, the following display is shown (this display is shown with some sample information):

```
 HISTORY SCROLL                              WKSTN   USER     JOB NAME    TIME
                                               W3   MAT535    W3141114  14.12.29
*EJ  14.11.14  14.12.29  00.01.15  01/06/83  MAT535    W3  DEFINX21
                                               W1   JMG543    W1141146  14.12.35
  UPDATE TWOFIVE,TWOFIVE,CHAPTWO,0,,,FIVE,,SORTLIB,DASH
    #UPDATE *ALL                               W1   JMG543    W1141146  14.12.37
  HELP BLDLIBR                                 W3   MAT535    W3141333  14.13.34
                                               W1   JMG543    W1141146  14.13.57
*EJ  14.11.46  14.13.56  00.02.10  01/06/83  JMG543    W1  HELP
                                               W3   MAT535    W3141333  14.14.00
*EJ  14.13.33  14.14.00  00.00.27  01/06/83  MAT535    W3  HELP
                                               **   ********  ********  14.14.06
*EP  14.13.56  14.14.06  00.00.10  01/06/83  JMG543    P1  HELP      W1141146
  HELP HISTORY                                 W3   MAT535    W3141423  14.14.23
                                               W3   MAT535    W3141423  14.14.32
  HISTORY CRT,ALL,ALLWS,ALLENTS,ALLDAYS,000000,235959,SYSTEM,NOERASE
  SEU HELP                                     W2   JAJ532    W2141432  14.14.32
    #SEU,#SEULIB *ALL                          W2   JAJ532    W2141432  14.14.33
  HELP LIST ,,,,,,,,SORTLIB                    W1   JMG543    W1141433  14.14.34
  HISTORY CRT procedure is running             W3   MAT535    W3141423  14.14.36
  SEU procedure is running                     W2   JAJ532    W2141432  14.14.37
                    NO MORE ENTRIES IN HISTORY FILE
      Cmd7-End of job    Cmd8-Display oldest    Cmd9-Display newest
   Direction- B   Scan data- ____________________  Wkstn-     User- ________
```

Figure 4-10. Sample HISTORY CRT Display

The most recent entries in the history file for the user ID of the requester display station are displayed first.

Scan values may be entered at the bottom of the display. If you enter one or more values at the bottom of this display, the system searches the history file for those entries matching the scan values. **Scan data** applies only to the text portion of the history file entries. **Wkstn** and **User** apply only to the control information portion of the entries. The work station ID, user ID, job name, and time are considered to be control information for the display. The following scan values may be specified:

Direction Enter F for a forward scan or B for a backward scan. If you do not enter a value in this field, the system assumes a backward (B) scan through the history file.

Scan Data Enter a string of 1 through 20 characters to search for within the statement portion of the history file entries.

Wkstn Enter a 2-character display ID.

User Enter a 1- to 8-character user ID. If a user ID other than ALL was entered on the HISTORY procedure, this field will already be filled in and you will not be able to change it.

When you press the Enter key after entering a value into the above fields, the system searches the history file for the specified value. If you specify more than one scan value, all the specified conditions must be met for the search to be successful.

If a forward scan is specified (F in the direction field), the search starts with the first line currently displayed. If the search is successful, the characters found are underscored and highlighted.

To continue the scan operation with the same values, press the Enter key. The scan will continue beginning with the history file entry following the last entry currently displayed. If the specified scan values are not found in the history file, a message is displayed.

If a backwards scan is specified (B in the direction field), the search starts with the last line on the display, and scans backwards through the history file. If the scan is successful, the characters found are underscored and highlighted. If you press the Enter key to continue the scan, the search will resume beginning with the history file entry that comes before the first line on the display. If the specified scan values are not found in the history file, a message is displayed.

Command and Function Keys You Can Use

The following function and command keys are available on the history scroll display shown in Figure 4-10.

Roll Up (Roll ↑) Key

Used to page forward through the next, or the more recent, history file entries. When the last, or the most recent, entry in the history file is displayed, a *last entry* message is displayed. If the Roll Up key is used when the last entry is shown, the current display is reshown.

Roll Down (Roll ↓) Key

Used to page backwards through the previous, or the older, history file entries. When the first, or the oldest, entry in the history file is displayed, a *first entry* message is displayed. If the Roll Down key is used when the first entry is shown, the current display is reshown.

Command Key 3

Used to change the roll factor of the Roll Up (Roll ↑) and the Roll Down (Roll ↓) keys; that is, the number of history file entries to be paged through.

To change the roll factor for the remainder of the HISTORY procedure, type the new number of entries to roll in the roll factor field, and press the Enter key. Valid entries for the roll factor field are 1 through 999. If a value is not entered in this field, the system assumes the number of entries shown on a full display.

To change the number of entries to roll for one operation only of the roll key, type the new number of entries to roll, and then use the Roll Up or the Roll Down key. Valid entries for the roll factor field are 1 through 999. If a value is not entered in this field, the system assumes the number of entries shown on a full display.

Command Key 4

Used to specify whether you want to display the following control information for each entry to the history file:

- Work station ID
- User ID
- Job name
- Time

If the control information is currently displayed, use command key 4 to cause the screen to not display the control information. If the control information is not currently displayed, use command key 4 to cause the screen to display the control information.

Command Key 5

Used to specify whether you want to display the nonviewed entries to the history file. Viewed entries are those that were displayed before they were logged to the history file. Nonviewed entries include *E entries and all OCL statements generated by procedures.

If nonviewed entries are currently displayed, use command key 5 to cause the screen to not display the nonviewed entries. If only viewed entries are currently displayed, use command key 5 to cause the screen to also display nonviewed entries.

Command Key 6

Used to define what values must be met before an entry to the history file is selected to be displayed. That is, if a history file entry does not match the entries specified on this display, those history file entries are not shown. The following display is shown when command key 6 is pressed:

```
                        CHANGE SELECTION CRITERIA              Optional-*
          Cancels or changes the scroll display selection criteria

Work station ID . . . . . . . . . . . . . . . . . . . . . . . . __        *
User ID . . . . . . . . . . . . . . . . . . . . . . . . . . . . ________  *
Job name  . . . . . . . . . . . . . . . . . . . . . . . . . . . ________  *
Start time  . . . . . . . . . . . . . . . . . . . . . . hhmmss ________  *
Stop time . . . . . . . . . . . . . . . . . . . . . . . hhmmss ________  *
Entry date  . . . . . . . . . . . . . . . . . . . . system format ______  *


Cmd6-To cancel above selection and return to scroll display.
Cmd7-To return to scroll display without any change to selection criteria
```

If you entered a user ID other than ALL on the HISTORY procedure, the user ID field will already be filled in and you will not be able to change it. If you entered a display station ID, a start or stop time, or a date on the HISTORY procedure, these fields will be filled in, and you will be able to change them.

Command Key 7

Used to end the HISTORY procedure.

Command Key 8

Used to display the oldest history file entry.

Command Key 9

Used to display the newest history file entry.

Command Key 10

Used to control whether only the history file entries beginning with *EJ (end-of-job) or *EP (end-of-print) are displayed. If the *E entries and the other history file information are currently displayed, command key 10 causes only the *E entries to be displayed. If only the *E entries are displayed, command key 10 causes the *E entries and the other history file information to be displayed.

Sample HISTORY Listing

Figure 4-11 shows a sample printout of a history file. Following the figure, the items shown on the printout are explained.

```
HISTORY FILE DISPLAY          Master Console - W1       User - MAT535       Date - 10/31/82     Time - 14.50.03     Page - 1
                                        HISTORY LIST,ALL,ALLWS,ALLENTS,103182,110000,111900,SYSTEM,NOERASE
Date of first entry this page - 10/31/82                                                                WKSTN   USER     JOB NAME    TIME
                                                                                                        -----  --------  --------  --------
  BLDFILE BOTTLING,S,RECORDS,2,80,,T,,,NDFILE,,0                                                         W2   DMM543    W2105916  11.00.23
   BLDFILE procedure is running                                                                          ''   ''        ''        11.00.26
  OFF                                                                                                    W4   ********  ********  11.00.29
*EJ  10.59.16  11.00.34  00.01.18  10/31/82  DMM543    W2  HELP                                         W2   DMM543    W2105916  11.00.34
  HELP BLDFILE                                                                                          ''   ''        W2110045  11.00.46
  BLDFILE STEEL,S,RECORDS,2,80,,T,,,NDFILE,,0                                                           ''   ''        ''        11.01.06
   BLDFILE procedure is running                                                                         ''   ''        ''        11.01.08
*EJ  11.00.45  11.01.14  00.00.29  10/31/82  DMM543    W2  HELP                                         ''   ''        ''        11.01.14
  BASIC ,STROMBER                                                                                       ''   ''        W2110141  11.01.42
   BASIC procedure running                                                                              ''   ''        ''        11.01.43
  SYS-1076  Options (   3 ) CIM2                                                                        ''   ''        ''        11.01.45
  Invalid SIZE parameter                                                                                ''   ''        ''        11.01.45
  3                                                                                                     ''   ''        ''        11.01.55
*EJ  11.01.41  11.01.56  00.00.15  10/31/82  DMM543    W2  BASIC                                        ''   ''        ''        11.01.56
  BASIC STROMBER                                                                                        ''   ''        W2110202  11.02.02
   BASIC procedure running                                                                              ''   ''        ''        11.02.04
  5                                                                                                     ''   ''        ''        11.02.16
  ID-0008 SYS-6300  Options (012  ) SPWT                                    Master Console - W1   D248INFO  ********  11.03.59
  Printer P1 and the system are not communicating                    0200   Master Console - ''   ''        ********  11.04.00
          MAT535      TEMPLETEMLIB                                          Master Console - ''   MAT535    ********  11.05.04
    4,2                                                                     Master Console - ''   ''        ********  11.05.11
    ID-0008   REPLY command sucessful                                       Master Console - ''   ''        ********  11.05.11
                                                                              System Entry - **   ********  ********  11.05.12
*EP  10.50.19  11.05.12  00.14.53  10/31/82  DMM543    P1  BASIC     W2104722  0001  PP
                                                                              System Entry - **   ********  ********  11.17.49
*EP  11.15.01  11.17.49  00.02.48  10/31/82  MAT535    P1  COPYPRT   W1111119  MIKE  NC
  ID-0012 SYS-1404  Options (012  ) CSSM                                  Master Console - W1   MAT535    ********  11.17.50
  On printer P2 change to forms number MIKE...                            Master Console - ''   ''        ********  11.17.50
    7,1                                                                   Master Console - ''   ''        ********  11.17.58
    ID-0012   REPLY command sucessful                                     Master Console - ''   ''        ********  11.17.58
  ID-0013 SYS-1405  Options (012  ) CSSM                                  Master Console - ''   ''        ********  11.17.59
  Do you want spool separator pages on printer P2...                      Master Console - ''   ''        ********  11.17.59
                                                                              System Entry - **   ********  ********  11.18.03
*EP  11.17.49  11.18.03  00.00.14  10/31/82  MAT535    P1  COPYPRT   W1111119  MIKE  NC
    8,1                                                                   Master Console - W1   MAT535    ********  11.18.03
    ID-0013   REPLY command sucessful                                     Master Console - ''   ''        ********  11.18.04
                                                                              System Entry - **   ********  ********  11.18.16
*EP  11.18.03  11.18.16  00.00.13  10/31/82  MAT535    P1  COPYPRT   W1111119  MIKE  NC
                                                                              System Entry - **   ********  ********  11.18.32
*EP  11.18.16  11.18.32  00.00.16  10/31/82  MAT535    P1  COPYPRT   W1111119  MIKE  NC
                                                                              System Entry - **   ********  ********  11.18.43
*EP  11.17.49  11.18.43  00.00.54  10/31/82  MAT535    P2  COPYPRT   W1111639  MIKE  NC
                                                                              System Entry - **   ********  ********  11.18.46
*EP  11.18.33  11.18.46  00.00.13  10/31/82  MAT535    P1  COPYPRT   W1111119  MIKE  NC
                                                                              System Entry - **   ********  ********  11.18.56
*EP  11.18.46  11.18.56  00.00.10  10/31/82  MAT535    P1  COPYPRT   W1111119  MIKE  NC

    Number of entries displayed - 119
```

Figure 4-11. Sample Listing of the History File

Description of the Items on the History File Listing

Heading Information

WORKSTATION indicates the display station from which the HISTORY procedure was run.

USER indicates who ran the HISTORY procedure (the user ID).

TIME indicates the time the HISTORY procedure was run.

DATE indicates the date the HISTORY procedure was run.

Body of Report

Statement indicates the procedures, commands, OCL statements, and messages that were written to the History file. The statements are indented on the listing as follows:

Position	History File Text
1	Any entries beginning with *E.
3	Any entries that were displayed before being logged to the history file.
7	All OCL statements run during a procedure.
5	All other entries.

The *EJ (end-of-job) entries contain the following information:

1. The job's starting time.
2. The job's ending time.
3. The amount of time used to run the job (only valid for jobs that run less than 24 hours).
4. The date the job was run.
5. The user ID of the operator who ran the job.
6. The display station from which the job was run.
7. The name of the procedure that began the job.
8. Whether the job was a multiple requester terminal job (MRT), a nonrequester terminal job (NRT), run from the job queue (JOBQ), or a single requester terminal job (blank).

The *EP (end-of-print job) entries contain the following information:

1. The job's starting time.
2. The job's ending time.
3. The amount of time used to print the job.
4. The date the job was printed.
5. The user ID of the operator who created the print job.
6. The printer ID from which the job was printed.
7. The name of the procedure that created the print job.
8. The name of the job that created the print job.
9. The forms number used with the printed output.
10. A 2-character job completion code:

NC	Normal completion.
CP	An operator canceled the print job.
GP	An operator changed the print job.
HP	An operator held the print job.
PP	An operator stopped the print job.
TP	An operator restarted the print job.

WKSTN indicates the work station ID of the display station or printer associated with the history file entry.

USER indicates the user ID of the operator associated with the history file entry.

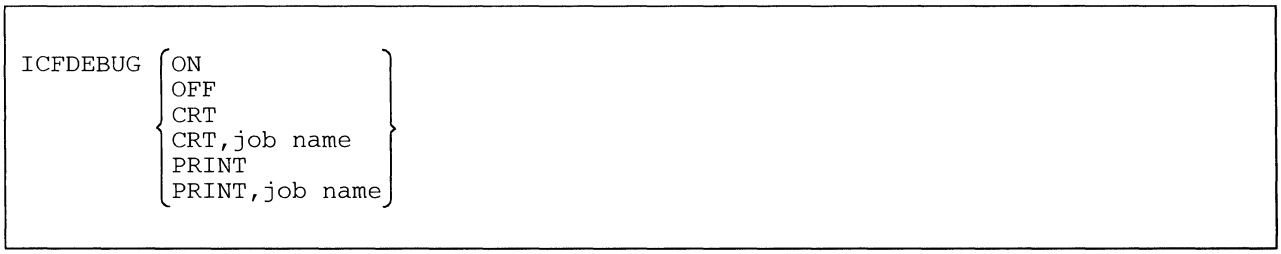
JOB NAME indicates the job associated with the history file entry.

TIME indicates the time the statement was logged to the history file.

ICFDEBUG Procedure

The ICFDEBUG procedure controls the running of the Interactive Communications feature (SSP-ICF) debug program. After the debug file is displayed or printed, the file is deleted.

For more information about SSP-ICF and the ICFDEBUG procedure, see the manual *Using System/36 Communications*.



S9020134-0

ON specifies that the debug program is to be started. The program begins recording SSP-ICF activities in a disk file. Any previous file created by the ICFDEBUG procedure is deleted.

OFF specifies that the debug program is to be stopped. No further SSP-ICF activity is to be recorded and the disk file created by the ICFDEBUG procedure is deleted.

CRT specifies that the file previously created by the ICFDEBUG procedure is to be displayed.

CRT,job name specifies that entries for the named job are to be displayed from the file previously created by the ICFDEBUG procedure.

job name specifies the 8-character name of the job. You can use the STATUS USERS control command to display the job names.

PRINT specifies that the file previously created by the ICFDEBUG procedure is to be printed.

PRINT,job name specifies that entries for the named job are to be printed from the file previously created by the ICFDEBUG procedure.

job name specifies the 8-character name of the job. You can use the STATUS USERS control command to display the job names.

Example 1

This example shows how to start the SSP-ICF debug program.

```
ICFDEBUG ON
```

Example 2

This example shows how to print the activity recorded after the procedure command ICFDEBUG ON was entered.

```
ICFDEBUG PRINT
```

ICVERIFY Procedure

The ICVERIFY procedure verifies the installation of the Interactive Communications feature (SSP-ICF) subsystems. See the manual *SSP-ICF Guide and Examples* for information about running this procedure.

IDDUDCT Procedure

The IDDUDCT procedure calls the interactive data definition utility (IDDU) to allow you to create, revise, delete, rename, or print a data dictionary.

For more information about IDDU, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For IDDU, specify:

  ```
  READINFO #IDDDOC,#IDDFLDR
  ```

- Select option 11 from the Use IDDU menu
- Press the Help key from any IDDU display

```
IDDUDCT
```

S9020135-0

The IDDUDCT procedure has no parameters.

IDDUDFN Procedure

The IDDUDFN procedure calls the interactive data definition utility (IDDU) to allow you to create, revise, copy, delete, rename, print, or show a list of where a field, format, or file definition is used.

For more information about IDDU, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For IDDU, specify:

  ```
  READINFO #IDDDOC,#IDDFLDR
  ```

- Select option 11 from the Use IDDU menu
- Press the Help key from any IDDU display

```
IDDUDFN
```

S9020136-0

The IDDUDFN procedure has no parameters.

IDDUDISK Procedure

The IDDUDISK procedure calls the interactive data definition utility (IDDU) to allow you to create, link, or unlink a disk file, or enter or update data in a disk file.

For more information about IDDU, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For IDDU, specify:

  ```
  READINFO #IDDDOC,#IDDFLDR
  ```

- Select option 11 from the Use IDDU menu
- Press the Help key from any IDDU display

```
IDDUDISK
```

S9020137-0

The IDDUDISK procedure has no parameters.

IDDULINK Procedure

The IDDULINK procedure links or unlinks a file on disk with a file definition in a data dictionary. Linking to the file definition allows a program or utility to access the file as it is defined by the file definition.

For more information about IDDU, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For IDDU, specify:

  ```
  READINFO #IDDDOC,#IDDFLDR
  ```

- Select option 11 from the Use IDDU menu
- Press the Help key from any IDDU display

To link a file on disk to a file definition:

```
IDDULINK LINK,file name,data dictionary name,file definition,[mmddyy]
                                                             [ddmmyy]
                                                             [yymmdd]
```

S9020138-0

To unlink a file on disk from a file definition:

```
IDDULINK UNLINK,file name,,,[mmddyy]
                            [ddmmyy]
                            [yymmdd]
```

S9020479-1

To unlink all the file definitions in a data dictionary:

```
IDDULINK UNLINK,ALL,data dictionary name
```

S9020480-0

LINK specifies that the file on disk is to be associated with the file definition in the data dictionary.

UNLINK specifies that the association between the file on disk and the file definition is to be removed.

file name specifies the name of the file on disk to be linked or unlinked.

ALL specifies that all file definitions in the specified data dictionary are to be unlinked from their associated files on disk. ALL can be specified only if UNLINK is specified.

data dictionary name specifies the data dictionary that contains the file definitions to be linked or unlinked. If the first parameter is UNLINK and the second parameter is a file name, this parameter is not required.

file definition specifies the name of the file definition to be linked to the disk file. If the first parameter is UNLINK, this parameter is not required.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be linked or unlinked. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If a date is not specified and more than one file exists with the same file name, the most recent file created is linked or unlinked.

IDDUPRT Procedure

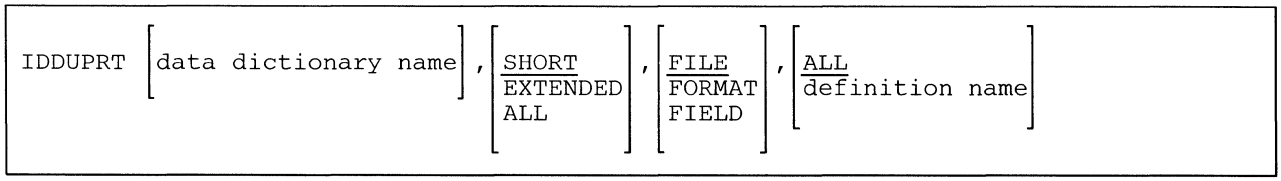
The IDDUPRT procedure prints field, format, or file definitions in a data dictionary.

For more information about IDDU, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For IDDU, specify:

  ```
  READINFO #IDDDOC,#IDDFLDR
  ```

- Select option 11 from the Use IDDU menu
- Press the Help key from any IDDU display



S9020139-0

data dictionary name specifies the data dictionary that contains the definitions to be printed.

SHORT specifies that the short version of the definition is to be printed. The short version always includes the date the definition was last revised and the user ID of the user who revised it; the date the definition was created and the user ID of the user who created it; and a short comment. The short version of a field definition includes the type of data, field length, and number of decimal places allowed in the field. A short format definition includes the input and output record length and field list. A short file definition includes the file type, maximum record length, and record format list. If no parameter is specified, SHORT is assumed.

EXTENDED specifies that the extended version of the definition is to be printed. The extended version always includes all the information supplied in a short definition plus a long comment and a list of where the definition is used. The extended version of a field definition includes additional attributes and numeric editing. An extended format definition includes communication attributes and record identification codes.

ALL specifies that the extended version of the definition is to be printed. For a file definition, all extended versions of format definitions within the file are also to be printed. For a format definition, all extended versions of field definitions within the format are also to be printed.

FILE specifies that one or all file definitions within the specified data dictionary are to be printed. If no parameter is specified, FILE is assumed.

FORMAT specifies that one or all format definitions within the specified data dictionary are to be printed.

FIELD specifies that one or all field definitions within the specified data dictionary are to be printed.

ALL specifies that all definitions within the specified data dictionary of the specified type are to be printed. If no parameter is specified, ALL is assumed.

definition name specifies the name of a single definition of the specified type to be printed.

IDDURBLD Procedure

The IDDURBLD procedure allows programs and utilities to use updated data definitions. Changes made to a definition are normally applied when the interactive data definition utility (IDDU) ends. If a program or utility is using the definition, however, changes cannot be applied until after the program or utility ends. The IDDURBLD procedure is used after the program ends to apply the changes that IDDU could not apply when it ended.

For more information about IDDU, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For IDDU, specify:

  ```
  READINFO #IDDDOC,#IDDFLDR
  ```

- Select option 11 from the Use IDDU menu
- Press the Help key from any IDDU display

```
IDDURBLD data dictionary name
```

S9020140-0

data dictionary name specifies the data dictionary that contains the data definition or definitions to be updated.

IDDUXLAT Procedure

RPG source specifications can be used to describe the content and layout of the disk files used by an RPG program. These file descriptions are internal to the program, and are known only to and can be used only by that program. The data in these files are called **program defined data**.

The interactive data definition utility (IDDU) also allows you to describe the content and layout of files. These descriptions are called **field**, **format**, and **file definitions**, and are stored in a **data dictionary**. The data in the files described by IDDU definitions are called **externally described data**.

You can use IDDU to create or maintain definitions simply by responding to prompts on a display. IDDU definitions are used by Query/36 and DisplayWrite/36 to access data in the files described by the definitions; RPG source specifications are not required and are not used by Query/36 and DisplayWrite/36.

The IDDUXLAT procedure translates the RPG source specifications contained in RPG program source members or in Text Management System (TMS) data definitions into IDDU definitions. The IDDUXLAT procedure is an easy way of creating new IDDU field, format, and file definitions using the information contained in existing RPG source specifications. If you use the IDDUXLAT procedure to translate your RPG source specifications, you might not need to separately use IDDU to create the definitions for the disk files you want to access through Query/36 or DisplayWrite/36.

IDDUXLAT

To create the IDDU definitions, the IDDUXLAT procedure interprets the entries in the following RPG specifications:

- H (control) specifications
- F (file) specifications
- I (input) specifications

These specifications must describe input or update disk files (except record address files and table files); all other file types, including work station and communications files, are ignored and cannot be translated using the IDDUXLAT procedure.

Using the I specifications in the RPG source member, the IDDUXLAT procedure creates the equivalent IDDU field definitions, as well as the equivalent IDDU format definitions for each record type contained in the file. The IDDUXLAT procedure uses the H and F specifications to create the IDDU file definition for each file described in the RPG source specifications.

Using the IDDUXLAT procedure, you can specify that only those source members of a particular subtype are translated. You can, for example, choose to translate all source members contained in a particular library into IDDU definitions, or translate only the source members containing TMS data definitions. In addition, you can choose to translate a specific source member, or a specified range of source members. For example, you can translate only the source member named RPGPROG1, or all source members that have names alphabetically between the names PROGAAA and PROGLLL (source member PROGAAA is translated, but PROGZZZ is not).

The IDDU definitions created by the IDDUXLAT procedure are placed in a specified IDDU data dictionary. The RPG source specifications that IDDUXLAT translates are not changed in any way.

To assist you in using the IDDU definitions created by the IDDUXLAT procedure, you can choose to print out the following information:

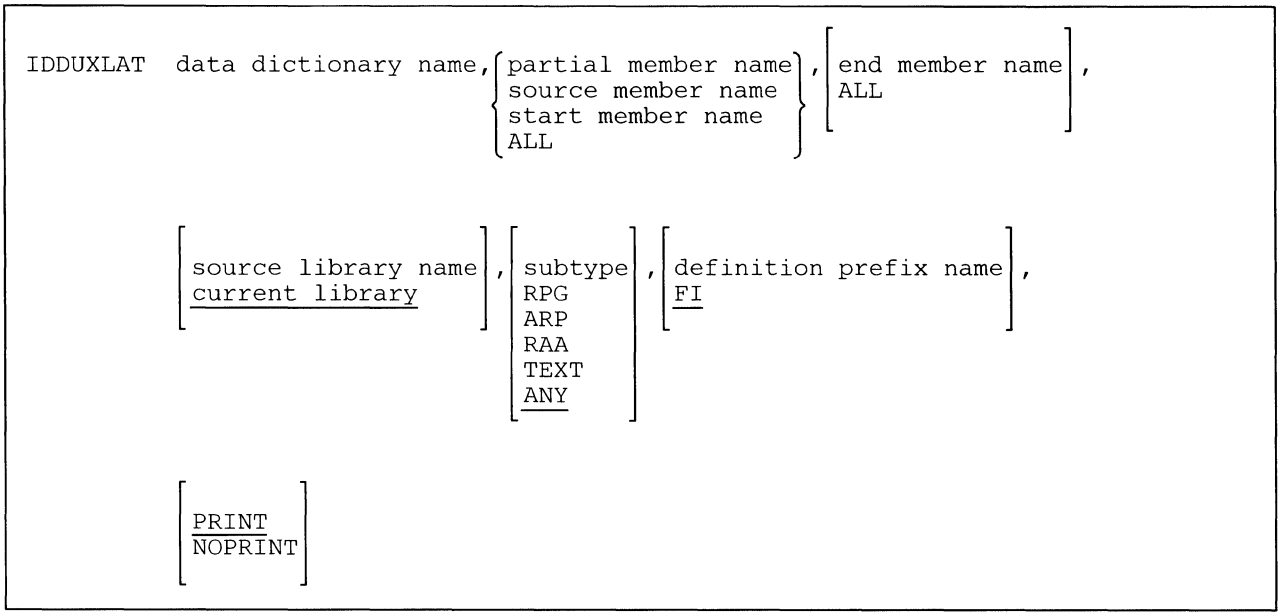
- A list of the parameters you specified in the IDDUXLAT procedure
- A list of the source specifications translated by the IDDUXLAT procedure and the errors encountered
- A list of the IDDU definitions created by the IDDUXLAT procedure
- Statistics showing the number of errors encountered in the source specifications and the number of IDDU definitions created by the IDDUXLAT procedure

Once the IDDU definitions and the files they describe are linked (see the IDDULINK procedure for more information), the definitions can then be used by the following utilities:

- Query/36, to produce reports from data contained in the file
- DisplayWrite/36, to use Query/36 to merge data contained in the file into a document

For more information about IDDU, read *Getting Started with Interactive Data Definition Utility*, SC21-8003, or the online information for IDDU (select option 11 from the IDDU system help menu). For detailed information about using the IDDUXLAT procedure, see Appendix G, "Using the IDDUXLAT Procedure" on page G-1 in this manual.

The IDDUXLAT procedure can be run from any display station.



S9020481-0

data dictionary name specifies the name of the IDDU data dictionary in which the IDDU field, format, and file definitions are stored. A data dictionary name must be specified; the specified data dictionary must already exist on your system. To create a data dictionary, use the IDDUDCT procedure.

partial member name specifies a 1- to 7-character string used in selecting the source members translated by the IDDUXLAT procedure. If ALL is also specified for the end member name, IDDUXLAT translates all the source members whose name begins with the specified character string.

source member name specifies the name of a single source member containing source specifications used to create IDDU definitions.

If you specify the name of a source member, but do not specify an end member name, only the specified source member is translated.

start member name specifies the first source member in a group of source members translated by the IDDUXLAT procedure.

If you also specify an end member name, all source members that have a name alphabetically between and including the start member name and the end member name are translated.

ALL If used for the source member parameter, all source members in the source library are translated by the IDDUXLAT procedure. If an end member name is specified, ALL must not be specified for the source member parameter.

If used for the end member parameter, all source members with a name beginning with the specified partial member name are translated.

You can use the subtype parameter (RPG, ARP, RAA, TEXT, or ANY) to indicate that only the source members of a particular subtype are translated.

end member name specifies the last source member in a group of source members translated by the IDDUXLAT procedure.

If you also specify a start member name, all source members that have a name alphabetically between and including the start member name and the end member name are translated.

source library name specifies the name of the library containing the source member or members translated by the IDDUXLAT procedure. If a source library name is not specified, the current library is assumed.

subtype specifies the particular kind of source member that is translated. One of the following must be specified:

RPG specifies that only the source members containing RPG specifications are translated.

ARP specifies that only the source members containing RPG auto report specifications are translated.

RAA specifies that only the source members containing RPG and RPG auto report specifications are translated.

TEXT specifies that only the source members containing TMS data definitions are translated. If TEXT is specified, the TMS data definitions must be contained in the source library named #DATADEF.

ANY specifies that all source members as indicated in the second and third parameters are translated. If you do not specify a subtype, ANY is assumed.

definition prefix name specifies a two-character prefix used in generating definition names.

Whenever possible, the IDDUXLAT procedure uses the field name shown in an I specification as the name of the corresponding field definition. In some cases, however, the field name might be the same as the name of an existing field definition. If the data type, field length, and number of decimal positions of the field described by the field definition are different than those of the field described by the I specification, IDDUXLAT generates a different and unused field definition name. IDDUXLAT uses the specified definition prefix name and adds a 4-digit sequence number. The generated field definition name is in the form xxnnnn, where xx is the specified definition prefix name, and nnnn is the sequence number.

If the field described by an I specification is the same as a field described by an existing field definition, a new field definition is not created from the I specification.

The specified prefix is also used in generating format definition and file definition names. The generated format definition names are in the form xxnnnn, where xx is the specified definition prefix name, and nnnn is a sequence number. If a file name specified in an F specification is the same as an existing file definition name, the specified prefix is used to generate a different and unused file definition name. The generated file definition name is in the form xxnnnnnn, where xx is the specified definition prefix name, and nnnnnn is a sequence number.

If you do not specify a definition prefix name, FI is assumed.

PRINT specifies that the following is printed:

- A list of the parameters you specified in the IDDUXLAT procedure
- A list of the source specifications translated by the IDDUXLAT procedure and any errors encountered
- A list of the IDDU definitions created by the IDDUXLAT procedure
- Statistics showing the number of errors encountered in the source specifications and the number of IDDU definitions created by the IDDUXLAT procedure

PRINT is the default.

NOPRINT specifies that no listing is printed.

Example 1

Translate all TMS data definitions contained in the library named #DATADEF. The IDDU definitions created by IDDUXLAT will be stored in the IDDU data dictionary named DICTABC. If duplicate names are found, the names of the definitions will begin with the characters FI. All possible printed output is requested.

```
IDDUXLAT DICTABC,ALL,,#DATADEF,TEXT,FI,PRINT
```

Example 2

Translate all source members contained in the library named RPGLIB. The IDDU definitions will be stored in the IDDU data dictionary named DICTABC. If duplicate names are found, the names of the definitions will begin with the characters FI. All possible printed output is requested.

```
IDDUXLAT DICTABC,ALL,,RPGLIB,ANY
```

Example 3

Translate all RPG auto report source members contained in the library named RPGLIB. The IDDU definitions will be stored in the IDDU data dictionary named DICTABC. If duplicate names are found, the names of the definitions will begin with the characters AR. No printed output is requested.

```
IDDUXLAT DICTABC,ALL,,RPGLIB,ARP,AR,NOPRINT
```

Example 4

Translate a single source member named RPGPROG1 contained in the library named RPGLIB. The IDDU definitions will be stored in the IDDU data dictionary named DICTABC. If duplicate names are found, the names of the definitions will begin with the characters PR. No printed output is requested.

```
IDDUXLAT DICTABC,RPGPROG1,,RPGLIB,,PR,NOPRINT
```

Example 5

Translate a group of source members contained in the library named RPGLIB. The first source member to be translated is PROGAAA, and the last source member to be translated is PROGLLL. Only RPG source members should be translated. The IDDU definitions will be stored in the IDDU data dictionary named DICTABC. If duplicate names are found, the names of the definitions will begin with the characters FI. All possible printed output is requested.

```
IDDUXLAT DICTABC,PROGAAA,PROGLLL,RPGLIB,RPG
```

Example 6

Translate a group of source members contained in the library named RPGLIB. All source members whose name begins with the characters PROG should be translated. The IDDU definitions will be stored in the IDDU data dictionary named DICTABC. If duplicate names are found, the names of the definitions will begin with the characters FI. All possible printed output is requested.

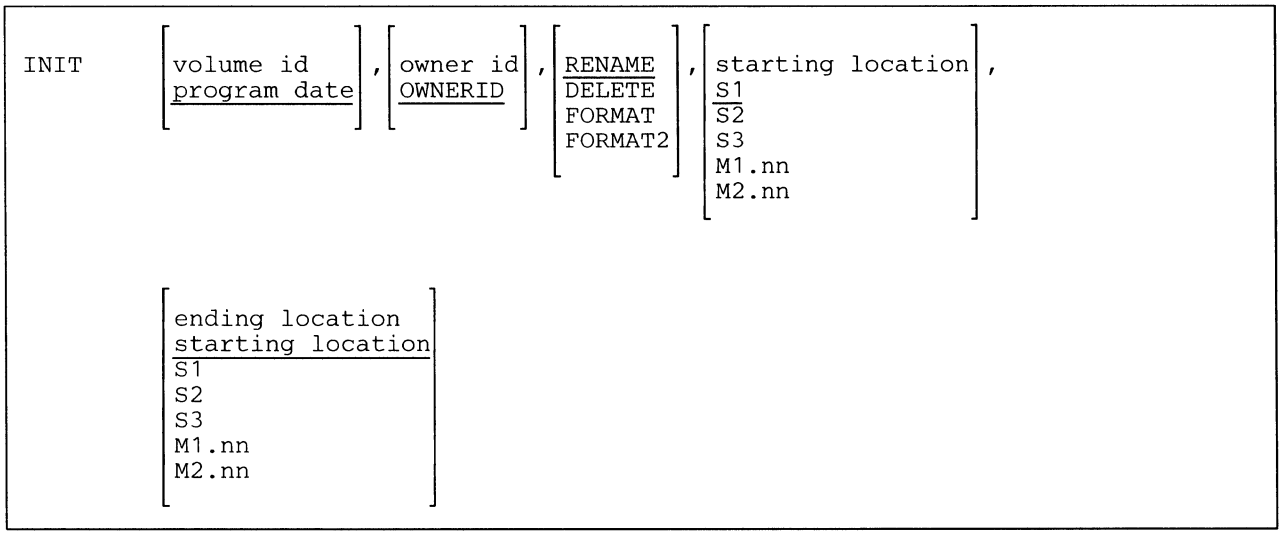
```
IDDUXLAT DICTABC,PROG,ALL,RPGLIB,ANY
```

INIT Procedure

The INIT procedure prepares one or more diskettes so they can be used to save files and libraries. This preparation is called **initialization**. You can also rename or erase a diskette. The INIT procedure does some or all the following functions:

- Writes identifying names on the diskette, called the volume ID (also called the volume identifier or pack ID) and the owner ID.
- Formats the control portion of the diskette (cylinder 0).
- Ensures that the diskette is usable.
- Writes sector addresses on the diskette.

The INIT procedure runs the $INIT utility program.



S9020141-0

volume id specifies the following:

- If RENAME, FORMAT, or FORMAT2 is specified, this parameter specifies the volume ID to assign the diskette. The volume ID can be from 1 through 6 characters (either alphabetic or numeric). The volume ID is placed in the volume ID field of the diskette volume label. If a volume ID is not specified, the program date (job step date) is written in the volume ID field. The date is written in yymmdd (year-month-day) format.
- If DELETE is specified, this parameter specifies the volume ID of the diskette that is to have its files deleted. The volume ID specified is checked against the existing volume ID of the diskette to ensure that the correct diskette is inserted.

owner id specifies the owner ID of the diskette, and can be from 1 through 14 characters (either alphabetic or numeric). If an owner ID is specified and if RENAME, FORMAT, or FORMAT2 is also specified, the owner ID is placed in the owner ID field of the diskette volume label. If an owner ID is not specified, the word OWNERID is written in the owner ID field.

RENAME specifies that the diskette is to be renamed (that is, the volume ID and owner ID fields in the diskette are to be assigned new values). Any files on the diskette are not affected. If a parameter is not specified, RENAME is assumed.

DELETE specifies that any active files on the diskette are to be removed, thereby making room on the diskette to hold more information.

FORMAT specifies how the surface of the diskette is to be initialized. For a diskette 1 diskette, the diskette is formatted in the 128-byte per sector format. For a diskette 2D diskette, the diskette is formatted in the 256-byte per sector format.

If FORMAT is specified, the diskette can contain basic data exchange files. For more information about basic data exchange files, see the "TRANSFER Procedure" on page 4-542.

FORMAT2 specifies how the surface of the diskette is to be initialized. For a diskette 1 diskette, the diskette is formatted in the 512-byte per sector format. For a diskette 2D diskette, the diskette is formatted in the 1024-byte per sector format. This format should generally be used. FORMAT2 diskettes cannot contain basic data exchange files.

starting location specifies the starting location of a diskette slot to be used. If a parameter is not specified, S1 is assumed. This parameter is ignored if the system does not have a diskette magazine drive. The starting location must be less than the ending location.

ending location specifies the ending location of a diskette slot to be used. If a parameter is not specified, the starting location is assumed. This parameter is ignored if the system does not have a diskette magazine drive.

The ending location (if specified) must be greater than the starting location and must not cause a jump from the diskette slots (S1, S2, and S3) to the magazine slots (M1 and M2). That is, specifying a starting location of S3 and an ending location of M1 is not correct. You can specify a starting location in magazine 1 and an ending location in magazine 2.

The starting and ending locations can be any of the following:

S1, S2, or S3 specifies one of the three diskette slots.

M1 or M2 specifies one of the two diskette magazine slots. If M1 or M2 is specified for the starting location, and no ending location is specified, the entire magazine (all 10 slots) is processed. If M1 or M2 is specified for the ending location, it is the same as specifying M1.10 or M2.10, respectively.

M1.nn or M2.nn specifies a specific position within a magazine. **nn** is a number from 01 through 10 specifying the position within the magazine.

Example 1

Rename a diskette so that the new volume ID is VOL001 and the new owner ID is YOURNAME. The diskette is in diskette slot S1.

```
INIT VOL001,YOURNAME
```

Example 2

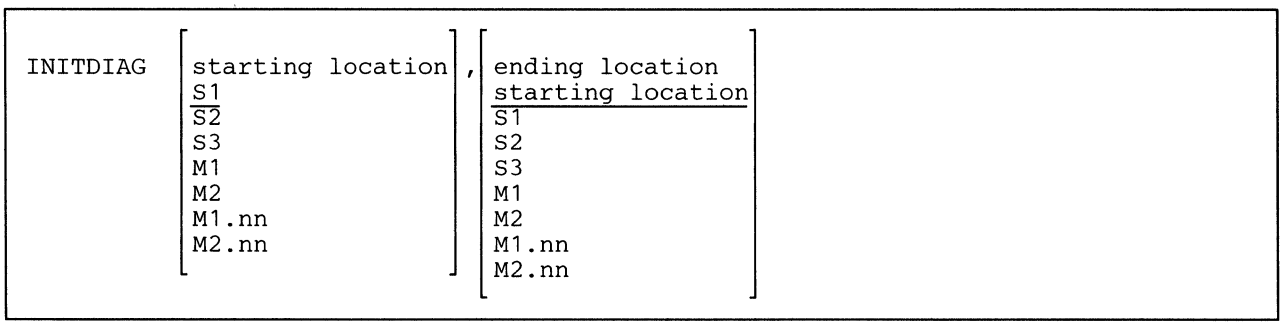
This example shows how to initialize ten diskette 1 diskettes in 512-byte format (or ten diskette 2D diskettes in 1024-byte format) so the volume ID is VOL002, and the owner ID is OWNERID. The diskettes are contained in a diskette magazine that is located in magazine slot M1.

```
INIT VOL002,,FORMAT2,M1
```

INITDIAG Procedure

The INITDIAG procedure initializes diagnostic (microcode) diskettes. The INITDIAG procedure can be run on any work station, evoked or run on the job queue.

The INITDIAG procedure runs the $INIT utility program.



S9020508-0

starting location specifies the location of the first diskette to be initialized as a diagnostic diskette. If a parameter is not specified, S1 is assumed. This parameter is ignored if the system does not have a diskette magazine drive. The starting location must be a number less than the ending location.

ending location specifies the location of the last diskette to be initialized. If a parameter is not specified, the starting location is assumed. This parameter is ignored if the system does not have a diskette magazine drive.

The ending location (if specified) must be a number greater than the starting location and must not cause a jump from the diskette slots (S1, S2, and S3) to the magazine slots (M1 and M2). For example, specifying a starting location of S3 and an ending location of M1 is not correct. However, you can specify a starting location in magazine 1 and an ending location in magazine 2.

The starting and ending locations can be any of the following:

S1, S2, or S3 specifies one of the three diskette slots.

M1 or M2 specifies one of the two diskette magazine slots. If M1 or M2 is specified for the starting location, and no ending location is specified, the entire magazine (all 10 slots) is processed. If M1 or M2 is specified for the ending location, it is the same as specifying M1.10 or M2.10, respectively.

M1.nn or M2.nn specifies a specific position within a magazine. **nn** is a number from 01 through 10 specifying the position within the magazine.

Example

To initialize a diskette located in slot 2:

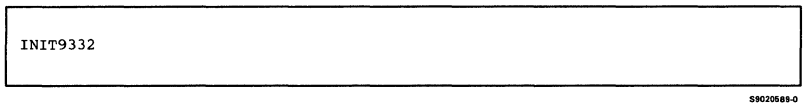
```
INITDIAG S2
```

INIT9332 Procedure

The INIT9332 procedure initializes and formats the 9332 Disk Unit to allow the 9332 Disk Unit to transfer data with the system. The INIT9332 procedure requires that the drive test diskette be inserted into the diskette drive. The drive test diskette is located in the storage box along with other diskettes and manuals. The operator must end all jobs running on the system or INIT9332 will not run.

After the INIT9332 presents warnings and the Insert Diskette display, it will present a selection display of all 9332 Disk Units that have been configured on the system. The selection display will present the logical address and the corresponding unit serial number of each 9332 Disk Unit configured. At this selection display, the operator is instructed to select the disk unit(s) that the INIT9332 will initialize.

Note: Running INIT9332 will destroy programs and data stored on the 9332 external disk(s) being initialized. It is recommended that all programs and data in the external disk be saved or backed up. In order to run the INIT9332 procedure to a 9332 Disk Unit, it must be configured as part of the system.



The INIT9332 procedure has no parameters.

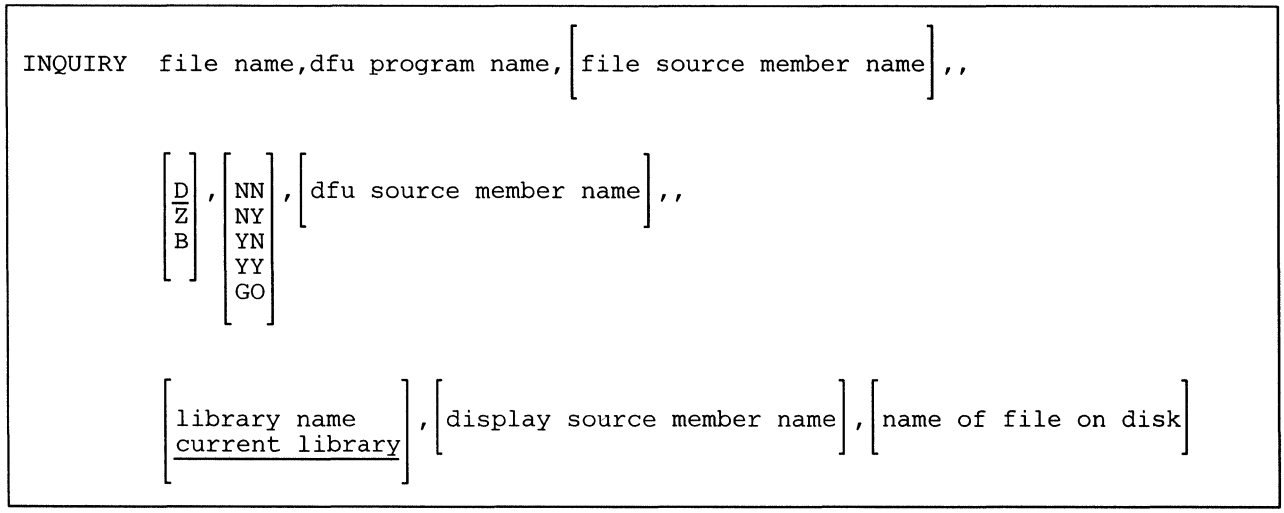
Example

This example initializes and formats the 9332 Disk Unit.

```
INIT9332
```

INQUIRY Procedure

The INQUIRY procedure allows you to display a disk file using the data file utility (DFU). File specifications are used to define the format of the records in the file. For more information about DFU, see the *DFU Guide*.



S9020142-0

file name specifies the file to be displayed. The file name can be from 1 through 8 alphameric characters.

dfu program name specifies the DFU program to be used to process the file. If the program does not exist in the library, DFU starts the setup procedures. If the program does exist in the library, DFU goes directly to displaying the file.

file source member name specifies the source member containing the file description (F-specifications) and record input specifications (I-specifications) that describe the file to be processed. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are taken as the data description.

This parameter is prompted for if it is not specified. This parameter is required if the specified format description does not exist and job setup must be performed.

D, Z, or B indicates whether unkeyed numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are **D**, **Z**, or **B**. If no parameter is specified, D is assumed.

D or B specifies a data file with blank fill of unkeyed numeric fields.

Z specifies a data file with zero fill of unkeyed numeric fields.

NN, NY, YN, YY, or GO specifies how the source specifications are to be processed. This 2-character parameter is used if the DFU program does not exist and the setup must be performed. It indicates whether the DFU specifications for this job are already stored in the library, and whether they are to be stored in the library when the setup is complete. The source member of DFU specifications is stored or looked for in the current library, unless a library name is specified.

NN Stored DFU specifications are not used for this program setup. You are prompted to create the specifications but they are not saved in the library.

NY Stored DFU specifications are not used for this program setup. You are prompted to create the specifications; once created, they are stored in the library.

YN Stored DFU specifications are used for this program setup. You are not prompted, but can update the specifications before the format description is created. If you change the specifications at the display station, only this program is affected; the changes are not stored in the library.

YY Stored DFU specifications are used for this program setup. You are not prompted, but can update the specifications before the DFU program is created. The specifications in the library are replaced by the updated specifications.

GO Stored DFU specifications are used for this program setup. You are not prompted and cannot update the specifications before the DFU program is created unless errors are found. If errors are found in the specifications, you must correct the specifications before the DFU program is created. However, the stored DFU specifications are not changed.

dfu source member name specifies the source member that contains, or will contain, saved DFU specifications. This parameter is required if the DFU source processing parameter is not NN.

library name specifies the library that contains or will contain the DFU specifications. All library members associated with the DFU job are looked for, or stored, in this library. If this parameter is not specified, the current library is assumed.

display source member name specifies the member in which DFU is to store the display format source specifications when setting up a DFU job. If this parameter is specified, you can change the generated display format source specifications at another time, compile them to replace the existing display format load member, and then run the same DFU job with the data displayed in the redefined fields.

name of file on disk specifies the name in the disk VTOC of the file to be displayed, if it is different from the name specified in the DFU program. If this parameter is specified, you can have several programs that refer to different, logical files display the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file to be displayed by the DFU program, you are prompted for that parameter.

Example

This example shows how to display a disk file named FILE1. The DFU source specifications to be used to display the records are named DFUSETUP. The specifications are in the current library.

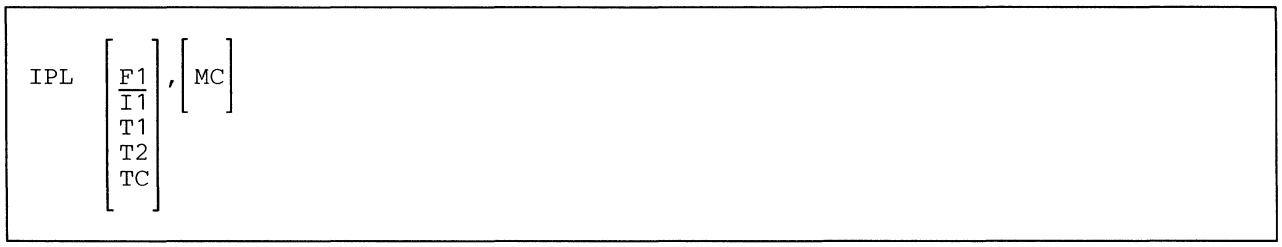
```
INQUIRY FILE1,DFIL1FMT,,,,GO,DFUSETUP
```

IPL Procedure

The IPL procedure allows you to initiate an IPL of your system through program control. You can use this procedure to:

- Perform an IPL from disk without running through the IPL hardware diagnostics
- Reload the system library from diskette
- Reload the system library from tape
- Update the system microcode

The IPL procedure requires that no user tasks be active. This includes spooling and tracing to alternate trace tables. If security is active on the system, you must run this procedure from a system service device, and you must have a security classification of at least system console operator.



S9020576-0

F1 specifies that an IPL is to be initiated from the disk. If this parameter is not specified, F1 is assumed. This can be used to provide a fast IPL (no hardware diagnostics are run) after PTF application, after using the CNFIGSSP procedure, or after using the SETCOMM procedure.

I1 specifies that the diskette drive is to be used to reload the system library. The diskettes must have been created by saving the system library to diskette using the SAVELIBR procedure. For systems with magazine diskette drives, only location M1.01 is supported as the slot containing the first diskette.

T1, T2, or TC specifies that a tape drive is to be used to reload the system library. T1 indicates that the tape containing the system library is mounted on tape drive 1. T2 indicates that the tape containing the system library is mounted on tape drive 2. TC indicates that the tape containing the system library is a tape cartridge. The reload tape must have been created by saving the system library on tape using the SAVELIBR procedure. Multiple volume tapes are not supported.

MC specifies that the system microcode will be updated using the microcode diskette that was shipped from IBM Software Distribution (ISD).

Example

The following example shows how you can reload the system library from diskette.

```
IPL I1
```

ITF Procedure

The Interactive Terminal Facility (ITF) procedure allows the System/36 user to send and receive data through applications such as TELEMAIL[1], and electronic message service offered by GTE Telenet for asynchronous terminals. Before you can start ITF, you must enable an asynchronous subsystem using the ENABLE procedure. Refer to the manual *Using the Asynchronous Communications Support*, for more information about ITF.

```
ITF   location name
```

S9020530-0

location name specifies the name of the remote location with which you want to communicate.

Example

To communicate with a remote location name of Chicago.

```
ITF CHICAGO
```

1 TELEMAIL and Telenet are registered servicemarks of the GTE Telenet Communications Corporation.

IWLOAD Procedure

The IWLOAD procedure creates a library named #IWLIB and copies PC Support/36 from diskette into that library. IWLOAD copies additional support into the system library (#LIBRARY). Two files, #IWPCLD1 and #IWPCLD2, are also copied from diskette. These files (also known as disks), contain the IBM Personal Computer portion of PC Support/36. For more information, see the manual *PC Support/36 Technical Reference.*

The IWLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the IWSAVE procedure. See the "IWSAVE Procedure" on page 4-245 for information about how to save PC Support/36 on diskette.

The IWLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration.* You should normally use the CNFIGSSP procedure to add support to the system. You should use the IWLOAD procedure to restore support that has been saved by IWSAVE.

If PC Support/36 is not currently on the system, you must use the TOLIBR procedure to copy the diskette file IWS into #LIBRARY before running IWLOAD.

If #LIBRARY was backed up with PC Support/36 on the system and then replaced before IWLOAD is run, you do not have to copy the diskette file IWS using the TOLIBR procedure.

```
IWLOAD [A1 ] , [S1   ]
       [A2 ]   [S2   ]
       [A3 ]   [S3   ]
       [A4 ]   [M1.nn]
               [M2.nn]
```

S9020144-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and the files #IWPCLD1, #IWPCLD2, and #IWLIB are placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, the files are placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

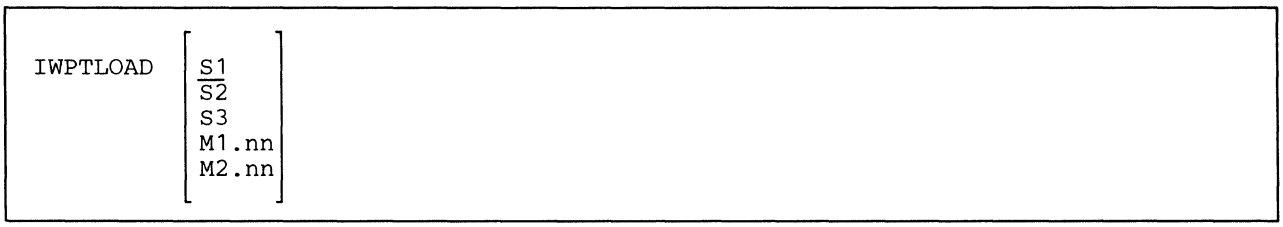
Create the files #IWPCLD1, #IWPCLD2, and #IWLIB on disk and copy PC Support/36 from diskette.

```
IWLOAD
```

IWPTLOAD Procedure

The IWPTLOAD procedure copies PC Support/36 pass-through support from a backup diskette to the PC Support/36 library, #IWLIB, and system library, #LIBRARY. The IWPTLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the IWPTSAVE procedure. See the "IWPTSAVE Procedure" on page 4-244 for more information about how to save PC Support/36 pass-through support on diskette.

Note: PC Support/36 pass-through support requires PC Support/36 to be on the system to load the PC Support/36 pass-through support. If PC Support/36 is not on the system, you will be prompted to load PC Support/36 before PC Support/36 pass-through support can be loaded.



S9020622-0

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

Example

Copy PC Support/36 pass-through support to #IWLIB and #LIBRARY from a backup diskette.

```
IWPTLOAD
```

IWPTSAVE Procedure

The IWPTSAVE procedure copies PC Support/36 pass-through support from the libraries #IWLIB and #LIBRARY onto diskette. The IWPTLOAD procedure should be used to load PC Support/36 pass-through support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPIWS. The diskette must be in slot 1.

```
IWPTSAVE
```

S9020623-0

The IWPTSAVE procedure has no parameters.

Example

Copy PC Support/36 pass-through support onto diskette.

```
IWPTSAVE
```

IWSAVE Procedure

The IWSAVE procedure copies PC Support/36 onto two diskettes. PC Support/36 from the libraries #IWLIB and #LIBRARY is copied. You should use the "IWLOAD Procedure" on page 4-242 to load PC Support/36 from the backup diskettes. The diskettes to contain the saved copy must have a volume ID of PPIWS and be located in diskette slot S1.

```
IWSAVE
```

S9020145-0

The IWSAVE procedure has no parameters.

JOBSTR Procedure

The JOBSTR procedure copies a diskette, tape, or tape cartridge file that contains one or more procedure members and source members to a specified library. You can also specify the name of a procedure to be run after the members in the diskette, tape, or tape cartridge file are copied.

The file containing the procedure and/or source members must be readable by the JOBSTR procedure. The file can be either a **sector-mode** file or a **record-mode** file.

Sector mode files and record mode files are created by the FROMLIBR procedure. You can also create a record mode file using a system other than the System/36. See the "FROMLIBR Procedure" on page 4-193 for more information about creating these files.

Only source and procedure members can be stored as record-mode files. Records in a record-mode file can be from 40 to 120 characters long, but all records in a file must have the same length. The SSP puts blanks at the end of lines that are shorter than the record length or truncates records to the specified record length. The members in a record-mode file are preceded by a COPY statement and are followed by a CEND statement. See "COPY and CEND Statements" on page A-69 for information about these statements.

JOBSTR runs the $MAINT utility program and may process a JOBQ OCL statement.

```
JOBSTR   file name,[procedure name],[SAVE / NOSAVE],[library name / current library],

         [Q / jobq prty],[S1 / S2 / S3 / M1.nn / M2.nn],[AUTO / NOAUTO],[I1 / T1 / T2 / TC],[REWIND / LEAVE / UNLOAD]
```

S9020146-1

file name specifies the name of the file containing one or more source or procedure members.

procedure name specifies the procedure to be run after the members are copied. The procedure need not be one of the members that is to be copied, but must be in the same library as the copied members.

SAVE specifies that the procedure specified by **procedure name** is to remain in the library after it runs. If a parameter is not specified, SAVE is assumed.

NOSAVE specifies that after the procedure specified by **procedure name** has run, it is to be removed from the library.

library name specifies the library into which the members are to be copied. If a library name is not specified, the current library is assumed.

Q specifies that the procedure is to be run from the job queue, and has a job queue priority of 3; see the description of *jobq prty*, which follows, for a description of the job queue priority. If no parameter is specified but a procedure name is specified, the procedure is run as part of the job containing the JOBSTR procedure.

jobq prty specifies the job queue priority for the job; that is, the job's order of processing from the job queue. The job queue priority can be any decimal number from 0 through 5. The system begins running jobs in order of decreasing job queue priority. For example, all jobs with a job queue priority of 5 are run before any other jobs in the job queue. Jobs with the same job queue priority are run in the order they were placed in the job queue. Jobs with a job queue priority of 1 are the last jobs run by the system. Job queue priority 0 is usually stopped, that is, any jobs placed on the job queue with a priority of 0 will not be run until the system operator starts priority 0. If no parameter is specified but a procedure name is specified, the procedure is run as part of the job containing the JOBSTR procedure.

S1, S2, or S3 specifies the diskette slot containing the first diskette from which members are to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette from which members are to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.)

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01 and continues through M2.10.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive is used.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.
- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.
- If T1 or T2 is specified, only that specified tape drive is used for all tape volumes.
- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

I1 specifies that the file to be copied is on diskette.

T1, T2, or TC specifies that the file to be copied is on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

REWIND specifies that the tape should be rewound after the JOBSTR procedure has run. REWIND is assumed if this parameter is not specified. This parameter is not allowed if F1 or I1 is specified.

LEAVE specifies that the tape should be left where it is after the JOBSTR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is not allowed if F1 or I1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the JOBSTR procedure has run. This parameter is not allowed if F1 or I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

Examples

For the following examples, a diskette file labeled JOBS contains the following job stream:

```
// COPY NAME-PROC1,LIBRARY-P,HIST-NO
// LOAD PROGRAM
// RUN
// END
// CEND
// COPY NAME-PROC2,LIBRARY-P
// LOAD PROGRAM2
// FILE NAME-FILEABC
// RUN
// END
// CEND
// COPY NAME-MEMBER1,LIBRARY-S
THIS IS A SOURCE MEMBER
This is the second line of the source member
// CEND
```

Example 1

Copy the procedure members (PROC1 and PROC2) and the source member (MEMBER1) into the library named MYLIB.

```
JOBSTR JOBS,,,MYLIB
```

Example 2

Copy the procedure members (PROC1 and PROC2) and the source member (MEMBER1) into a library called MYLIB and then run PROC1 (PROC1 should be saved).

```
JOBSTR JOBS,PROC1,,MYLIB
```

KEYS Procedure

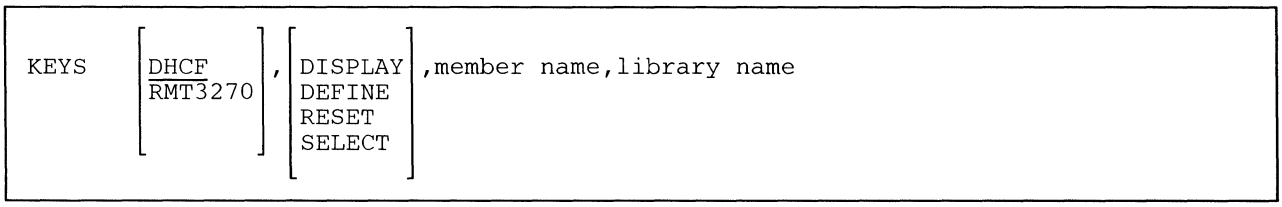
The KEYS procedure is used to perform the following functions:

- Display the current 3270-to-5250 keyboard map, which could be the default map assigned by the Distributed Host Command Facility (DHCF) when a 3270 user signs on DHCF, or the default map assigned by the 3270 Remote Attachment support when a 3270 Remote Attachment is varied on, or a user-defined map.
- Define a different set of keyboard mapping values for a 3270 keyboard.
- Select a keyboard map from any of those defined for that 3270 user.
- Reset the current keyboard map to be the DHCF-supplied default keyboard map or the RMT 3270-supplied default keyboard map.

The KEYS procedure can be run by a host system user from any 3270 remote attach display station, or it can be run by a System/36 user (on a 5250 display). All of the options listed can be performed at a 3270 display station, but only the define option can be performed at a 5250 display.

Note: DHCF: Before the KEYS procedure can be entered at a 3270 device (or one that is emulating a 3270 device), DHCF must be active on the System/36; the user must have successfully logged on the HCF host system and signed on the System/36.

RMT 3270: Before the procedure can be entered at a 3270 device (or one that is emulating a 3270 device), the device must be varied on and the user must be signed to the System/36.



S9020509-1

DHCF specifies that the type of keyboard mapping to be used is for DHCF. That is, the keyboard and display station that are to be used as a remote 5250 is a 3277, 3278, or 3279 display. If this parameter is not specified, DHCF is assumed.

RMT3270 specifies that the type of keyboard mapping to be used is for 3270 Remote Attachment support. 3270 Remote Attachment support allows a 3274 Remote Controller to be attached via communications lines to a System/36. This allows remote 3277, 3278, and 3279 displays to be used.

DISPLAY specifies that the current keyboard map for the 3270 display is to be presented. When DISPLAY is specified, the keyboard map currently being used for the 3270 display station is shown; it could be the DHCF-supplied default map, or the RMT3270-supplied default map, or a user-defined map. The DISPLAY option is valid only at a 3270 display station.

DEFINE specifies that a new 3270-to-5250 keyboard map is to be defined and stored in a source member or that an existing mapping member to be edited. Only members originally created by the KEYS procedure can be edited. A new map can be defined (or an existing map can be redefined) from either a 3270 or 5250 display station, but it cannot be activated at a 5250 display station. The default map members (HCFDFLTS and DFLT3270 in #LIBRARY) cannot be edited. A user-defined map member must be selected to be active.

SELECT specifies that a previously defined keyboard map is to be activated as the current keyboard map for use in a DHCF session. The map is selected from any of those listed when the KEYS procedure specifies the DISPLAY option. The SELECT option is valid only at a 3270 display station. The default map members (HCFDFLTS and DFLT3270 in #LIBRARY) cannot be selected. The RESET option causes these to be activated.

RESET specifies that the current user-defined keyboard map is to be replaced by the DHCF-supplied default keyboard map. The RESET option is valid only at a 3270 display station.

member name specifies the name of the keyboard mapping member to be selected, defined, or redefined. This parameter is required if the DEFINE or SELECT option is specified. It is not valid if the DISPLAY or RESET option is specified.

library name specifies the name of the library containing the keyboard mapping member to be selected, defined, or redefined. If this parameter is not specified for the DEFINE or SELECT options, the current session library is assumed. This parameter is not valid for the DISPLAY and RESET options.

Example 1

This command starts the definition of a 3270 keyboard mapping member that is to be named KEYS3279 and stored in the JONES library. This command could be entered by a host system user or a System/36 user.

```
KEYS DHCF,DEFINE,KEYS3279,JONES
```

Example 2

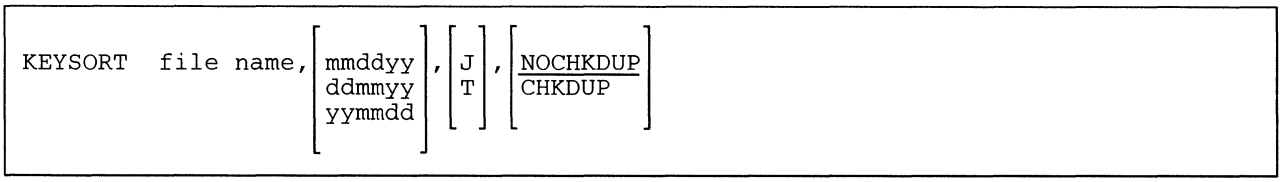
This command, entered at a 3270 display station, specifies that the user-defined 3270 keyboard map named KEYS3279 is to be activated as the current keyboard map for the session. The name of the member containing this map should be located in the current session library (JONES, in this case).

```
KEYS ,SELECT,KEYS3279
```

KEYSORT Procedure

The KEYSORT procedure sorts the keys for a specified indexed disk file. The primary index of the specified indexed file is sorted, unless you specify an alternative index for the file name. An indexed file cannot be sorted by the KEYSORT procedure if some other job has exclusive use of the file (that is, the job is not sharing the file). For more information about file sharing, see the "FILE OCL Statement (for Disk Files)" on page 5-32. For more information about sorting index keys, see the *Concepts and Programmer's Guide*.

The KEYSORT procedure runs the $DDST utility program.



S9020147-0

file name specifies the indexed file whose keys are to be sorted.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If a date is not specified, the index keys in the file with the specified label and the most recent creation date are sorted.

J or T specifies the retention of the file. By specifying J, or T you are defining a specific file to be processed. When a file retention is not specified, the order of search is first J (job files), then T (resident files). The KEYSORT is done on the first file found with the specified file name and date. (The KEYSORT procedure only sorts the keys of one file.)

NOCHKDUP specifies that no check should be made for duplicate keys. If no parameter is specified, NOCHKDUP is assumed.

CHKDUP specifies that the KEYSORT procedure should check for duplicate keys in the index. If duplicate keys are found, a message is displayed.

Example

Sort the index keys in an indexed disk file named PAYRCD.

```
KEYSORT PAYRCD
```

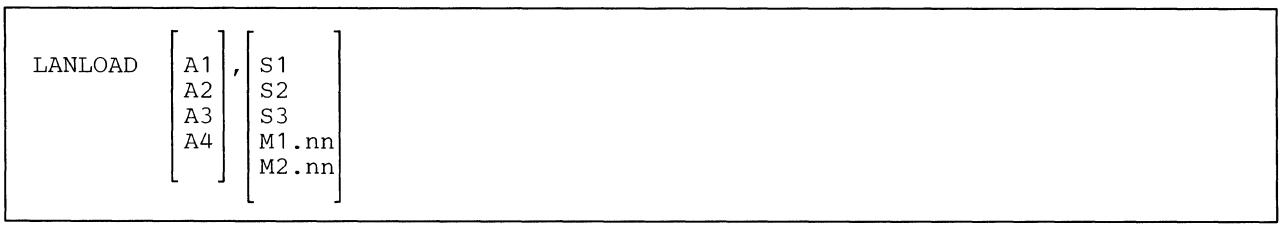
LANLOAD Procedure

The LANLOAD procedure creates a library named #LANLIB and copies the LAN communications support from diskette into that library and the system library, #LIBRARY. The LANLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the LANSAVE procedure. See the "LANSAVE Procedure" on page 4-254 for information about how to save the LAN communications support on diskette.

The LANLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the LANLOAD procedure to restore support that has been saved by LANSAVE.

If LAN is not currently on the system, you must use the TOLIBR procedure to copy the diskette file LAN into #LIBRARY before running LANLOAD.

If #LIBRARY is backed up with LAN on the system and then replaced before LANLOAD is run, you do not have to copy the diskette file LAN using the TOLIBR procedure.



S9020624-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #LANLIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #LANLIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

Example

Create #LANLIB on disk and copy the LAN communications support from diskette.

```
LANLOAD
```

LANSAVE Procedure

The LANSAVE procedure copies the LAN communications support to diskette. The LAN communications support from the libraries #LANLIB and #LIBRARY is copied. You should use the "LANLOAD Procedure" on page 4-253 to load the LAN communications support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPLAN and be located in diskette slot S1.

```
LANSAVE
```

S9020625-0

The LANSAVE procedure has no parameters.

Example

Copy the LAN communications support onto diskette.

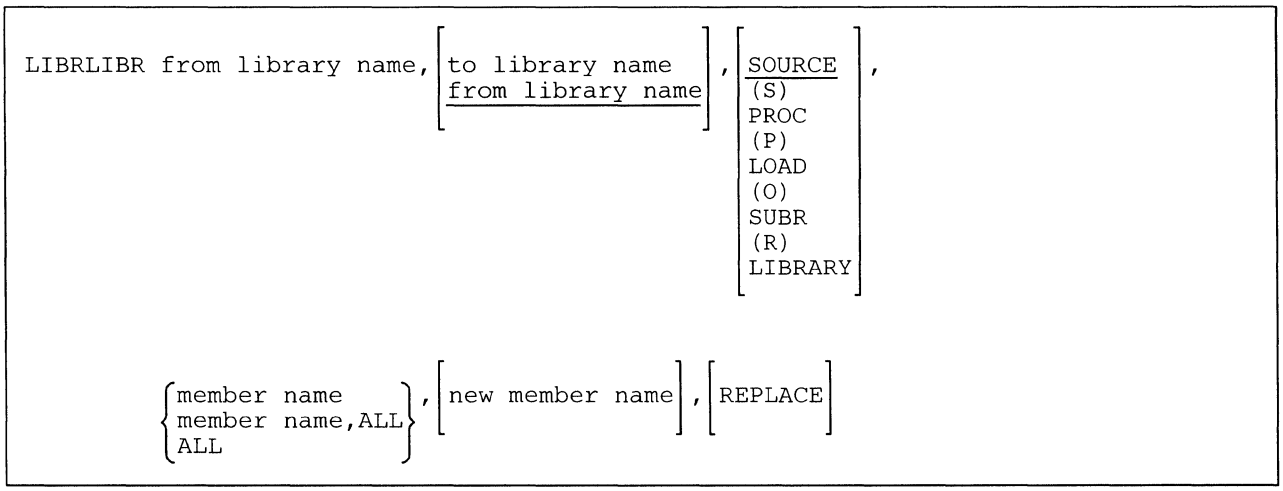
```
LANSAVE
```

LIBRLIBR Procedure

The LIBRLIBR procedure copies one or more library members from one library to another library. If you want to rename library members, see the "CHNGEMEM Procedure" on page 4-94.

The LIBRLIBR procedure cannot copy IBM-supplied library members. These library members are the IBM-supplied procedures, source members, and load members that come with the SSP and other program products. Library members that are not IBM-supplied members are library members you have created. To copy IBM-supplied library members, see "Copy Members from One Library to Another (LIBRLIBR Procedure)" on page A-64.

The LIBRLIBR procedure runs the $MAINT utility program.



S9020148-0

from library name specifies the library from which the one or more library members are to be copied. If no library name is specified, it is prompted for.

to library name specifies the library to which one or more members are to be copied. If no library name is specified, the from library name is assumed.

Note: If the from library name and the to library name parameters are the same, the member being copied must be given a new name.

SOURCE or S specifies that only source members are to be copied. If no parameter is specified, SOURCE is assumed.

PROC or P specifies that only procedure members are to be copied.

LOAD or O specifies that only load members are to be copied.

SUBR or R specifies that only subroutine members are to be copied.

LIBRARY specifies that all types of members (SOURCE, PROC, LOAD, and SUBR) are to be copied.

member name specifies the library member to be copied.

member name,ALL specifies that all member names beginning with the characters specified by member name are to be copied. Up to 7 characters can be specified for the member name.

ALL specifies that all members are to be copied. ALL is not allowed if the from library and the to library are the same.

new member name specifies the new member names of the copied members. The new name must be specified if the from library and the to library are the same. If **member name,ALL** was specified, the new member name must contain the same number of characters as the member name. The remaining characters in each member name are not changed.

If a new member name is not specified, the names of the members are not changed.

REPLACE specifies that if the *to library* contains a library member with the same name as the member being copied, the member being copied is to replace the existing member.

If REPLACE is not specified, members are placed in the library until a member is found with the same name as the member to be copied. When this occurs, a message is displayed and the operator can either continue processing (and replace the existing member) or cancel the procedure. If the operator continues processing and other duplicates are found during the job, the existing members are automatically replaced and no further messages are displayed indicating the duplicate members.

Example 1

Copy the source member named TEST from a library named MYLIB into a library named YOURLIB.

```
LIBRLIBR MYLIB,YOURLIB,,TEST
```

Example 2

Copy all the library members named TEST from a library named MYLIB into a library named YOURLIB, and replace any members named TEST in YOURLIB.

```
LIBRLIBR MYLIB,YOURLIB,LIBRARY,TEST,,REPLACE
```

Example 3

Copy all the procedures beginning with PAY from a library named TESTLIB to a library named PAYLIB. The members being copied are to replace any existing procedure members in PAYLIB that begin with PAY.

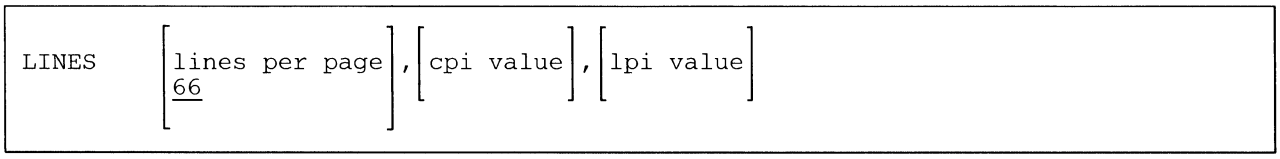
```
LIBRLIBR TESTLIB,PAYLIB,P,PAY,ALL,,REPLACE
```

LINES Procedure

The LINES procedure changes the number of lines per page, the horizontal print density (characters per inch), and the vertical print density (lines per inch) for printed output from a display station session. A job placed on the job queue uses the lines per page and the characters per inch values that were in effect when the job was placed on the queue.

You can also use the PRINT procedure to change these settings, see the "PRINT Procedure" on page 4-350.

The LINES procedure processes a FORMS OCL statement.



S9020149-0

lines per page specifies the number of lines per page. The maximum number of lines that can be specified is 112. If no value is specified, 66 is assumed. The lines per page specification remains in effect until either the session ends or a new lines per page value is specified by:

- Another LINES procedure
- The PRINT procedure
- A FORMS OCL statement
- A PRINTER OCL statement (for that print file only)
- The SET procedure or the $SETCF utility program

For SSP utility programs, the printer skips to a new page when six less than the number of lines specified are printed. For example, if 66 is specified, the printer skips to a new page after printing line 60. Also, the printing starts on line 6 of the new page. Therefore, if you specify 66, you really get 54 (66 - 12 = 54) lines per page. If LINES 13 is specified, one line is printed per page. When 12 or less lines are specified, the printing occurs on every line of each page.

For print operations from your programs, the SSP indicates an overflow condition when six less than the number of lines specified are printed.

cpi value specifies the horizontal print density (that is, characters per inch) to be used for printed output from the display station session. The values that can be specified are 10 or 15. The printer being used must support the horizontal print density specified. If the value specified is not supported by the printer, a message is displayed and you can continue or cancel the job. If no characters per inch value is specified, the horizontal print density is not changed.

The characters per inch value remains in effect until either the session ends or a new value is specified by:

- Another LINES procedure
- The PRINT procedure
- A FORMS OCL statement
- A PRINTER OCL statement (for that print step only)

If the switch option (0, zero) was specified for the characters per inch value when the printer was configured, this parameter is ignored.

lpi value specifies the vertical print density (lines per inch) to be used for printed output from the display station session. The values that can be specified are 4, 6, and 8. If the output is printed on a printer that does not support vertical print density, the lines per inch value is ignored.

The lines per inch value remains in effect until either the session ends or a new lines per inch value is specified by:

- Another LINES procedure
- The PRINT procedure
- A FORMS OCL statement
- A PRINTER OCL statement (for that print step only)

If no lines per inch value is specified and it was not previously set during the session, the system uses the value that was set during work station configuration (either 4, 6, or 8). If no lines per inch value was specified during work station configuration, the system uses a value of 6.

If the switch option (0, zero) was specified for the lines per inch value when the printer was configured, this parameter is ignored.

Example 1

Change the number of lines per page to 33.

```
LINES 33
```

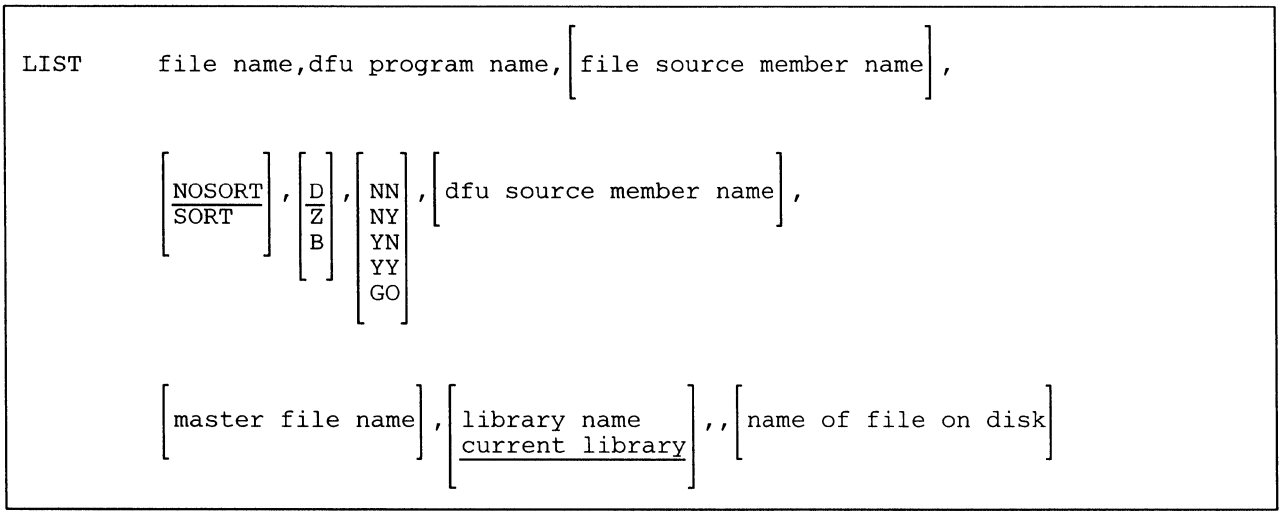
Example 2

This example shows how to change the lines printed per page to 88, the horizontal print density to 15 characters per inch, and the vertical print density to 8 lines per inch. The paper you want to use is 11 inches high (11 x 8 = 88 lines per page).

```
LINES 88,15,8
```

LIST Procedure

The LIST procedure allows you to print and also sort a disk file using the data file utility (DFU). File specifications are used to define the format of the records in the file. For more information about DFU, see the *DFU Guide*.



S9020150-0

file name specifies the file to be printed. The file name can be from 1 through 8 alphameric characters.

dfu program name specifies the DFU program to be used to process the file. If the program does not exist in the library, DFU starts the setup procedures. If the program exists in the library, DFU goes directly to listing the file.

file source member name specifies the source member containing file description (F-specifications) and record input specifications (I-specifications) that describe the file to be processed. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are taken as the data description.

This parameter is required if the specified format description does not exist.

NOSORT specifies that the file is not to be sorted before it is printed. If no parameter is specified, NOSORT is assumed.

SORT specifies that the file is to be sorted before it is printed.

D, Z, or B indicates whether unkeyed numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are **D**, **Z**, or **B**. If no parameter is specified, D is assumed.

D or B specifies a data file with blank fill of unkeyed numeric fields.

Z specifies a data file with zero fill of unkeyed numeric fields.

NN, NY, YN, YY, or GO specifies how the source specifications are to be processed. This 2-character parameter is used if the DFU program does not exist and the program setup must be performed. It indicates whether the DFU specifications for this program are already stored in the library, and whether they are to be stored in the library when the setup is complete. This source member of DFU specifications is stored or looked for in the current library, unless a library name is specified.

NN Stored DFU specifications are not used for this program setup. You are prompted to create the specifications but they are not saved in the library.

NY Stored DFU specifications are not used for this program setup. You are prompted to create the specifications; once created, they are stored in the library.

YN Stored DFU specifications are used for this program setup. You are not prompted, but can update the specifications before the format description is created. If you change the specifications at the display station, only this program is affected; the changes are not stored in the library.

YY Stored DFU specifications are used for this program setup. You are not prompted, but can update the specifications before the format description is created. The specifications in the library are replaced by the updated specifications.

GO Stored DFU specifications are used for this program setup. You are not prompted and cannot update the specifications before the program is created unless errors are found. If errors are found in the specifications, you must correct the specification before the program is created. However, the stored DFU specifications are not changed.

dfu source member name specifies the source member that contains or will contain saved DFU specifications. This parameter is required if the DFU source processing parameter is not NN.

master file name specifies the indexed file that contains information related to the file. See the *DFU Guide* for more information about this file.

library name specifies the library that contains or will contain the DFU specifications. All library members associated with the DFU program are looked for, or stored, in this library. If this parameter is not specified, the current library is assumed.

name of file on disk specifies the name in the disk VTOC of the file to be printed, if it is different from the name specified in the DFU program. If this parameter is specified, you can have several programs that refer to different, logical files print the same file actually on disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file to be printed by the DFU program, you are prompted for that parameter.

Example

This example shows how to print a disk file named FILE1. The DFU source specifications to be used to display the records is named DFUSETUP. The specifications are in the current library.

```
LIST FILE1,DFIL1FMT,,,,GO,DFUSETUP
```

LISTDATA Procedure

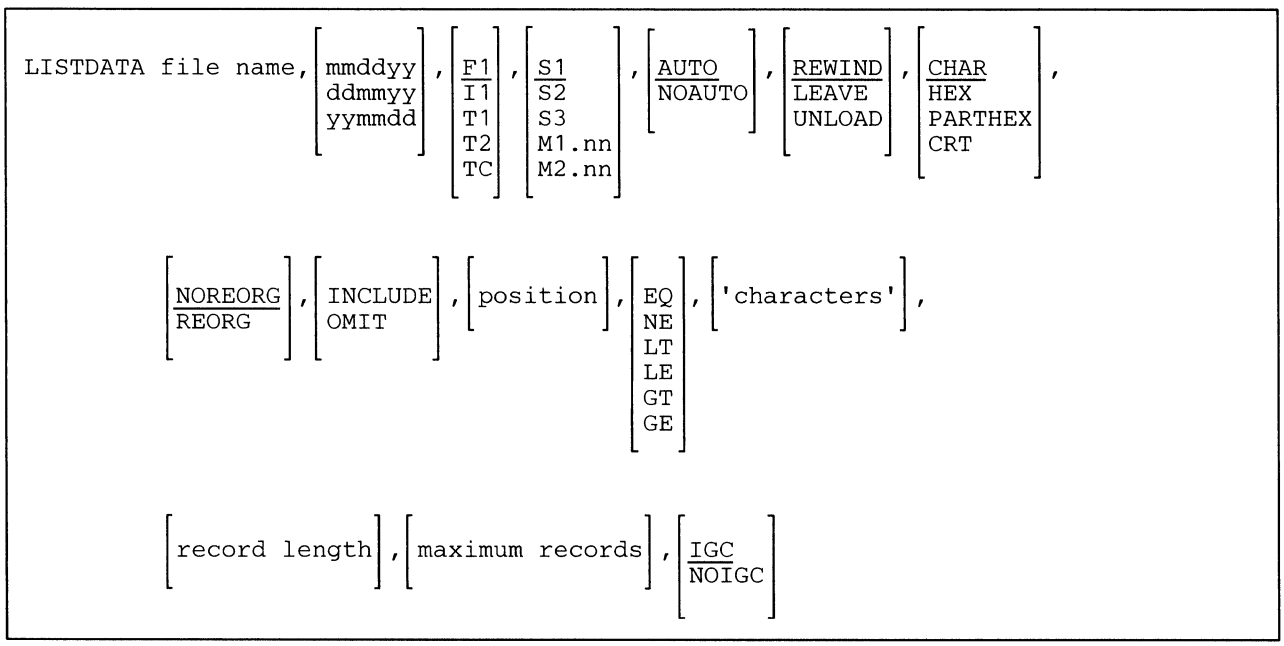
The LISTDATA procedure lists, on the system list device assigned to the display station, all or part of a disk, tape, tape cartridge, or uncompressed diskette file copied using the $COPY utility. During the list operation, the LISTDATA procedure can omit or include specific records. If you try to print with ideographic headings at a nonideographic printer, blanks are printed in place of the IGC characters.

For information about assigning the system list device, see the "SYSLIST Procedure" on page 4-498. When the system list device is the display screen (CRT), the information is displayed as if you had entered the CRT parameter.

The LISTDATA procedure can process a file that is being used by another job on the system, if that job's FILE statement specifies DISP-SHRRR or DISP-SHRRM; see the "FILE OCL Statement (for Disk Files)" on page 5-32 for more information about the DISP parameter.

The LISTDATA procedure can be used to list the contents of disk files on a remote system. See the *Distributed Data Management Guide*.

The LISTDATA procedure runs the $COPY utility program.



S9020151-1

file name specifies the file to be listed. The file can be of any file organization.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be listed. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If a date is not specified and more than one file exists with the same file name, the most recent file created is listed.

F1 specifies that the file to be listed is on disk. If no parameter is entered, F1 is assumed.

I1 specifies that the file to be listed is on diskette.

T1, T2, or TC specifies that the file to be listed is on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

S1, S2, or S3 specifies the diskette slot containing the first diskette of the file to be processed. If no parameter is specified, S1 is assumed. Parameter 4 is invalid if parameter 3 is T1, T2, TC, or F1.

M1.nn or M2.nn specifies the magazine location containing the first diskette of the file to be processed. M1 indicates the first magazine; and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. Parameter 4 is invalid if parameter 3 is T1, T2, TC, or F1.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original tape drive is used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape or tape cartridge should be rewound after the LISTDATA procedure has run. REWIND is assumed if this parameter is not specified.

LEAVE specifies that the tape or tape cartridge should be left where it is after the LISTDATA procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the LISTDATA procedure has run. If UNLOAD and TC are specified, the tape will be rewound after processing.

CHAR, PARTHEX, HEX, or CRT specifies that the records are to be listed on the current system list device. If the current system list device is a printer, these parameters also specify how the records are to be printed. If the current system list device is the display station (CRT), the records are displayed as if the CRT parameter had been specified; the CHAR, PARTHEX, and HEX specifications are ignored because the CRT specification allows you to select these functions by using command keys. You can use the STATUS SESSION command to determine your current system list device, and you can use the SYSLIST procedure to change your system list device.

Normally, 100 bytes of each record are printed per line. For sequential and direct files, the relative record number is printed with each record. For indexed files, the index key is printed with each record if REORG is specified; otherwise, the indexed file records are printed the same as sequential or direct files.

CHAR specifies that if the current system list device is a printer, only the printable characters in the record are printed. If no parameter is specified, CHAR is assumed. Any characters that are not printable are printed as blanks. For example, the following record contains an unprintable character in the third position.

```
12 4567
```

PARTHEX specifies that if the current system list device is a printer, the printable characters are printed and only the hexadecimal representations of the unprintable characters are printed. For example, the following record contains an unprintable character in the third position. That character's value is hex FA:

```
12 4567
  F
  A
```

HEX specifies that if the current system list device is a printer, the printable characters are printed and the hexadecimal representations of all characters are printed. For example, the following record contains an unprintable character in the third position. That character's value is hex FA:

```
12 4567
FFFFFFF
12A4567
```

CRT specifies that (no matter what the current system list device is) the records are to be displayed at the display station. If CRT is specified for a job placed on the job queue, the records are listed on the session printer.

The records can be displayed in both character format and hexadecimal format.

REORG specifies that if the file to be listed is an indexed file, the records are to be listed sequentially by key. (Deleted records are never listed regardless of the REORG or NOREORG parameter.) If REORG is specified, the system displays or prints the key above each record. Packed keys are expanded and noncontiguous keys are joined together before they are listed. If the key is displayed, the system truncates it after 70 characters. If a packed key that has been expanded is printed, the system truncates it after 120 characters.

NOREORG specifies that the records in the file are to be listed in the position they occur in the file. (Deleted records are never listed regardless of the REORG or NOREORG parameter.)

INCLUDE or OMIT specifies whether specific records in the file are to be included in or omitted from the listing. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If only 'characters' or position is specified, INCLUDE and EQ are assumed.

position specifies, for each record, the first character to be compared with the comparison characters. The position can be any number from 1 through 4096. If a position is not specified, every position in the record is compared with the comparison characters until the condition is met.

EQ specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is to be included in or omitted from the listing.

If only 'characters' or position is specified, INCLUDE and EQ are assumed.

NE specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record is to be included in or omitted from the listing.

LT specifies that if the characters in the record indicated by position are less than the comparison characters, the record is to be included in or omitted from the listing.

LE specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record is to be included in or omitted from the listing.

GT specifies that if the characters in the record indicated by position are greater than the comparison characters, the record is to be included in or omitted from the listing.

GE specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record is to be included in or omitted from the listing.

'characters' specifies the comparison characters. Up to 30 characters can be specified, and they must be enclosed by apostrophes ('). Blanks and commas (,) may be included but apostrophes cannot be included as data.

record length specifies the length of the listed records, and can be any number from 1 through 4096. If this parameter is not entered, the entire record is listed. This parameter allows you to list only a portion of the records in the file.

If the record length of the file to be listed is less than the specified record length, the additional record positions in the listing are filled with blanks. If the record length of the file to be listed is greater than the specified record length, the extra positions in the file are truncated when the file is listed.

maximum records specifies the total number of records to be listed when you do not want the entire file listed. Any number greater than 0 can be entered. This parameter is ignored when the entries are being shown at a display station.

IGC specifies that the file might contain ideographic characters and, if possible, LISTDATA should display those characters. If no parameter is specified, IGC is assumed. However, this parameter is used only for systems with the ideographic version of the SSP and is ignored for nonideographic systems.

Notes:

1. *When a file is displayed on a system with the ideographic version of the SSP and IGC is specified or defaulted to, all occurrences of data bytes with a hexadecimal value of OE will be interpreted as shift-out characters and all occurrences of data bytes with a hexadecimal value of OF will be interpreted as shift-in characters. An attempt will be made to interpret all pairs of data bytes in between such OE and OF data bytes as ideographic characters. To avoid misinterpretation, care should be exercised when displaying the contents of a file that include data bytes with hexadecimal values of OE or OF, which are not intended to be shift-out or shift-in characters.*

2. *When nonideographic characters are printed, 100 characters are printed on each line. The characters are printed in print positions 1 through 100. When a mixture of ideographic and nonideographic characters are printed, the number of characters printed on a line is variable. Each ideographic character represents 2 bytes of information and requires 2 print positions on the line. If an ideographic character begins in print position 100, position 100 is left blank and the character is printed in positions 0 and 1 on the next line. (Position 0 is placed one position to the left of the normal starting position, which is position 1.)*

NOIGC specifies that the file contents should be displayed just as they would be for a nonideographic system. Any ideographic data in the file is treated as 1-byte alphanumeric characters.

Example 1

Print only the printable characters in all the records of the file named FILE1.

```
LISTDATA FILE1,,F1
```

Example 2

Display all the records in the file named FILE2, regardless of the current system list device.

```
LISTDATA FILE2,,,,,,CRT
```

Example 3

Print all the records in the diskette file named FILE1. Both the characters in the records and their hexadecimal representations are to be printed. The diskette is in slot 1.

```
LISTDATA FILE1,,I1,S1,NOAUTO,,HEX
```

Example 4

Print only the records in the disk file named FILE3 that have the character X in position 5 of the record. Only the printable characters in the records are to be printed.

```
LISTDATA FILE3,,,,,,,,INCLUDE,5,EQ,'X'
```

Example 5

Display a file named FILE4 from the reel of the tape mounted on tape drive T1. When the display is finished, the tape reel is to be rewound to the beginning of the tape.

```
LISTDATA FILE4,,T1,,,REWIND,CRT
```

Example 6

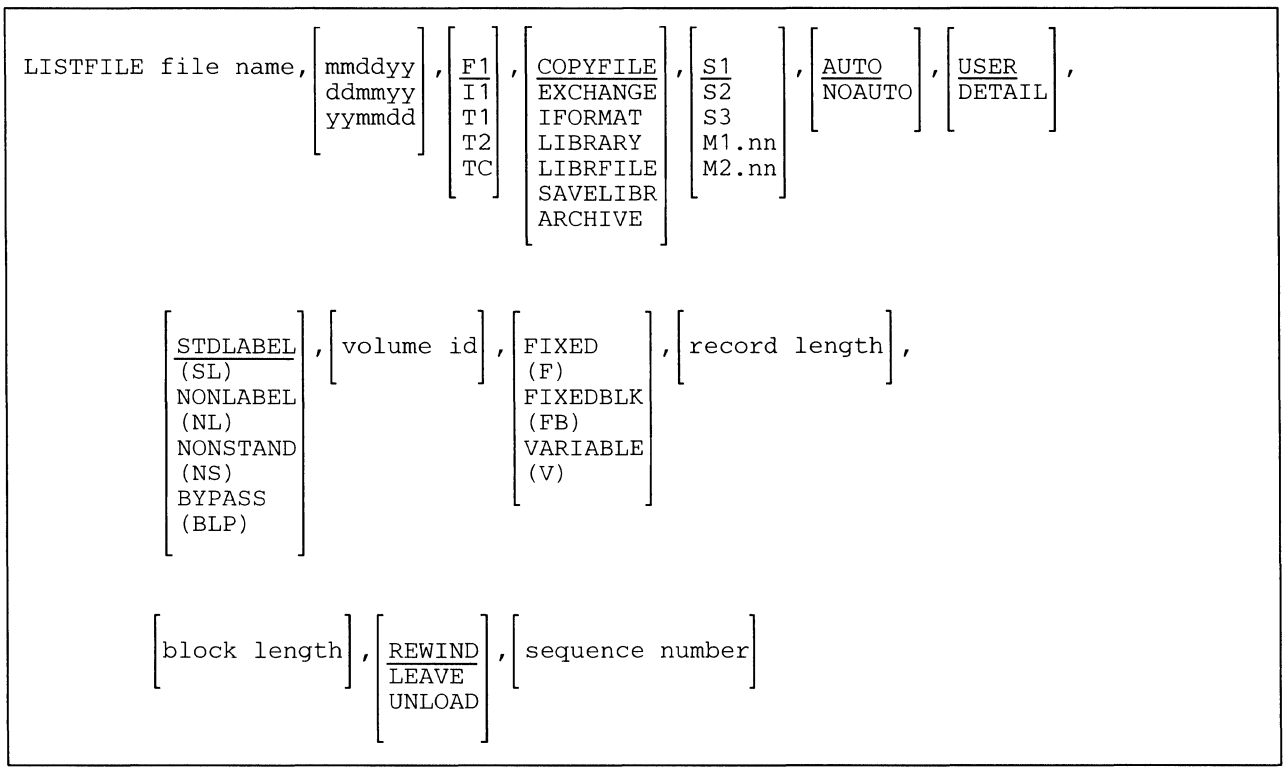
Display the contents of a tape cartridge file named DATA1 on the system list device.

```
LISTDATA DATA1,,TC
```

LISTFILE Procedure

The LISTFILE procedure lists the contents of a specified file, library, or archived folder member on the system list device assigned to the requester display station. If you try to print with ideographic headings at a nonideographic printer, blanks are printed in place of the ideographic characters. For information about assigning the system list device, see the "SYSLIST Procedure" on page 4-498.

The LISTFILE procedure runs the $BICR, $COPY, $MAINT, $TCOPY, or $TMSERV utility program, depending on the parameters entered.



S9020152-2

file name specifies the file to be listed.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be listed. For a disk file, the specified date must be in the same format as the session date. If the creation date for a disk file is not specified, and more than one file exists with the specified file name, the file with the most recent creation date is assumed.

For a diskette, tape, or tape cartridge file, the specified date must be in the same format as the creation date of the diskette, tape, or tape cartridge file. If the creation date for a diskette, tape, or tape cartridge file is not specified, and more than one file exists with the specified file name, the first file in the diskette VTOC or on the tape is processed.

F1 specifies that a file on disk is to be listed. If no parameter is specified, and LIBRARY, LIBRFILE, or COPYFILE is specified, F1 is assumed.

I1 specifies that a file on a diskette is to be listed. If no parameter is specified, and ARCHIVE, EXCHANGE, IFORMAT, or SAVELIBR is specified, I1 is assumed.

T1, T2, or TC specifies that a file on tape is to be listed. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

COPYFILE specifies that a data file on disk, diskette, tape, or tape cartridge is to be listed. The file must have been created by the $COPY utility (the SAVE procedure). If no parameter is specified, COPYFILE is assumed.

EXCHANGE specifies that a basic data exchange diskette file or tape exchange file is to be listed. See the "TRANSFER Procedure" on page 4-542 for information about basic data exchange files. If EXCHANGE is specified, F1 and TC are not allowed.

IFORMAT specifies that an I-exchange diskette file is to be listed. See the "TRANSFER Procedure" on page 4-542 for information about I-exchange files. If IFORMAT is specified, F1, T1, T2, and TC are not allowed.

LIBRARY specifies that a library on disk is to be listed. The file name specifies the name of the library. The listing is in the same format as a listing produced by the following LISTLIBR procedure:

```
LISTLIBR DIR,LIBRARY,file name
```

If LIBRARY is specified, I1, T1, T2, and TC are not allowed.

LIBRFILE specifies that a disk, diskette, tape, or tape cartridge file created by the FROMLIBR procedure is to be listed. The file may contain one or more library members. If the file was created with a record mode format, the members are limited to source members and procedure members.

The listing is the same as the listing produced by the LISTLIBR procedure. See the "LISTLIBR Procedure" on page 4-272 for more information.

SAVELIBR specifies that a library that was saved on diskette, tape, or tape cartridge with the SAVELIBR procedure is to be listed. If SAVELIBR is specified, F1 is not allowed.

ARCHIVE specifies that a folder member on diskette, tape, or tape cartridge is to be listed. If ARCHIVE is specified, F1 is not allowed.

S1, S2, or S3 specifies the diskette slot containing the first diskette of the file to be processed. If no parameter is specified, S1 is assumed. This parameter is invalid if F1, T1, T2, or TC is specified.

M1.nn or M2.nn specifies the magazine location containing the first diskette of the file to be processed. M1 indicates the first magazine; and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is invalid if F1, T1, T2, or TC is specified.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive is used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only the specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

USER or DETAIL specifies the type of information to be listed about each member. USER specifies general information about the member; DETAIL specifies more detailed information about the member. If no parameter is specified, USER is assumed. This parameter is allowed only when LIBRARY, LIBRFILE, or SAVELIBR are specified. For more information about the information listed by the USER and DETAIL parameters, see the "LISTLIBR Procedure" on page 4-272.

STDLABEL or SL specifies that the tape or tape cartridge to be processed has standard tape labels and all information needed to process the file can be taken from the tape file label. If this parameter is not specified, STDLABEL is assumed. If TC is specified for parameter 3, either this parameter must be left blank or STDLABEL (SL) must be specified.

NONLABEL or NL specifies that the tape to be processed does not have any labels and all information needed to process the file must be supplied. NONLABEL (NL) is invalid if TC was specified for parameter 3.

NONSTAND or NS specifies that the tape to be processed has nonstandard labels and all information needed to process the tape must be supplied. NONSTAND (NS) is invalid if TC was specified for parameter 3.

BYPASS or BLP specifies that the tape to be processed has standard labels and that label processing is to be bypassed. All information needed to process the file must be supplied. BYPASS (BLP) is invalid if TC was specified for parameter 3.

volume id specifies the volume identification of the tape or tape cartridge to be processed and can be 1 to 6 alphameric characters. It is used to check that the correct tape volume is mounted. If the correct tape is not mounted, an error message is issued. You can continue and process the file, or retry after mounting the correct tape, or cancel. This parameter is not allowed if NONLABEL, NONSTAND, or BYPASS is specified. If a multivolume tape is being listed, only the volume ID of the first tape is checked.

FIXED or F specifies that the record format of the tape file to be processed is fixed length, unblocked records. If the record format is not specified and EXCHANGE is specified, FIXED is assumed. FIXED (F) is invalid if TC was specified.

FIXEDBLK or FB specifies that the record format of the tape or tape cartridge file to be processed is fixed length, blocked records. If the record format is not specified, and the file is a COPYFILE, LIBRFILE, or SAVELIBR file, FIXEDBLK is assumed. If TC is specified, either this parameter must be left blank or FB must be specified.

VARIABLE or V specifies that the record format of the tape file to be processed is variable length, unblocked records. VARIABLE (V) is invalid if TC was specified.

Note: If the type of processing is STDLABEL and the record format specified (or defaulted to) does not match the record format in the tape file label, an error message is issued.

record length specifies the number of bytes in a logical tape or tape cartridge record. For variable length records this is the maximum length. The value specified can be from 18 to 4096 bytes. The record length parameter is required if NONLABEL, NONSTAND, or BYPASS is specified. If STDLABEL is specified and the record length is not, the record length is taken from the tape or tape cartridge file label. If both the record length and STDLABEL are specified, the record length specified must be the same as the record length in the tape or tape cartridge file label. If the record length is specified when listing a COPYFILE, LIBRFILE or SAVELIBR file, it must be 256.

block length specifies the number of bytes in a physical block of data of the tape or tape cartridge file. The value specified can be from 18 to 32767 bytes. This parameter is required if FIXEDBLK was specified and NONLABEL, NONSTAND, or BYPASS was specified. If FIXEDBLK and STDLABEL are specified, and the block length is not, the block length is taken from the tape or tape cartridge file label. If the block length is specified and both FIXEDBLK and STDLABEL are specified, the block length must be the same as the block length in the tape file label. This parameter is ignored if FIXED or VARIABLE is specified. If this parameter is specified when you list a COPYFILE or SAVELIBR file, it must be 24576. If this parameter is specified when you list a LIBRFILE, it must be 4096.

REWIND specifies that the tape or tape cartridge should be rewound after the LISTFILE procedure has run. If this parameter is not specified, REWIND is assumed.

LEAVE specifies that the tape or tape cartridge should be left where it is after the LISTFILE procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the LISTFILE procedure has run. If UNLOAD and TC are specified, the tape will be rewound after processing.

sequence number specifies which file to process by its position on the tape or tape cartridge. You can specify a number from 1 through 9999. If this parameter is omitted, and NONLABEL, NONSTAND, or BYPASS is specified, a sequence number of 1 is assumed. If this parameter is omitted, and STDLABEL is specified, the file will be found by its file label. If a sequence number is specified and STDLABEL is specified, the file is located by sequence number first, then the file label is checked. If it is not the correct file, the file is found by its file label.

Example 1

List the contents of a disk file named PAY.

```
LISTFILE PAY,,F1
```

Example 2

List the members of a library that is named MYLIB and was saved on diskette.

```
LISTFILE MYLIB,,I1,SAVELIBR
```

Example 3

List a file named FILEA that contains several library members.

```
LISTFILE MYLIB,,I1,LIBRFILE
```

Example 4

List the exchange file that is named EXFILE on tape. The tape was a nonlabeled tape. The record format was of variable length, unblocked records. The file was the third file on the tape. After listing the file, leave the tape where it finished processing.

```
LISTFILE EXFILE,,T1,EXCHANGE,,,,NL,,V,132,1320,LEAVE,3
```

Example 5

List a SAVELIBR file named MYLIB from tape cartridge. The file had been saved using the SAVELIBR procedure.

```
LISTFILE MYLIB,,TC,SAVELIBR
```

LISTLIBR Procedure

The LISTLIBR procedure lists the contents of a library. You can list the following:

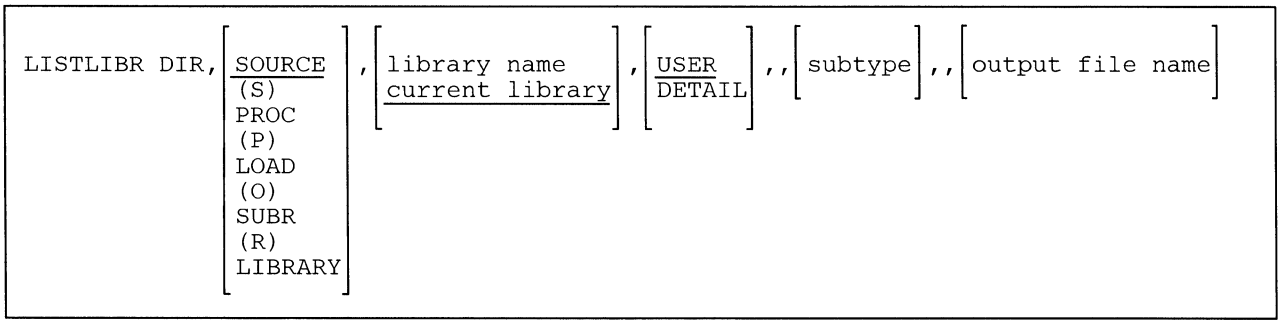
- Status information about a library. For example, the size of the library, the size of the directory, and how many diskettes are required to save the library (using the SAVELIBR procedure).
- The members contained in the library (the directory entries).
- The contents of the individual library members.
- The directory information for members by name, partial name, or subtype.

The listing can be directed to the system list device or to a disk file. If an output file name is specified, output is directed to that file; otherwise, output is directed to the system list device. For information about assigning the system list device, see the "PRINT Procedure" on page 4-350 or the "SYSLIST Procedure" on page 4-498.

The LISTLIBR procedure will not list IBM-supplied library members. These library members are the procedures, source members, and load members that come with the SSP or other program products (such as BASIC or RPG). Library members that are not IBM-supplied are members you have created. To list IBM-supplied members, see the "List Library Members and Information (LISTLIBR Procedure)" on page A-75.

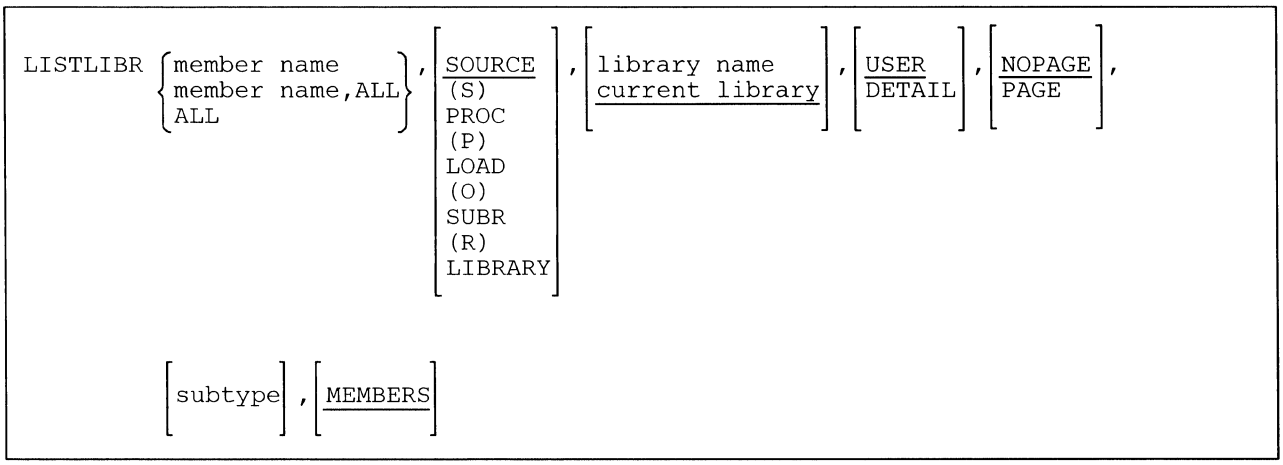
The LISTLIBR procedure runs the $MAINT utility program.

To list the names of each member and other information (the directory entries):



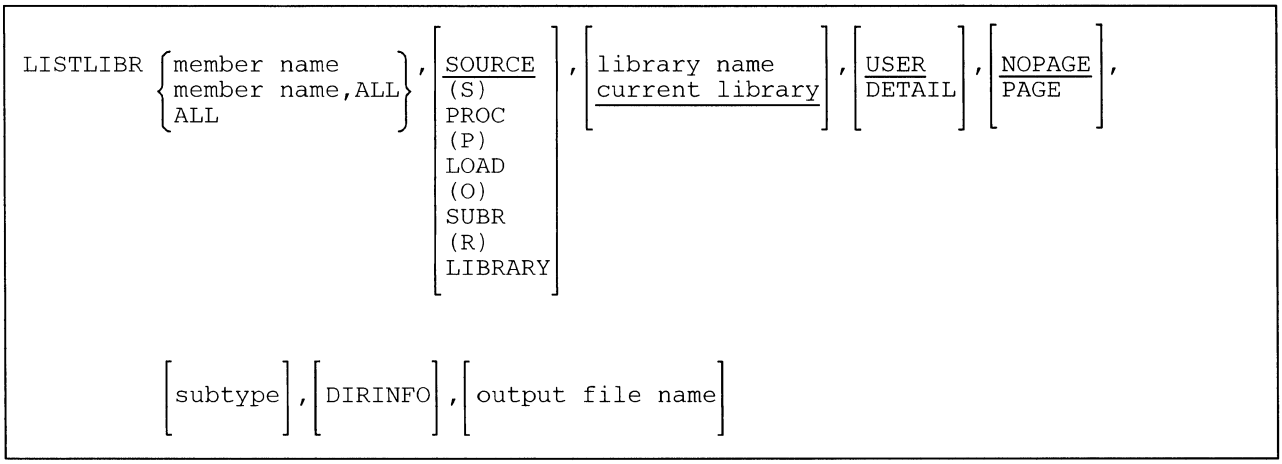
S9020153-1

To list the contents of one or more library members and their directory entries:



S9020154-0

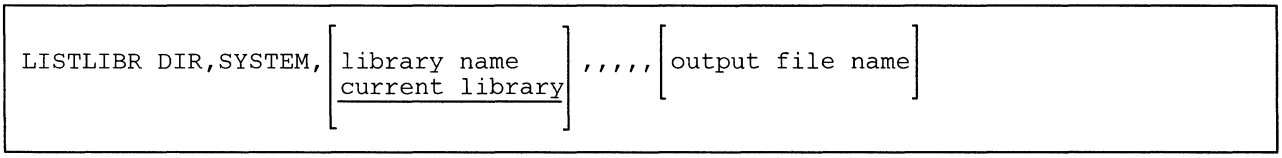
To list the directory information by name, partial name, or subtype:



S9020155-1

LISTLIBR

To list only the status information about a library:



S9020156-1

DIR specifies that the directory status information (if the second parameter is LIBRARY or SYSTEM) and the directory entries (if the second parameter is not SYSTEM) are to be listed.

member name specifies the library member to be listed.

member name,ALL specifies that the contents of the library members whose names begin with the specified characters (member name) are to be listed. Up to 7 characters can be specified for member name.

ALL specifies that all members of the library are to be listed.

SOURCE or S specifies that only library source members or directory entries are to be listed. If no parameter is specified, SOURCE is assumed.

PROC or P specifies that only the library procedure members or directory entries are to be listed.

LOAD or O specifies that only the library load members or directory entries are to be listed.

SUBR or R specifies that only the library subroutine members or directory entries are to be listed.

LIBRARY specifies that all library member types (SOURCE, PROC, LOAD, and SUBR) are to be listed.

SYSTEM specifies that only the status information for a library is to be listed. No library member names are listed. SYSTEM is valid only with DIR.

library name specifies the library containing the members to be listed. If a library name is not specified, the current library is assumed.

USER or DETAIL specifies the type of information to be listed about each member. USER specifies general information about the member; DETAIL specifies more detailed information about the member. If no parameter is specified, USER is assumed. For examples of the two types of listings, see Figure 4-13 on page 4-278 and Figure 4-14 on page 4-279.

output file name specifies the name of the disk file to which the output of the LISTLIBR procedure is to be written. The file name can have from one to eight alphameric characters and must not contain blanks, commas, or quotes. ALL is not a valid file name. If not specified, output is written to the SYSLIST device. The specified file is created as a sequential file on the local system. The record length is 132 bytes.

Note: A file or library with the specified name and the current date must not already exist on the disk.

NOPAGE or PAGE specifies, for source and procedure members only, whether each member is to be listed on a separate page. If no parameter is specified, NOPAGE is assumed. Load and subroutine members are always listed on separate pages.

NOPAGE specifies that as many members should be listed per page as can fit. You can use this option to save printer paper. See Figure 4-15 on page 4-280 for a sample of this listing.

PAGE specifies each member should be listed on a separate page.

subtype specifies that the members being listed are to be of the subtype specified. Valid subtypes are:

ARP RPG auto report member
ARS Automatic response member
ASM Assembler member
BAP BASIC procedure (source member)
BAS BASIC member
BGC Business graphics chart
BGD Business graphics data
BGF Business graphics format
COB COBOL member
CSM Communications and Systems Management member
CSP Cross-system product member
DFU Data file utility member
DLS Document library services member
DTA Data member
FMT Display format member
FOR FORTRAN member
ICF CNFIGICF procedure Interactive Communications Feature member
KEY KEYS procedure
MNU Menu member
MSG Message member
PHL Phone list member
QDE Query data entry member
QRY Query member
RPG RPG member
SRT Sort member
SSP CNFIGSSP procedure system configuration member
TXT Text member
UNS Unspecified
WSU Work station utility member
X25 X.25 packet switching link control

MEMBERS specifies that the actual contents of the specified members will be listed. If no parameter is specified, MEMBERS is assumed.

DIRINFO specifies that only the directory information for the specified members will be listed. Specifying DIRINFO without also specifying a member name is the same as specifying DIR as the first parameter.

Example 1

List the procedure member PAYROLL in the library called MYLIB.

```
LISTLIBR PAYROLL,PROC,MYLIB
```

Example 2

List all library members that have names beginning with PAY in the library named PAYLIB.

```
LISTLIBR PAY,ALL,LIBRARY,PAYLIB
```

Example 3

List status information and all the library directory entries for non-IBM-supplied members in the system library.

```
LISTLIBR DIR,LIBRARY,#LIBRARY
```

Example 4

List only the directory information for all library member types that begin with PAY in a library named PAYLIB.

```
LISTLIBR PAY,ALL,LIBRARY,PAYLIB,,,,DIRINFO
```

Example 5

List all the members of a library named PAYLIB that are of the subtype COB.

```
LISTLIBR ALL,LIBRARY,PAYLIB,,,COB
```

Example 6

List all the source members in the current library and direct the output to a file named MYLIST.

```
LISTLIBR DIR,,,,,,,MYLIST
```

Sample LISTLIBR Listings

The following show sample library listings. Following the sample listings is a description of the items on the listings.

```
LIBRARY - TEMLIB        STATUS              DATE  06/07/84      TIME  14.51


SIZE OF LIBRARY IN BLOCKS                                  250
SIZE OF DIRECTORY IN SECTORS                                75
USED LIBRARY MEMBER SECTORS                               2397
AVAILABLE MEMBER SECTORS                                    28
USED DIRECTORY ENTRIES                                     172
AVAILABLE DIRECTORY ENTRIES                                197
SIZE OF LIBRARY EXTENT IN BLOCKS                            50
USED EXTENT SECTORS                                        499
AVAILABLE EXTENT SECTORS                                     1


FIRST SECTOR OF LIBRARY (HEX)                         086967
LAST SECTOR OF LIBRARY (HEX)                          08732A
FIRST SECTOR OF DIRECTORY (HEX)                       086968
LAST SECTOR OF DIRECTORY (HEX)                        0869B1
FIRST SECTOR OF LIBRARY MEMBERS (HEX)                 0869B2
LAST SECTOR OF LIBRARY MEMBERS (HEX)                  08732A
NEXT AVAILABLE MEMBER SECTOR (HEX)                    08730F
FIRST SECTOR OF EXTENT (HEX)                          08732B

NUMBER OF DISKETTES REQUIRED TO SAVE LIBRARY-

                          TYPE 1 FORMAT (128)              4
                          TYPE 1 FORMAT2 (512)             3
                          TYPE 2D FORMAT (256)             1
                          TYPE 2D FORMAT2 (1024)           1
```

The number of available library blocks may be calculated from this line by dividing by 10.

These lines apply only when the library has an extent.

Figure 4-12. Library Status Listing

```
LIBRARY - TEMLIB      DIRECTORY          DATE  12/03/82     TIME  12.46

  NAME    TYPE  SUB TYPE    DATE     TIME   REF NUMBER          ATTRIBUTES          REL LEV  MRT MAX  PROGRAM SIZE/K BYTES  RECORD LENGTH  NUM STMTS  TOTAL SECTORS
--------  ----  --------  --------  -----  ----------  ---------------------------  -------  -------  --------------------  -------------  ---------  -------------
ALTCHR    O               03/26/82  15.58  000003                                   1                1                                                    2
CONCEPT   O     FMT       08/23/82  15.12  000001      FORMAT                       1                0                                                    9
CONCEPTS  O     FMT       08/24/82  14.47  000001      FORMAT                       1                0                                                   13
CONCMSG   O               10/14/82  14.36  000002      MSG                          1                0                                                    2
•
•
•
ALTCHR    P               10/14/82  16.01  000001      NOLOG                        1                                      120            4           1
ALTCHRBA  P               10/14/82  12.35  000001      NOLOG                        1                                      120            1           1
ASMEM     P     ASM       10/14/82  12.36  000001      NOLOG                        1                                      120            4           1
ASMM      P     ASM       10/14/82  12.38  000005      NOLOG                        1                                      120           11           2
CATLIST   P     DTA       10/24/82  10.39  000007      NOLOG                        1                                      120            9           2
•
•
•
ALTCHR    R               10/14/82  15.58  000003                                   1                                                                   3
CONCEPT   R     DFU       10/14/82  15.12  000001                                   1                                                                   3
CONCEPTS  R     BAS       10/14/82  14.55  000025                                   1                                                                  17
DFU       R     DFU       10/14/82  16.28  000001                                   1                                                                   3
EDITRECV  R               10/14/82  07.46  000001                                   1                                                                  12
EDITSCR   R               10/14/82  09.05  000001                                   1                                                                  68
•
•
•
A         S     DFU       10/14/82  16.22  000001                                   1                                       40            5           1
ALTCHR    S     ASM       10/14/82  15.55  000003                                   1                                       96           51           7
CONCEPT   S     RPG       10/14/82  15.02  000001                                   1                                       96            7           1
CONCEPTS  S     FMT       10/14/82  14.46  000001                                   1                                       80           96          10
CONCEPT1  S     DFU       10/14/82  15.12  000001                                   1                                       40            8           1
CONCMSG   S     MSG       10/14/82  14.36  000002      MSG                          1                                       96            6           2
```

Figure 4-13. Library Directory Listing, USER Parameter

```
LIBRARY - TEMLIB      DIRECTORY          DATE  12/03/82     TIME  12.48

  NAME    TYPE  SUB TYPE   DATE     TIME   REF NUMBER  ATTRIBUTES    REL LEV  MRT MAX  PROGRAM SIZE/K BYTES  NUM TEXT/ REC LNGTH  LINK ADDR/ NUM STMTS  RLD DISP  ENTRY ADDRESS  DISK ADDRESS  TOTAL SECTORS
--------  ----  --------  --------  -----  ----------  ------------  -------  -------  --------------------  -------------------  --------------------  --------  -------------  ------------  -------------
ALTCHR    0               03/26/82  15.58  000003      000000200000  1                 1                     02                   0000                  B9        0000           021660        2
CONCEPT   0     FMT       08/23/82  15.12  000001      100800201500  1                 0                     02                   0000                  00        0000           0217D6        9
CONCEPTS  0     FMT       08/24/82  14.47  000001      100800201500  1                 0                     02                   0000                  00        0000           02180B        13
CONCMSG   0               10/14/82  14.36  000002      000800200000  1                 0                                          0000                  00        0000           0217FF        2
•
•
•
ALTCHR    P               10/14/82  16.01  000001      400000200000  1                                       120                  4                                               021645        1
ALTCHRBA  P               10/14/82  12.35  000001      400000204000  1                                       120                  1                                               021646        1
ASMEM     P     ASM       10/14/82  12.36  000001      400000203100  1                                       120                  4                                               021647        1
ASMM      P     ASM       10/14/82  12.38  000005      400000203100  1                                       120                  11                                              02165B        2
CATLIST   P     DTA       10/24/82  10.39  000007      400000200200  1                                       120                  9                                               021755        2
COPYD     P     DTA       10/24/82  10.38  000008      400000200200  1                                       120                  18                                              0217CE        3
•
•
•
ALTCHR    R               10/14/82  15.58  000003      000000200000  1                                                            0000                  00        0000           02165D        3
CONCEPT   R     DFU       10/14/82  15.12  000001      000000201400  1                                                            0000                  00        0000           0217D3        3
CONCEPTS  R     BAS       10/14/82  14.55  000025      000000203200  1                                                            0000                  00        0000           0217EC        17
DFU       R     DFU       10/14/82  16.28  000001      000000201400  1                                                            0000                  00        0000           02176C        3
EDITRECV  R               10/14/82  07.46  000001      000000000000  1                                                            0000                  00        0000           0216AD        12
EDITSCR   R               10/14/82  09.05  000001      000000000000  1                                                            0000                  00        0000           021669        68
•
•
•
A         S     DFU       10/14/82  16.22  000001      000000001400  1                                       40                   5                                               021762        1
ALTCHR    S     ASM       10/14/82  15.55  000003      000000203100  1                                       96                   51                                              021531        7
CONCEPT   S     RPG       10/14/82  15.02  000001      000000203500  1                                       96                   7                                               0217D1        1
CONCEPTS  S     FMT       10/14/82  14.46  000001      000000201500  1                                       80                   96                                              021801        10
CONCEPT1  S     DFU       10/14/82  15.12  000001      000000001400  1                                       40                   8                                               0217D2        1
CONCMSG   S     MSG       10/14/82  14.36  000002      000000201700  1                                       96                   6                                               0217FD        2
```

Figure 4-14. Library Directory Listing, DETAIL Parameter

```
LIBRARY - TEMLIB            DATE  10/31/82     TIME  15.15          PAGE    1

                SUB                       REF                                  REL  MRT     PROGRAM    RECORD   NUM    TOTAL
  NAME    TYPE  TYPE    DATE     TIME    NUMBER          ATTRIBUTES             LEV  MAX  SIZE/K BYTES  LENGTH  STMTS  SECTORS
--------  ----  ----  --------  -----  ------  ------------------------------  ---  ---  ------------  ------  -----  -------

ALTCHR     P          10/14/82  16.01  000001  NOLOG                            1                        120      4      1

        // * 'LOADING ALTERNATE CHARACTERS (ALTCHR)'
        // LOAD ALTCHR
        // PRINTER NAME-ALTCHAR,CPI-15,LPI-8,SPOOL-YES
        // RUN

                SUB                       REF                                  REL  MRT     PROGRAM    RECORD   NUM    TOTAL
  NAME    TYPE  TYPE    DATE     TIME    NUMBER          ATTRIBUTES             LEV  MAX  SIZE/K BYTES  LENGTH  STMTS  SECTORS
--------  ----  ----  --------  -----  ------  ------------------------------  ---  ---  ------------  ------  -----  -------

ALTCHRBA   P          10/14/82  12.35  000001  NOLOG                            1                        120      1      1

        // PRINTER NAME-ALTCHAR,LINES-65,CPI-15,LPI-8

                SUB                       REF                                  REL  MRT     PROGRAM    RECORD   NUM    TOTAL
  NAME    TYPE  TYPE    DATE     TIME    NUMBER          ATTRIBUTES             LEV  MAX  SIZE/K BYTES  LENGTH  STMTS  SECTORS
--------  ----  ----  --------  -----  ------  ------------------------------  ---  ---  ------------  ------  -----  -------

ASMEM      P   ASM   10/14/82  12.36  000001  NOLOG                            1                        120      4      1

        // IF SUBR-'?1?,?2'?CLIB?'?' REMOVE ?1?,SUBR,?2?
        // IF LOAD-'?1?,?2'?CLIB?'?' REMOVE ?1?,LOAD,?2?
        ASM ?1?,?2?,?2?,MAC
        OLINK ?1?,?2?,?1?,?2?

                SUB                       REF                                  REL  MRT     PROGRAM    RECORD   NUM    TOTAL
  NAME    TYPE  TYPE    DATE     TIME    NUMBER          ATTRIBUTES             LEV  MAX  SIZE/K BYTES  LENGTH  STMTS  SECTORS
--------  ----  ----  --------  -----  ------  ------------------------------  ---  ---  ------------  ------  -----  -------

ASMM       P   ASM   10/14/82  12.38  000005  NOLOG                            1                        120     11      2

        // PRINTER              CONTINUE-YES,PRIORITY-0,SPOOL-YES
        // IF ?1?='' * 'Enter name of source member to compile'
        // IF SUBR-'?1R?,?2'?CLIB?'?' REMOVE ?1?,SUBR,?2?
        // IF LOAD-'?1?,?2'?CLIB?'?' REMOVE ?1?,LOAD,?2?
        ASM ?1?,?2?,?2?,MAC
        // IF ?CD?=1008 GOTO END
        OLINK ?1?,?2?,?1?,?2?
        // TAG END
        // PRINTER              CONTINUE-NO
        // IF DATAF1-TEMPLE DELETE TEMPLE,F1
        COPYPRT ,TEMPLE,CANCEL,CRT
```

Figure 4-15. Library Members Listing, NOPAGE Parameter

LIBRARY - SAMPLIB DIRECTORY DATE 12/30/83 TIME 15.10

NAME	TYPE	SUB TYPE	DATE	TIME	REF NUMBER	ATTRIBUTES	REL LEV	MRT MAX	PROGRAM SIZE/K BYTES	RECORD LENGTH	NUM STMTS	TOTAL SECTORS
F971	O		16/82/06	12.12		NONINQ	1		1			4
F971A	O		02/24/83	10.01	000015	FORMAT	1		0			9
F971B	O		09/22/82	17.31	000004		1		1			1
F980	O		12/03/82	07.26	000001	FORMAT	1		0			5
F982	O		09/17/82	13.56	000006		1		1			4
F985	O	COB	01/25/83	14.31	000001	FORMAT	1		0			3
F965A	R	FOR				SSP	2					5
F965B	R	FOR				SSP	2					2
FMTS	S	FMT	06/01/83	08.33	000009		1			96	24	4

Figure 4-16. Library Directory Listing, DIRINFO Parameter with Member Name F,ALL Specified

Description of the Items on the Library Directory Listing

NAME specifies the name of the library member.

TYPE specifies the type of library member:

O Load member
P Procedure member
R Subroutine member
S Source member

SUBTYPE specifies the subtype of the member:

ARP RPG auto report member
ARS Automatic response member
ASM Assembler member
BAP BASIC procedure (source member)
BAS BASIC member
BGD Business graphics data
BGF Business graphics format
COB COBOL member
CSM Communications and Systems Management member
CSP Cross-system product member
DFU Data file utility member
DLS Document library services member
DTA Data member
FMT Display format member
FOR FORTRAN member
ICF CNFIGICF procedure Interactive Communications Feature member
KEY KEYS procedure
MNU Menu member
MSG Message member
PHL Phone list member
QDE Query data entry member
QRY Query member
RPG RPG member
SRT Sort member
SSP CNFIGSSP procedure system configuration member
TXT Text member
UNS Unspecified
WSU Work station utility member
X25 X.25 packet switching link control

DATE specifies the date (in system format) the member was either created or last updated.

TIME specifies the time the member was either created or last updated.

REF NUMBER specifies the reference number assigned to the member.

ATTRIBUTES for USER parameter: One or more of the following identifiers may be listed with the library member:

CONFIG A CNFIGSSP configuration load member
FORMAT A display format member
IBM An IBM-supplied member
MENU A menu source or load member
MRT A multiple requester terminal program or procedure
NEP A never-ending program or procedure
NOLOG The procedure's OCL statements are not logged to the history file
NONINQ The load member program in noninquirable
PDATA The procedure passed data to a program, not parameters
SSP An SSP member

REL LEV is the release level of the SSP when the member was created or when the member was updated using SEU or SDA. For members copied from an IBM System/32 sector-mode file, the version level is 232. For members copied from an IBM System/34 sector-mode file, the version level is the release level of the IBM System/34.

MRT MAX is for load members (O) only. The maximum number of requester display stations that can be be attached to the program.

PROGRAM SIZE is for load members (O) only. The number of K bytes required to run the program.

RECORD LENGTH is for source (S) and procedure (P) members only. The length of the statements in the member.

NUM STMTS is for source (S) and procedure (P) members only. The number of statements in the member.

TOTAL SECTORS specifies the total number of sectors used by the member.

Differences For DETAIL Parameter Listings:

ATTRIBUTES for DETAIL parameter specifies 6 bytes of attributes or indicators giving detailed characteristics of the member. The indicators are listed in hexadecimal only.

First Byte:

Bit	Meaning when On (Set to 1)
0	This is an SSP member.
1	For a **load member** (O), the program can do privileged SVC (supervisor call) instructions. For a **procedure member** (P), the statements generated by the procedure are not logged in the history file.
2	The program is not inquirable.
3	For a **load member** (O), the member is a display format load member. For a **procedure member** (P), only data (not parameters) can be passed on the procedure command that calls the procedure.
4	This program requires that $WORK and $SOURCE work files be allocated. $SOURCE must be filled from the keyboard or a source member.
5	This SSP module is not part of the basic SSP system.
6	A PTF (program temporary fix) has been applied to this program.
7	This load member contains overlays.

Second Byte:

Bit	Meaning when On (Set to 1)
0	This is a dedicated program (the program cannot be run while other jobs are running).
1	This is an NEP (never-ending program).
2	This member has a cross-reference format index table. Valid for SSP load members only.
3	If password security is active, the program can only be run by an operator that has a security classification of system operator or higher. If password security is not active, the program can only be run from the system console.
4	This member cannot be run. (It cannot be loaded with a LOAD OCL statement.)
5	Indicates that the program contains a program common area. This program requires that a new load address be calculated at load time to ensure that the program is placed in main storage at a point beyond its own common region.
6	This program reads utility control statements.
7	This program contains a where-to-go table. It is used by the transient cross-reference program (#MAXRF) that resolves the label and addresses in the program.

Third Byte:

Bit	Meaning when On (Set to 1)
0	This program requires that $WORK2 be allocated.
1	This program is not swappable.
2	Indicates high-level dedication. This program cannot be run if any other jobs are running or if a user is signed on at any other display station.
3	This program requires the FORTRAN instruction set (the scientific instruction set).
4	This member is a configuration record. Created by either CNFIGSSP or CNFIGICF.
5	This member must be transferred to (the module should not be loaded using the LOAD OCL statement).
6	This display format load member is cross-referenced.
7	This MRT (multiple requester terminal) program requires a new copy of the MRT program to be loaded.

Fourth Byte:

Bit	Meaning when On (Set to 1)
0	This program requires the BASIC instruction set.
1	This member is an IBM-pad member.
2	Indicates whether the member is from this system (rather than from another IBM system, such as the IBM System/34).
3	This member is supplied by IBM. This bit is used to prevent IBM-supplied members from being deleted, copied, or listed.
4	This member is contained in a library extent.
5	Reserved.
6	This member is a system transient.
7	Only one copy of this program can be running.

Fifth Byte:

Value	Source Member Type
02	Data member
11	Automatic response member
12	RPG auto-report program specifications
13	BASIC procedure
14	Data file utility
15	Display format specifications
16	Menu
17	Message member
18	Phone list
19	Sort specifications
31	Assembler program
32	BASIC program
33	COBOL program
34	FORTRAN program
35	RPG program specifications
36	WSU program specifications
40	Type not specified
41	Business graphics chart
42	Business graphics data
43	Business graphics format
51	SSP-ICF configuration member
52	System configuration member
53	Editable text member
54	Free form text member
55	Hard copy text member

Value	Source Member Type
56	X.25 packet switching link control
57	Communications and Systems Management member
58	Query member
59	Cross-system product member
5A	Query data entry member
5B	Document library services member
5C	KEYS procedure

Sixth Byte:

Bit	Meaning when On (Set to 1)
0	Dynamically privileged module.
1	Member must be swapped.
2-7	Reserved.

NUM TEXT/REC LNGTH specifies the following:

- For **source** (S) and **procedure** (P) members, the length of the statements in the member.
- For **load members** (O), the number of text sectors contained in the member, excluding RLDs (relocation dictionaries), which are the parts of a load member that are used for adjusting main storage addresses when the member is moved to main storage.
- For **subroutine members** (R), the category value assigned when the subroutine was assembled.

LINK ADDR/NUM STMT for **load members** (O), the main storage address (in hexadecimal) assigned to the member when it is linked in main storage with other load members.

For **source members** (S) or **procedure members** (P), the number of statements in the member.

RLD DISP is for load members (O) only. The displacement (in hexadecimal) of the first RLD (relocation dictionary) in the first sector containing RLDs.

ENTRY ADDRESS is for load members (O) only. The main storage address (in hexadecimal) of the entry point of the member.

DISK ADDRESS specifies the address (in hexadecimal) of the first sector of the library member.

TOTAL SECTORS specifies the total number of sectors used to contain the library member.

When a USER listing is printed to the output file, the directory entry fields are in the following columns:

Beginning Column	Field Length	Item
1	8	NAME
12	1	TYPE
16	3	SUBTYPE
22	8	DATE
32	5	TIME
39	6	REF NUMBER
47	35	ATTRIBUTES (Columns 47, 54, 61, 68, 75) (Each attribute is 6 bytes.)
83	3	REL LEV
88	3	MRT MAX
98	3	PROGRAM SIZE/K BYTES
109	3	RECORD LENGTH
115	5	NUM STMTS
123	5	TOTAL SECTORS

When a DETAIL listing is printed to the output file, the directory entry fields are in the following columns:

Beginning Column	Field Length	Item
1	8	NAME
12	1	TYPE
16	3	SUBTYPE
21	8	DATE
31	5	TIME
37	6	REF NUMBER
44	12	ATTRIBUTES
57	3	REL LEV
62	3	MRT MAX
72	3	PROGRAM SIZE/K BYTES
84 85	3 2	REC LNGTH (P and S) NUM TEXT (O and R)
94 95	5 4	NUM STMTS (P and S) LINK ADDR (O and R)
104	3	RLD DISP
110	4	ENTRY ADDRESS
118	6	DISK ADDRESS
127	5	TOTAL SECTORS

When a library does not have an extent, the library status fields are on the following lines and in the following columns in the output file:

Line	Beginning Column	Field Length	Item
2	53	6	SIZE OF LIBRARY IN BLOCKS
3	53	6	SIZE OF DIRECTORY IN SECTORS
4	53	6	USED LIBRARY MEMBER SECTORS
5	53	6	AVAILABLE MEMBER SECTORS
6	53	6	USED DIRECTORY ENTRIES
7	53	6	AVAILABLE DIRECTORY ENTRIES
8	53	6	FIRST SECTOR OF LIBRARY (HEX)
9	53	6	LAST SECTOR OF LIBRARY (HEX)
10	53	6	FIRST SECTOR OF DIRECTORY (HEX)
11	53	6	LAST SECTOR OF DIRECTORY (HEX)
12	53	6	FIRST SECTOR OF LIBRARY MEMBERS (HEX)
13	53	6	LAST SECTOR OF LIBRARY MEMBERS (HEX)
14	53	6	NEXT AVAILABLE MEMBER SECTOR (HEX)
16	56	3	NUMBER OF TYPE 1 FORMAT DISKETTES
17	56	3	NUMBER OF TYPE 1 FORMAT2 DISKETTES
18	56	3	NUMBER OF TYPE 2D FORMAT DISKETTES
19	56	3	NUMBER OF TYPE 2D FORMAT2 DISKETTES

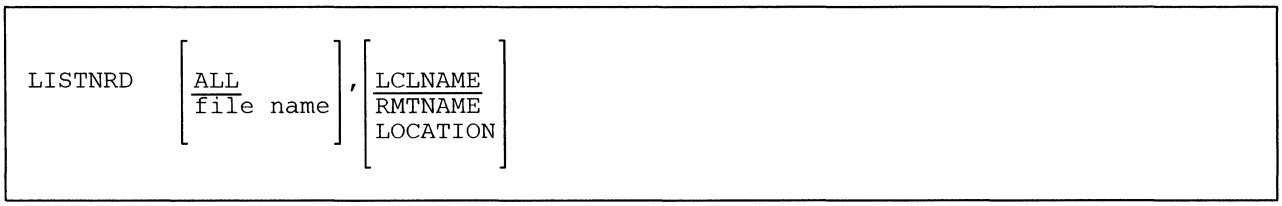
When a library has an extent, the library status fields apre on the following lines and in the following columns in the output file:

Line	Beginning Column	Field Length	Item
2	53	6	SIZE OF LIBRARY IN BLOCKS
3	53	6	SIZE OF DIRECTORY IN SECTORS
4	53	6	USED LIBRARY MEMBER SECTORS
5	53	6	AVAILABLE MEMBER SECTORS
6	53	6	USED DIRECTORY ENTRIES
7	53	6	AVAILABLE DIRECTORY ENTRIES
8	53	6	SIZE OF LIBRARY EXTENT IN BLOCKS
9	53	6	USED EXTENT SECTORS
10	53	6	AVAILABLE EXTENT SECTORS
11	53	6	FIRST SECTOR OF LIBRARY (HEX)
12	53	6	LAST SECTOR OF LIBRARY (HEX)
13	53	6	FIRST SECTOR OF DIRECTORY (HEX)
14	53	6	LAST SECTOR OF DIRECTORY (HEX)
15	53	6	FIRST SECTOR OF LIBRARY MEMBERS (HEX)
16	53	6	LAST SECTOR OF LIBRARY MEMBERS (HEX)
17	53	6	NEXT AVAILABLE MEMBER SECTOR (HEX)
18	53	6	FIRST SECTOR OF EXTENT (HEX)
20	56	3	NUMBER OF TYPE 1 FORMAT DISKETTES
21	56	3	NUMBER OF TYPE 1 FORMAT2 DISKETTES
22	56	3	NUMBER OF TYPE 2D FORMAT DISKETTES
23	56	3	NUMBER OF TYPE 2D FORMAT2 DISKETTES

LISTNRD Procedure

The LISTNRD procedure lists either a single entry or all entries from the network resource directory. The information will be listed on the system list device assigned to the requesting display station.

The LISTNRD procedure runs the $LABEL utility program.



S9020554-1

ALL specifies that all entries in the network resource directory are to be listed. If this parameter is not specified, ALL is assumed.

name specifies the local name of the file for which the network directory information is to be listed.

LCLNAME specifies that the entries are to be sorted and listed alphabetically by local name. This parameter is valid only if ALL is specified.

RMTNAME specifies that the entries are to be sorted and listed alphabetically by remote name. This parameter is valid only if ALL is specified.

LOCATION specifies that the entries are to be sorted and listed alphabetically by location name. This parameter is valid only if ALL is specified.

LOAD3601 Procedure

The LOAD3601 procedure is for the SSP-ICF Finance subsystem. The LOAD3601 procedure transmits the contents of a disk file from the System/36 to an IBM 3601 Controller. To run the LOAD3601 procedure, the 3601 Controller location must have been enabled to communicate with the system monitor session.

For more information on the LOAD3601 procedure and the SSP-ICF Finance subsystem, see the manual *Interactive Communications Feature: Finance Subsystem Reference.*

```
LOAD3601 file name,location name
```

S9020157-0

file name specifies the disk file that contains the diskette image to be transmitted to the controller.

location name specifies the location of the controller to which the diskette image is to be transmitted.

Example

To transmit the file named SETUP360 to a controller at location BANK1.

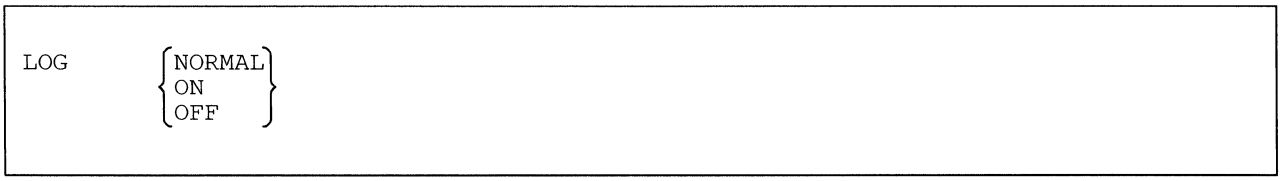
```
LOAD3601 SETUP360,BANK1
```

LOG Procedure

The LOG procedure indicates whether the OCL statements and utility control statements in a procedure are to be logged to the history file, regardless of the statement logging indicator in the procedure. This allows you to create your procedures with this indicator set to off, but still have the statements logged to the history file while you are debugging the procedure. The statements that start procedures and end of job information are always logged to the history file.

You can log the options from the system help support menus and from the user menus to the history file by entering LOG ON. This allows you to determine the options that an operator used. If you do not want to log the options, enter LOG NORMAL or LOG OFF.

The LOG statement processes a LOG OCL statement.



S9020158-0

NORMAL specifies that each procedure's logging indicator is to be used.

ON specifies that all OCL statements are to be logged to the history file, regardless of the procedure's logging indicator. Also, the menu options an operator took are logged.

OFF specifies that no OCL statements are to be logged to the history file, regardless of the procedure's logging indicator. Also, the menu options an operator took are not logged.

Example

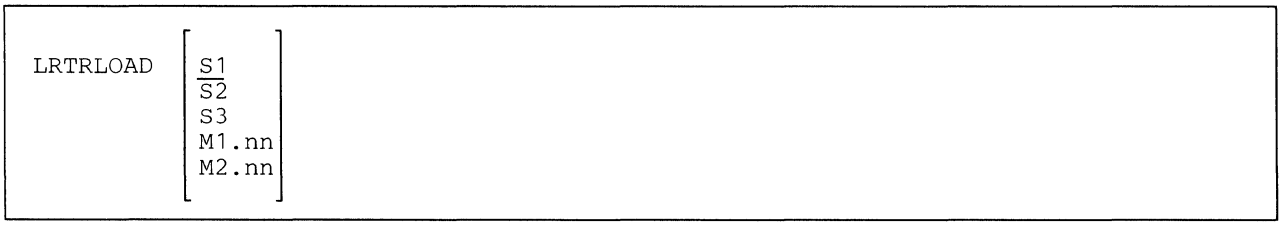
This example shows how to temporarily prevent the OCL statements in procedure PROCA from being logged to the history file.

```
LOG OFF
PROCA
LOG NORMAL
```

LRTRLOAD Procedure

The LRTRLOAD procedure copies the IBM Token-Ring Network from a backup diskette to the PC Support/36 library, #IWLIB, and system library, #LIBRARY. The LRTRLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the LRTRSAVE procedure. See the "LRTRSAVE Procedure" on page 4-296 for more information about how to save the IBM Token-Ring Network on diskette.

Note: The IBM Token-Ring Network requires PC Support/36 to be on the system to load the IBM Token-Ring Network. If PC Support/36 is not on the system, you will be prompted to load PC Support/36 before the IBM Token-Ring Network can be loaded.



S9020626-0

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

Example

Copy the IBM Token-Ring Network to #IWLIB and #LIBRARY from a backup diskette.

```
LRTRLOAD
```

LRTRSAVE Procedure

The LRTRSAVE procedure copies the IBM Token-Ring Network from the libraries #IWLIB and #LIBRARY onto diskette. The LRTRLOAD procedure should be used to load the IBM Token-Ring Network from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPIWS. The diskette must be in slot 1.

```
LRTRSAVE
```

S9020627-0

The LRTRSAVE procedure has no parameters.

Example

Copy the IBM Token-Ring Network onto diskette.

```
LRTRSAVE
```

MAINTX25 Procedure

The MAINTX25 procedure allows you to display information about the X.25 links and to alter certain logical channel and virtual circuit configuration information. The MAINTX25 procedure is described in the manual *Using System/36 Communications*.

```
MAINTX25
```

S9020159-0

The MAINTX25 procedure has no parameters.

Example

To run the MAINTX25 procedure:

```
MAINTX25
```

MCSCONV Procedure

The MCSCONV procedure allows you to convert library source members, library procedure members, data files, text folders, and data dictionaries from the national language version (NLV) hexadecimal representation to the multinational character set (MCS) value, and vice versa. The conversion procedures are described in Appendix D, "Multinational Character Set Conversion Utility."

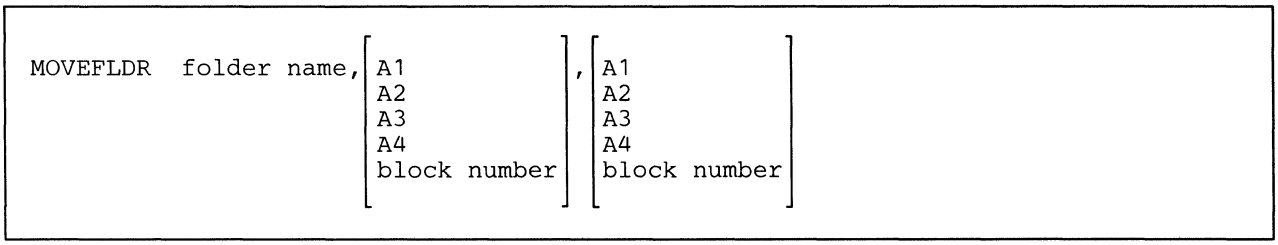
```
MCSCONV
```

S9020547-0

MOVEFLDR Procedure

The MOVEFLDR procedure allows you to move a folder from one disk location to another.

The MOVEFLDR procedure runs the $TMSERV utility program.



S9020556-0

folder name specifies the name of the folder to be moved.

A1, A2, A3, or A4 specifies the preferred location of the folder. If space is not available on the disk specified or a preferred location is not specified, the folder will be placed on the least used disk unit with enough space.

block number specifies the beginning block number of the folder.

A1, A2, A3, or A4 specifies the preferred location of the folder extent. If space is not available on the disk specified or a preferred location is not specified, the folder will be placed on the least used disk unit with enough space.

block number specifies the beginning block number of the folder extent.

Example

This example shows how to move a folder named MYFLDR to the third disk unit. If any extents exist for this folder, they will be moved to the fourth disk.

```
MOVEFLDR MYFLDR,A3,A4
```

MSGFILE Procedure

The MSGFILE procedure allows you to use the system message file (#MESSAGE). You can do the following:

- Define the size and location of the message file
- List the display stations and users that have messages in the message file
- Remove messages from the message file

Messages sent with the MSG control command or MSG OCL statement are saved in the message file.

To define the size and location of the message file:

```
MSGFILE DEFINE,[number of display stations to receive messages],
        [number of users to receive messages],[number of messages],
        [A1]
        [A2]
        [A3]
        [A4]
```

S9020160-0

To list the display stations and users that have messages in the message file:

```
MSGFILE SUMMARY
```

S9020161-0

To remove messages from the message file:

```
MSGFILE CANCEL,[ALL    ],[display station id]
               [DISPLAY]  [user id          ]
               [USER   ]
```

S9020162-0

DEFINE specifies the size and location of the message file. The values defined for the message file take effect when the MSGFILE procedure is run with the CANCEL,ALL parameters specified, or if an initial program load (IPL) is done when the message file is empty. The values remain in effect until new values are defined.

If password security is active, only a master security officer, security officer, or system operator can define values for the message file. If password security is not active, DEFINE can be specified only at the system console.

number of display stations to receive messages specifies the maximum number of display stations that can have messages in the message file at any one time. For example, if eight is specified, the message file can contain messages for eight different display stations at the same time.

The maximum number of display stations is 999. Because of the layout of the message file, the number specified is rounded up to the next multiple of eight. The initial default value is eight on a system with 30 megabytes or less of disk space and 128 on a system with greater than 30 megabytes of disk space.

number of users to receive messages specifies the maximum number of users that can have messages in the message file at any one time. For example, if eight is specified, the message file can contain messages for eight different users at the same time.

The maximum number of users is 9999. Because of the layout of the message file, the number specified is rounded off to the next multiple of eight. The initial default value is 8 on a system with 30 megabytes or less of disk space and 128 on a system with greater than 30 megabytes of disk space.

number of messages specifies the maximum number of messages that can be in the message file at any one time. The maximum number is 9999. The initial default value is 46 on a system with 30 megabytes or less of disk space and 217 on a system with more than 30 megabytes of disk space.

A1, A2, A3, or A4 indicates the disk preference for the message file; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, other units (if they exist) are checked, and the message file is placed on the least-used disk unit. If no parameter is specified, the message file is placed on disk A1.

SUMMARY specifies that a listing of the display stations and users that have messages in the message file is to be created. The listing is displayed or printed on the system list device. For information about assigning the system list device, see the "PRINT Procedure" on page 4-350 or the "SYSLIST Procedure" on page 4-498.

CANCEL specifies that messages should be removed from the message file.

ALL specifies that all messages in the message file are to be removed. A new message file is created with the values previously defined by the MSGFILE procedure. If the size of the message file was not previously defined, the default size is used.

If password security is active, only a master security officer, security officer, or system operator can remove all messages from the message file. If password security is not active, ALL can be specified only at the system console.

DISPLAY specifies that the messages for the specified display station are to be removed from the message file.

USER specifies that the messages for the specified user are to be removed from the message file.

display station id specifies the work station ID of the display station whose messages are to be removed from the message file.

If password security is active, a master security officer, security officer, or system operator can remove messages from the message file for any display station. Subconsole operators or display station operators can remove messages from the message file for the display station that they are using.

If password security is not active, an operator at the system console can remove messages from the message file for any display station. Operators at other display stations can remove messages from the message file for the display station that they are using.

If a display station ID is not specified, the messages for the display station from which the MSGFILE procedure was run are removed.

user id specifies the user ID of the operator whose messages are to be removed from the message file.

If password security is active, a master security officer, security officer, or system operator can remove messages from the message file for any user. Subconsole operators or display station operators can remove only their own messages.

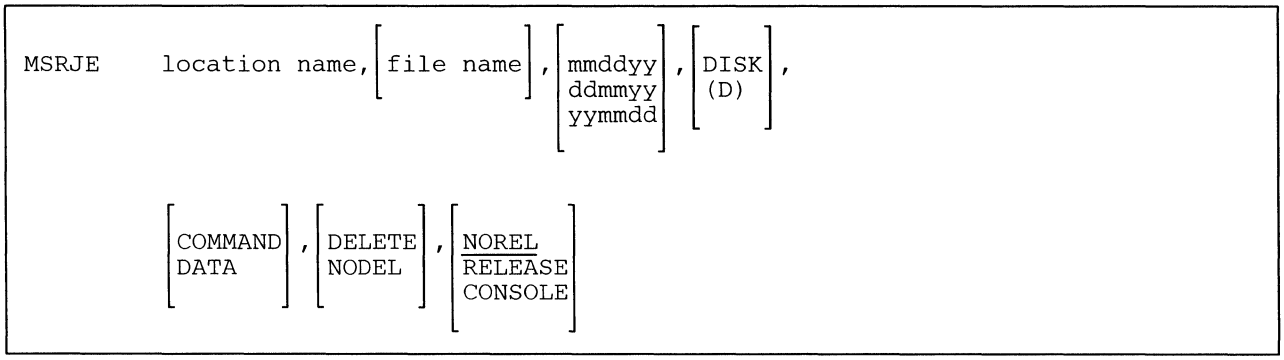
If password security is not active, an operator at the system console can remove messages from the message file for any user. Operators at other display stations can remove only their own messages.

If a user ID is not specified, the messages for the user who ran the MSGFILE procedure are removed.

MSRJE Procedure

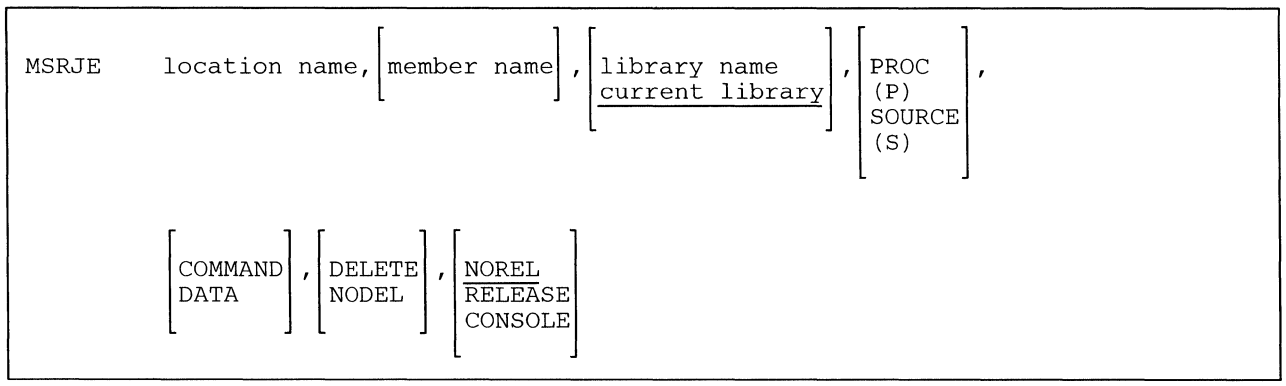
The MSRJE procedure starts the multiple session remote job entry (MSRJE) utility. For more information about MSRJE, see the *MSRJE Guide*.

To read a disk file:



S9020163-0

To read a library member:



S9020164-0

location name specifies the location with which communication is to be performed. The name is assigned when the MSRJE subsystem is configured during the CNFIGICF procedure. The location must have been activated by the ENABLE procedure.

file name specifies the disk file that is to be read. The file can be sequential, indexed, or direct.

member name specifies the library member that is to be read. The library member can be either a source or procedure member.

library name specifies the library that contains the library member to be read.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the disk file. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If this parameter is not specified, the most recent disk file is used.

DISK, PROC, or SOURCE specifies whether a disk file or library member is to be read. If no parameter is specified, the value that was specified when the MSRJE subsystem was defined using the CNFIGICF procedure (the TYPE parameter) is used.

DISK or D specifies that the name parameter is a disk file.

PROC or P specifies that the name parameter is a library procedure member.

SOURCE or S specifies that the name parameter is a library source member.

COMMAND or DATA specifies whether the disk file or library member being read contains only data or both data and control statements. If neither COMMAND nor DATA is specified, the value that was specified when the MSRJE subsystem was defined using the CNFIGICF procedure (the CMD parameter) is used.

COMMAND specifies that the disk file or library member contains MSRJE utility control statements. MSRJE processes all the MSRJE control statements and transmits all other records or statements. If a disk file is being read, the file is processed consecutively. The disk file or library member must have a record length of 80.

DATA specifies that the file or library member being read contains data. If a disk file is being read, the records are processed consecutively.

DELETE or NODEL specifies whether, any deleted records in the file are to be processed, if a disk file is being processed. If the disk file is not delete-capable, this parameter is ignored. If neither DELETE nor NODEL is specified, the value that was specified when the MSRJE subsystem was defined using the CNFIGICF procedure (the DEL parameter) is used.

DELETE specifies that deleted records are to be transmitted.

NODEL specifies that no deleted records are to be transmitted.

NOREL specifies that only a reader function is wanted, and that the display station is to remain attached to the MSRJE reader during the data transmission. This allows you to enter MSRJE commands. If no parameter is specified, NOREL is assumed.

RELEASE specifies that only a reader function is wanted, and that the display station is to be released from the MSRJE reader after initiation is complete to allow you to do other tasks at the display station while the data is being transmitted.

CONSOLE specifies that the display station being used is to be the MSRJE console. If another display station is already being used as the MSRJE console for this location, a message is displayed.

Example 1

This example shows how to sign on to MSRJE as the MSRJE console operator.

```
MSRJE STATION1,,,,,,CONSOLE
```

Example 2

This example shows how to sign on to MSRJE and transmit a library source member that contains both data and MSRJE process control statements contained in the source member. The name of the library member is PAYROLL; it is contained in the library named MYLIB.

```
MSRJE STATION1,PAYROLL,MYLIB,SOURCE,COMMAND
```

NOHALT Procedure

The NOHALT procedure specifies the automatic response severity level for the system, your session, or a single job. This allows you to have messages with response values to be automatically responded to by the system, rather than requiring an operator to enter the response to an error message.

For more information on automatic response and the severity level, see the "RESPONSE Procedure" on page 4-374.

If it is entered from the keyboard, it remains in effect until another NOHALT procedure is entered, or until the operator signs off the system. If it is specified in a procedure, it remains in effect until another NOHALT procedure is processed, or until the job ends.

The NOHALT procedure processes a NOHALT OCL statement.

```
NOHALT   severity level,[JOB    ]
                        [SESSION]
                        [SYSTEM ]
```

S9020165-0

severity level specifies the automatic response severity level. You can enter 0, 1, 2, 3, or 4:

0 Specifies that no messages are to be automatically responded to. If a message is displayed, the operator must enter a response to the message.

1 Specifies that any messages having a severity level of 1 are to be automatically responded to by the system.

2 Specifies that any messages having a severity level of 1 or 2 are to be automatically responded to by the system.

3 Specifies that any messages having a severity level of 1, 2, or 3 are to be automatically responded to by the system.

4 Specifies that any messages having a severity level of 1, 2, 3, or 4 are to be automatically responded to by the system.

JOB specifies that the severity level is to affect only the current job that contains the NOHALT procedure. If no parameter is specified when the NOHALT procedure is encountered in a job or procedure, JOB is assumed. After the job ends, the severity level that is in effect for the session is used.

SESSION specifies that the severity level is to affect your session; that is, all your jobs until you sign off the system. If the NOHALT procedure is within a procedure and SESSION is specified, the specified level does not take effect until the current job ends. If no parameter is specified when the NOHALT procedure is entered at the keyboard, SESSION is assumed.

SYSTEM specifies that the severity level is to establish the system severity level; that is, the severity level for display stations that **sign on after** the SYSTEM parameter is processed. Display stations that were already signed on are not affected.

To establish the system severity level before any operators can sign on to the system, include the NOHALT procedure in the initial program load (IPL) start-up procedure #STRTUP1. See the "#STRTUP1 Procedure" on page 4-4 for more information.

If password security is active, SYSTEM can be specified only by an operator having the classification of system operator or higher. If password security is not active, SYSTEM can be specified only at the system console. If the NOHALT procedure is within a procedure and SYSTEM is specified, the specified level takes effect immediately for the system automatic response severity level, but does not take effect for the display station session until the current job ends.

Example 1

To establish an automatic response severity level of 3 for the entire system, you could place the following statement in the procedure named #STRTUP1.

```
NOHALT 3,SYSTEM
```

Example 2

To establish an automatic response severity level of 3 for the payroll program, you could specify the following in the payroll procedure. The severity level of 3 affects only the statements that follow the NOHALT procedure; after the procedure ends, the session severity level is resumed.

```
NOHALT 3
// LOAD PAYROLL
// FILE NAME-EMPLOY,DISP-OLD
// RUN
```

Example 3

To establish an automatic response severity level of 3 for your session, enter the following from your keyboard.

```
NOHALT 3
```

OFCBPRT Procedure

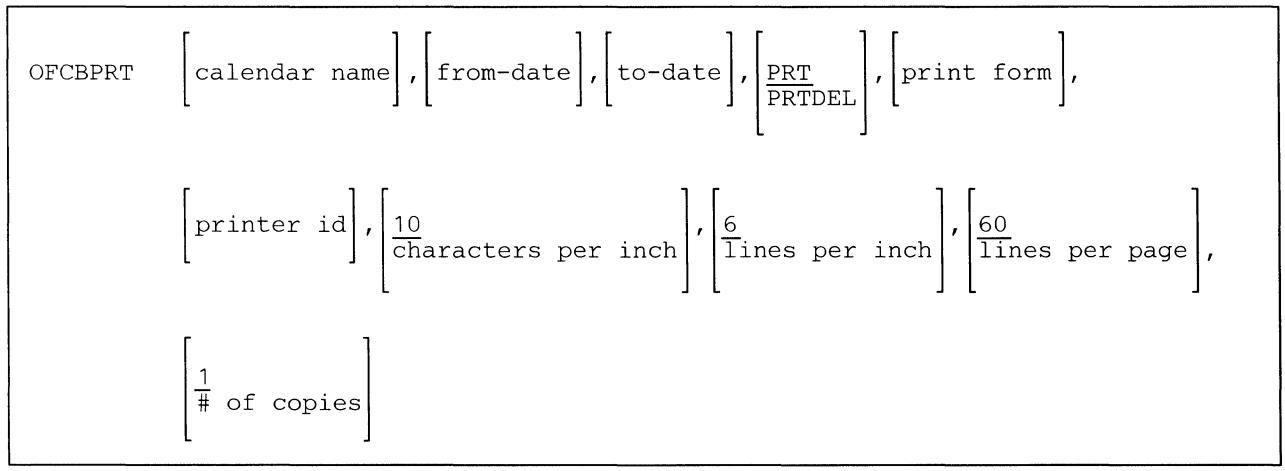
The OFCBPRT procedure allows a user to request batch printing and deleting of calendar items without going through Personal Services/36 menus and displays.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020522-0

calendar name specifies the name of the calendar to be printed or deleted. If no parameter is specified, the calendar name from the user's profile is assumed.

from-date, to-date specifies the range of the calendars to be printed or deleted. If **from-date** is not specified, the from-date is calculated from the calendar items. If **to-date** is not specified, the to-date is the last date of the previous week.

PRT specifies that the calendar is to be printed. If this parameter is not specified, PRT is assumed.

PRTDEL specifies that the calendar is to be printed and then deleted.

print form specifies the type of paper to be used. If this parameter is not specified, 0001 is assumed.

printer id specifies the ID of the printer to be used to print the calendar. If this parameter is not specified, the printer ID specified in the user's profile is used.

characters per inch specifies the number of characters per inch to be printed. If this parameter is not specified, 10 is assumed.

lines per inch specifies the number of lines per inch to be printed. If this parameter is not specified, 6 is assumed.

lines per page specifies the number of lines per page to be printed. If this parameter is not specified, 60 is assumed.

# of copies specifies the number of copies to be printed. If this parameter is not specified, 1 is assumed.

OFCCAL Procedure

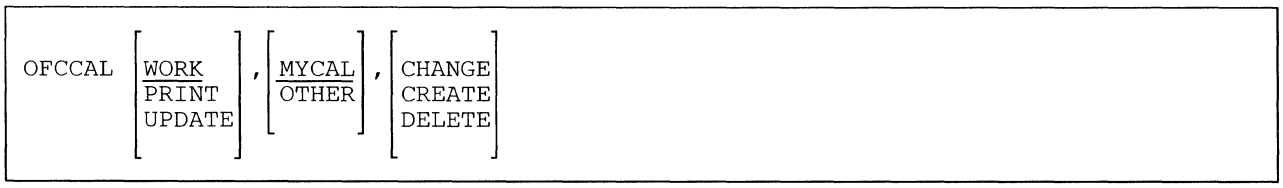
The OFCCAL procedure creates or maintains a calendar using Personal Services/36. You can create, change, or delete a calendar, and schedule, reschedule, or cancel calendar items.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020166-0

WORK specifies that you want to look at or change the items in a calendar or a group of calendars. If no parameter is specified, WORK is assumed.

PRINT specifies that you want to print or delete calendar items. You can also print or delete the items in a calendar for a range of dates.

UPDATE a calendar. UPDATE changes the description or manager of a calendar.

MYCAL specifies that you want to work with the calendar or group of calendars that is listed in your office profile. If no parameter is specified, MYCAL is assumed.

OTHER specifies that you want to work with a calendar other than the calendar or group or calendars that is listed in your office profile.

CHANGE specifies that you want to change the description or manager of a calendar. If resource security is active, you can also change the calendar entry in the resource security file.

CREATE specifies that you want to create a calendar. To create a calendar, you assign a calendar manager and specify the maximum number of calendar items. You can also associate a description with the calendar. If resource security is active, you can also add the calendar to the resource security file.

DELETE specifies that you want to remove a calendar from the system.

OFCCANCL Procedure

The OFCCANCL procedure stops Personal Services/36 background mail tasks.

For more information about the OFCCANCL procedure, see the manual *Administering Personal Services/36 in the Office*, SC09-1062.

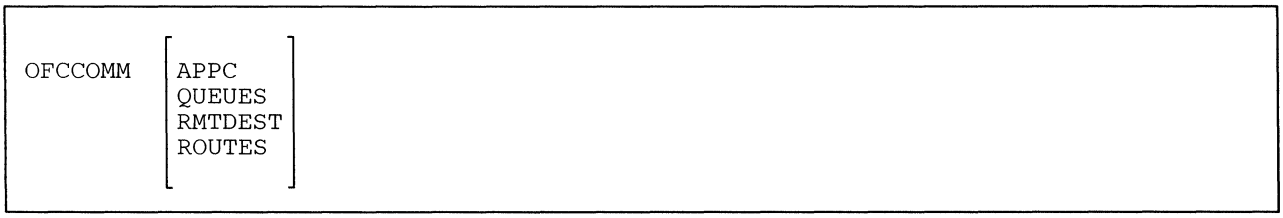
S9020590-0

The OFCCANCL procedure has no parameters.

OFCCOMM Procedure

The OFCCOMM procedure allows you to maintain System/36 or Personal Services/36 communications definitions. System/36 communications definitions refers to using Advanced Program-to-Program Communications (APPC) to specify information for the Interactive Communications Feature (ICF) subsystems definitions. Personal Services/36 communications definitions include queue definitions, remote destination definitions, and communication routes.

Note: The OFCCOMM procedure should not be used when communications is active.



S9020577-0

APPC specifies System/36 communications definitions using APPC are to be maintained.

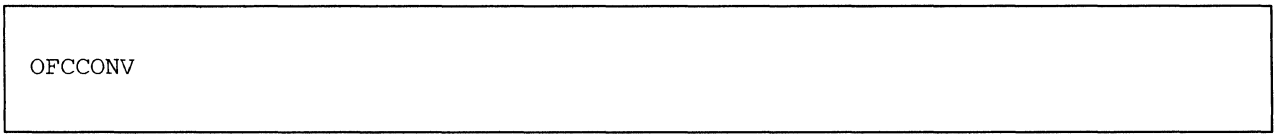
QUEUES specifies communications queue definitions are to be maintained.

RMTDEST specifies remote destinations are to be maintained.

ROUTES specifies communications routes are to be maintained.

OFCCONV Procedure

The OFCCONV procedure converts user profile data from Release 4 format to Release 5 format.



S9020578-0

The OFCCONV procedure has no parameters.

OFCDATA Procedure

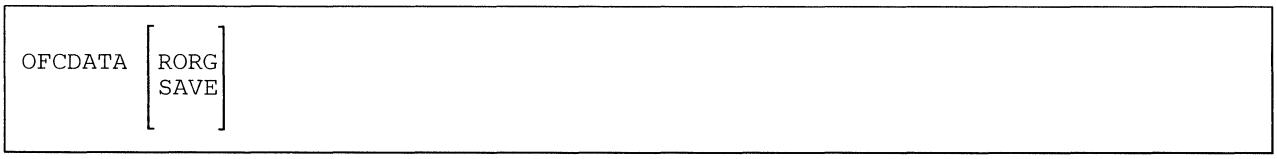
The OFCDATA procedure reorganizes or saves office information. Office information consists of Personal Services/36 files and folders, including calendars and mail logs. The OFCDATA procedure should be run when no one else is using Personal Services/36.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020167-0

RORG specifies that the office information is to be reorganized.

SAVE specifies that the office information is to be saved on diskette or tape.

OFCDFLT Procedure

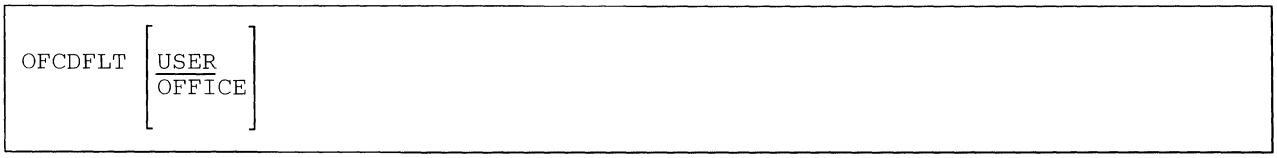
The OFCDFLT procedure changes the default values used by Personal Services/36. If password security is active, a security officer can change system default values. You can change the values in your own office profile.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020168-0

USER specifies that you want to change the default values in your own office profile. If no parameter is specified, USER is assumed.

OFFICE specifies that you want to change system default values. You must be a security officer to change these values.

OFCDIR Procedure

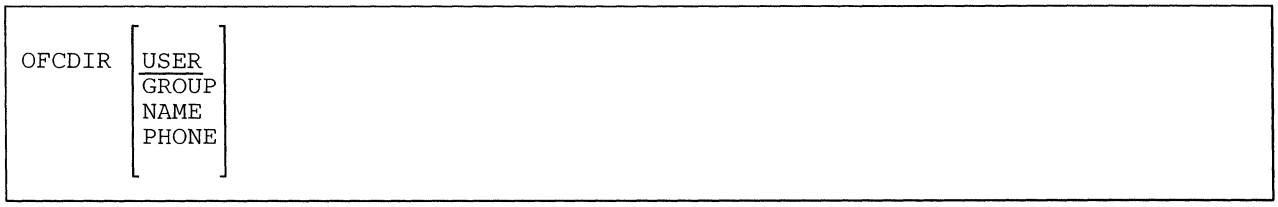
The OFCDIR procedure selects different ways of looking at the directory.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020169-0

USER specifies that you want to look at names and addresses, distribution addresses, and telephone numbers in alphabetic order by user identification. If no parameter is specified, USER is assumed.

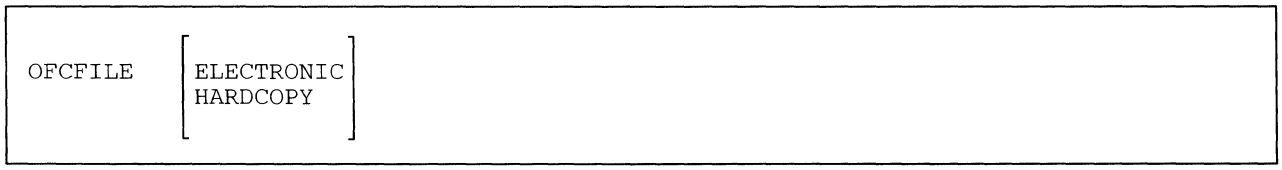
GROUP specifies that you want to look at names, user identifications, distribution addresses, and telephone numbers for the members of a group. You cannot add or delete names in the directory or group.

NAME specifies that you want to look at addresses of users in the directory in alphabetic order by name. You can add, change, or delete entries in the directory.

PHONE specifies that you want to look at telephone numbers of users in the directory in alphabetic order by name. You can add, change, or delete entries in the directory.

OFCFILE Procedure

The OFCFILE procedure allows you to file an electronic document in a library or you can log information about a hard copy document, which has been received, in a library.



S9020579-0

ELECTRONIC specifies that an electronic document is to be filed in a library.

HARDCOPY specifies that information about a hard copy document is to be logged in a library.

OFCGRP Procedure

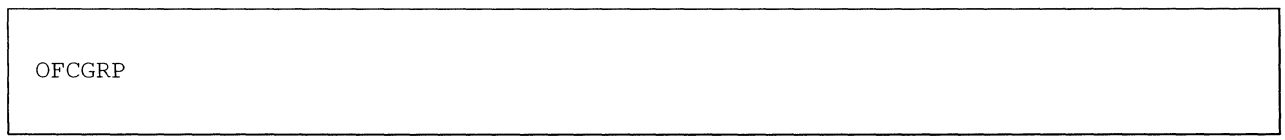
The OFCGRP procedure allows you to work with a user group.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020170-0

The OFCGRP procedure has no parameters.

OFCINSTL Procedure

The OFCINSTL procedure installs Personal Services/36 files on the system.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display

```
OFCINSTL
```

S9020171-0

The OFCINSTL procedure has no parameters.

OFCLDF Procedure

The OFCLDF procedure allows you to view, add, delete, and change the descriptions of access codes, document classes, and keywords for each library.

```
OFCLDF   [ ACCESS  ]
         [ CLASS   ]
         [ KEYWORD ]
```

S9020580-0

ACCESS specifies that you want to view, add, delete, or change the description of access codes.

CLASS specifies that you want to view, add, delete, or change the description of document classes.

KEYWORD specifies that you want to view, add, delete, or change the description of keywords.

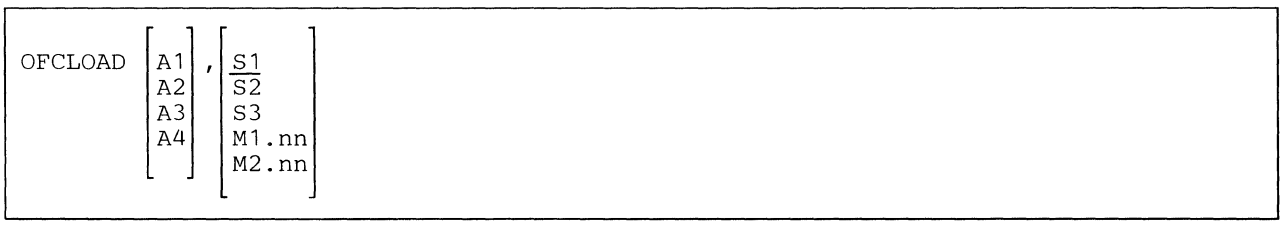
OFCLOAD Procedure

The OFCLOAD procedure creates a library named #OFCLIB and copies the Personal Services/36 support from diskette into that library. OFCLOAD copies additional support into the system library (#LIBRARY). The OFCLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the OFCSAVE procedure. See the "OFCSAVE Procedure" on page 4-322for information about how to save the Personal Services/36 support on diskette.

The OFCLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the OFCLOAD procedure to restore support that has been saved by OFCSAVE.

If Personal Services/36 is not currently on the system, you must use the TOLIBR procedure to copy the diskette file OFC into #LIBRARY before running OFCLOAD.

If #LIBRARY was backed up with Personal Services/36 on the system and then replaced before OFCLOAD is run, you do not have to copy the diskette file OFC using the TOLIBR procedure.



S9020172-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #OFCLIB is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, #OFCLIB is placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #OFCLIB on disk and copy the Personal Services/36 support from diskette.

```
OFCLOAD
```

OFCMAIL Procedure

The OFCMAIL procedure works with mail. You can look at the status of all entries in your mail log or just the action items. You can also send mail and look at the status of mail that you have sent.

Personal Services/36 automatically logs any mail that you send or receive through the system, and you can log any hard-copy mail that you send or receive. You can create or delete a mail log with the OFCMAIL procedure or work with a particular user's mail log.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display

```
OFCMAIL  [ STATUS   ]
         [ ACTION   ]
         [ ALTERLOG ]
         [ HARDCOPY ]
         [ MAILOUT  ]
         [ RECEIVE  ]
         [ REVLOG   ]
         [ SELECT   ]
         [ SEND     ]
         [ SENDNOTE ]
```

S9020173-1

STATUS specifies that you want to look at the status of entries in your mail log. You can look at new entries, entries of outgoing mail, and action items. If no parameter is specified, STATUS is assumed.

ACTION specifies that you want to look at the action items in your mail log. You can look at the item itself or the memo slip attached to it, change the description of the item, and print, file, send, or delete the item. You can also delete the entry for an item from your mail log.

ALTERLOG specifies that you want to create or delete a mail log, either for yourself or for a user whose mail you handle. If resource security is active, you can also add your mail log to the resource security file.

HARDCOPY specifies that you want to log hard-copy mail. You can enter the same information that Personal Services/36 automatically logs for mail sent or received through the system.

MAILOUT specifies that you want to look at entries of outgoing mail. You are shown a list of mail that you have sent but the addressee has not yet received.

RECEIVE specifies that you want to look at a list of the new mail in your mail log. You can look at the item itself or the memo slip attached to it, change the description of the item, and print, file, send, or delete the item. You can also delete the entry for an item from your mail log.

REVLOG specifies that you want to look at a list of the entries in your mail log, ordered by date with the most recent entries first. You can look at the item itself or the memo slip attached to it, change the description of the item, and print, file, send, or delete the item. You can also delete the entry for an item from your mail log.

SELECT specifies that you want to select a user's mail log to work with from a list of users who have mail logs.

SEND specifies that you want to send mail. You can specify a distribution list, priority, and security classification for the item. You select the document to send from a list of documents.

SENDNOTE specifies that you want to send a document in the form of a note. You can specify a distribution list, subject, reference, confirmed delivery, priority level, security, and the copy list. You can select the distribution list from a list of users or groups. You can also imbed an existing DW/36 document that is stored in a text folder.

OFCMAINT Procedure

The OFCMAINT procedure maintains office information. You can maintain communications queue definitions, communications routes, or mail folders. You can also list the contents of a mail folder.

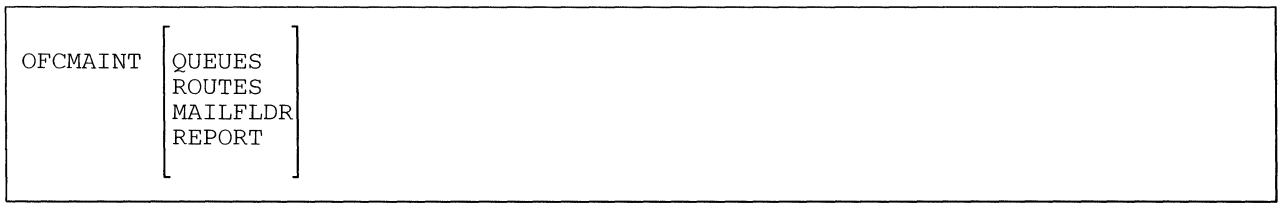
Except when listing the contents of a mail folder, the OFCMAINT procedure should be run when no one else is using Personal Services/36.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020174-0

QUEUES specifies that you want to look at each communications queue definition and add or select entries to be updated or deleted.

ROUTES specifies that you want to look at each communications route and add or select entries to be updated or deleted.

MAILFLDR specifies that you want to look at or maintain a mail folder. You can move an item to a document folder and look at it or delete the item from the mail folder.

REPORT specifies that you want to produce a report that lists the items in the mail folder and the users who have access to each item.

OFCMSG Procedure

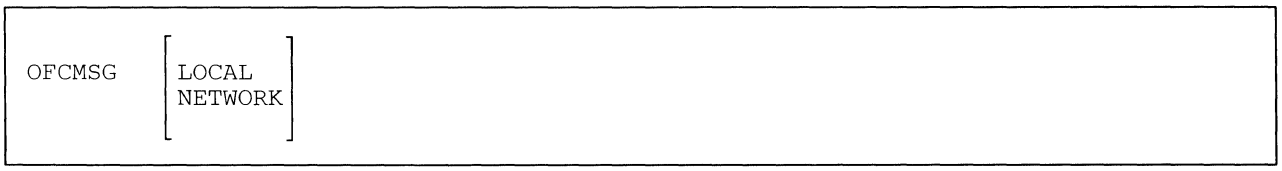
The OFCMSG procedure allows you to send messages to a group.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020175-1

LOCAL specifies that you want to send a message to user IDs or to work station IDs on your system.

NETWORK specifies that you want to send a message to user IDs or work station IDs on your system, on another system, or on independent work stations. The length of the message text must be under 256 characters.

OFCQ Procedure

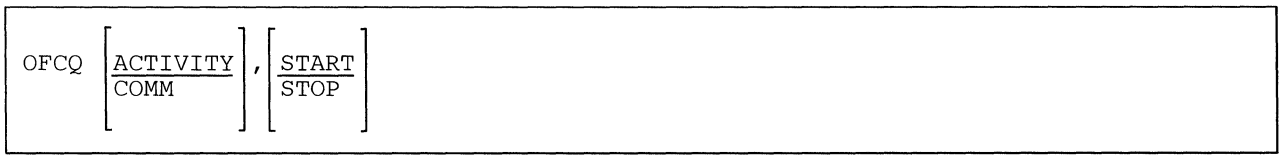
The OFCQ procedure controls the activity and communications queues.

For more information about Personal Services/36, see the online information. To read the online information:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020176-0

ACTIVITY specifies that you want to start or stop the activity queue. If no parameter is specified, ACTIVITY is assumed.

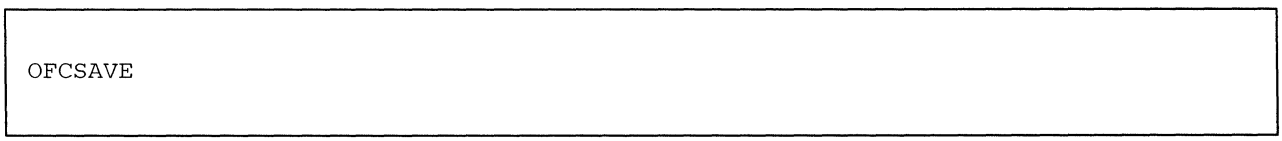
COMM specifies that you want to control the communications queues.

START specifies that you want to start the activity queue. Batch jobs and messages scheduled on Personal Services/36 calendars are not performed or responded to if the activity queue is not started. If no parameter is specified, START is assumed.

STOP specifies that you want to stop the activity queue.

OFCSAVE Procedure

The OFCSAVE procedure copies the Personal Services/36 support to diskette. The Personal Services/36 support from the libraries #OFCLIB and #LIBRARY is copied. You should use the "OFCLOAD Procedure" on page 4-317 to load the Personal Services/36 support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPOFC.

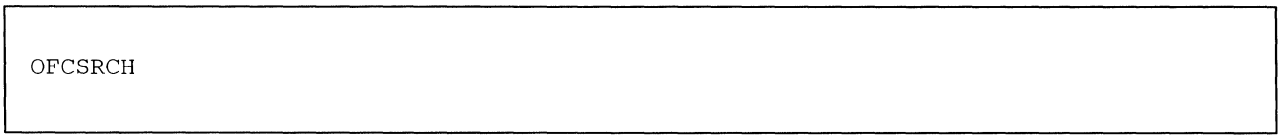


S9020177-1

The OFCSAVE procedure has no parameters.

OFCSRCH Procedure

The OFCSRCH procedure works with searches and documents found by a search.

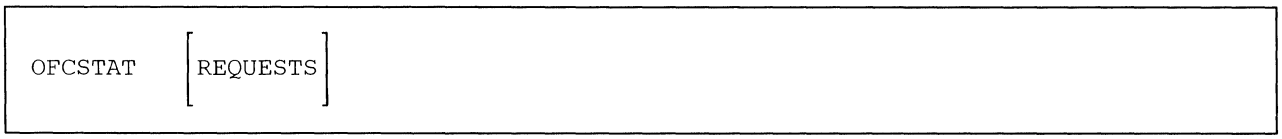


S9020581-0

The OFCSRCH procedure has no parameters.

OFCSTAT Procedure

The OFCSTAT procedure allows you to view a list of library requests. Library requests are the functions that can be carried out using library services such as filing, searching for, and retrieving documents.



S9020582-0

REQUESTS specifies that you want to view the status of library services requests.

OFCUSER Procedure

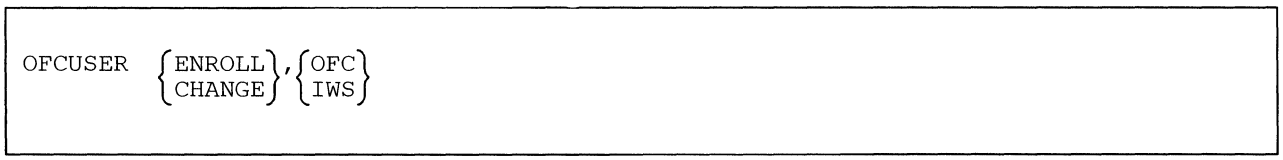
The OFCUSER procedure enrolls or changes the enrollment of general and indirect users of Personal Services/36. An indirect user is not enrolled.

For more information about Personal Services/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For Personal Services/36, specify:

  ```
  READINFO #OFCDOC,#OFCFLDR
  ```

- Select option 11 from the Use Personal Services/36 menu
- Press the Help key from any Personal Services/36 display



S9020178-1

ENROLL specifies that you want to enroll Personal Services/36 users.

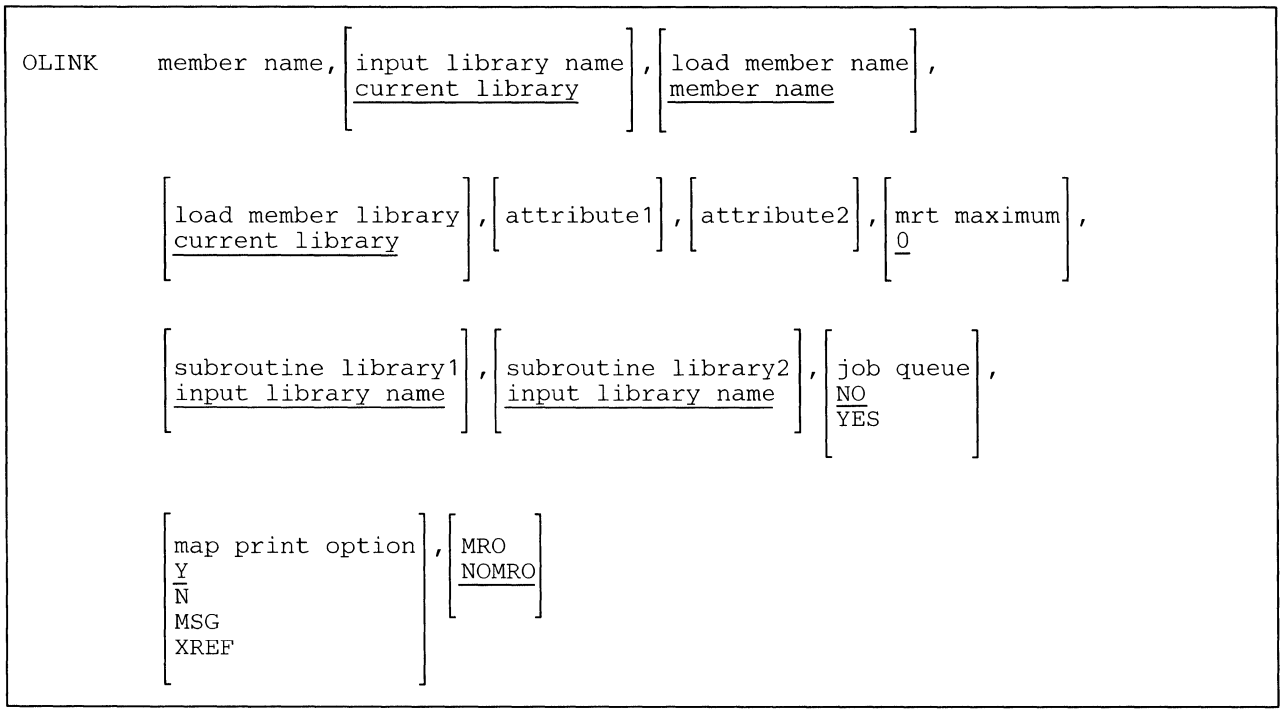
CHANGE specifies that you want to look at a list of general or indirect users. From the list, you can change or remove directory entries or enrollments.

OFC specifies that you want to enroll or change the enrollment of a general user.

IWS specifies that you want to enroll or change the enrollment of an indirect user.

OLINK Procedure

The OLINK procedure link-edits compiled programs and subroutines to create a library load member that can be run. For more information about the OLINK procedure, see the *OLE Guide*.



S9020179-1

member name specifies the compiled program to be link edited into a load member that can be run. The member must be a library subroutine member. If this parameter is not entered, a display is shown that prompts you for the parameters.

input library name specifies the library that contains the compiled program to be link edited. If no parameter is entered, the current library is assumed.

load member name specifies the name to assign the link edited program. If no parameter is entered, the member name is assumed.

load member library specifies the library that is to contain the link edited program. If no parameter is entered, the current library is assumed.

attribute1,attribute2 specifies attributes or indicators that are to be assigned to the link edited load member. Only two attributes can be entered. The attributes you can enter are as follows:

COM indicates that the load member requires the calculation of a new load address when the program is loaded. This ensures that the new main storage load address is beyond the program's own common region.

DED indicates that the load member must run alone on the system (dedicated). No other programs can be run.

LSC indicates that the program can only be loaded (and run) from the system console.

NEP indicates that the program is a never ending program. This means that any system resources (such as disk files or printers) used by the program will not be available to other programs. This prevents the other programs from waiting until the NEP program ends.

NEX indicates that the load member cannot be run (it is not executable). The member cannot be loaded using a LOAD OCL statement. If NEX is specified, none of the other attributes can be specified.

NIQ indicates that the program is not inquirable; that is, another program cannot be run while the NIQ program is being run.

NSW indicates that the program is not swappable. This means that once the program is loaded, it remains in main storage until the program ends and is not swapped in and out of main storage. For information on swapping, see the *Concepts and Programmer's Guide*.

SIS indicates that the program requires the Scientific Instruction Set to run.

SRQ indicates that the program requires that the $SOURCE and $WORK files be allocated.

UCS indicates that the program reads utility control statements.

mrt maximum indicates that the program is a multiple requester terminal (MRT) program, and specifies the maximum number of terminals that can be using the program at any one time. Any number from 0 through 255 can be entered.

If 0 is entered, the program is a single requester terminal, not an MRT. If no parameter is entered, 0 is assumed.

subroutine library1,subroutine library2 specifies the names of up to two libraries to be searched for subroutine members. If no parameters are specified, the input library is assumed.

job queue specifies whether the OLINK job is to be placed on the job queue.

NO specifies that the OLINK job is not to be placed on the job queue. If no parameter is entered, NO is assumed.

YES specifies that the OLINK job is to be placed on the job queue.

map print option specifies the type of printed output to be produced from the link-editing. If no parameter is specified, Y is assumed.

Y specifies the printing of a storage map and messages.

N specifies that no storage map or messages are to be printed.

MSG specifies that only messages are to be printed.

XREF specifies the printing of a storage map, cross-reference listing, and messages.

MRO or NOMRO specifies whether the compiler is to use memory resident overlays. If no parameter is specified, NOMRO is assumed.

MRO specifies that the compiler is to use memory resident overlays.

NOMRO specifies that the compiler is not to use memory resident overlays.

Example

This example shows how to link edit the program PAYROLL. The compiled program is contained in the library named PAYLIB, and the link-edited load member is to be placed in PAYLIB. The name of the link edited load member is also to be PAYROLL. The program is to be noninquirable and can only be run from the system console. Any subroutines needed during the link edit process are to be gotten from the library named SUBLIB.

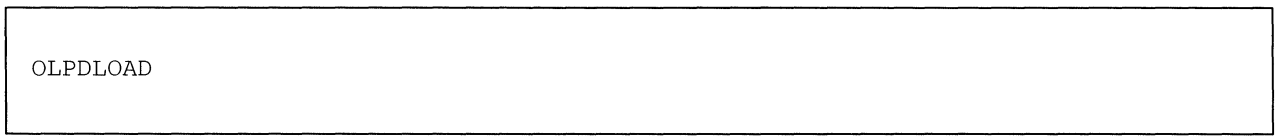
```
OLINK PAYROLL,PAYLIB,,PAYLIB,NIQ,LSC,,SUBLIB
```

OLPDLOAD Procedure

The OLPDLOAD procedure copies the online problem determination (OLPD) support from diskette to disk, using the backup diskettes created by the OLPDSAVE procedure. See the "OLPDSAVE Procedure" on page 4-329 for information about how to save the OLPD support on diskette.

The OLPDLOAD procedure copies OLPD support for base OLPD into library #ONLPD, and files #PSPTITL, PD1.SCRN, PD1.SCR2, and PD1.CTRL. In addition, if feature OLPD is loaded on the system when the OLPDSAVE procedure is run, the feature OLPD is restored to the system by OLPDLOAD. Feature OLPD support for tape is loaded into files PD2.TAPE, PD2.TAP2, and PD2.TCTL; feature OLPD support for communications is loaded into files PD3.COMM, PD3.COM2, and PD3.CCTL; feature OLPD support for local area network (LAN) is loaded into files PD4.LANA, PD4.LAN2, and PD4.LCTL; and feature OLPD support for external disk is loaded into files PD5.DISK, PD5.DIS2, and PD5.DCTL.

Insert the volume 1 diskette with volid-BKOLPD from the OLPDSAVE procedure. (On a system with a magazine, insert volume 1 in S1 and volume 2 in S2.) You will not have a volume 2 diskette if there is no feature OLPD support for tape, LAN, or external disk.



S9020583-0

The OLPDLOAD procedure has no parameters.

Example

Copy the OLPD support to disk from diskette.

```
OLPDLOAD
```

OLPDSAVE Procedure

The OLPDSAVE procedure copies the online problem determination (OLPD) support to diskette. The OLPD support is copied from library #ONLPD and all of the OLPD optional support files. To load OLPD support from the backup diskettes, use the "OLPDLOAD Procedure" on page 4-328.

Initialize two diskettes 2D to a volid of BKOLPD, FORMAT2 (1024 bytes per sector). Have a diskette (volume 1) in the drive when you start the OLPDSAVE procedure. (On a magazine drive, place the volume 1 diskette in S1 and volume 2 in S2.)

Two diskettes are needed if support is present for tape, local area network (LAN), or external disk on the 5362 System Unit. You need only one diskette if the base OLPD and the communications OLPD are the only support loaded.

Note: If a member of any group (such as PD1, PD2, PD3, or PD5 groups) is not on the system or has an error, none of the members of that group will be saved. For instance, if PD3.COM2 was deleted or never copied to the system, this procedure would not save the members of that group that are on the system (PD3.COMM or PD3.CCTL).

```
OLPDSAVE
```

S9020531-0

The OLPDSAVE procedure has no parameters.

Example

Copy the OLPD support to diskette.

```
OLPDSAVE
```

ORGANIZE Procedure

The ORGANIZE procedure is supported only for compatibility with the IBM System/34. To organize a disk file, see the "COPYDATA Procedure" on page 4-111. To organize a disk file and save it on diskette, see the "SAVE Procedure" on page 4-416.

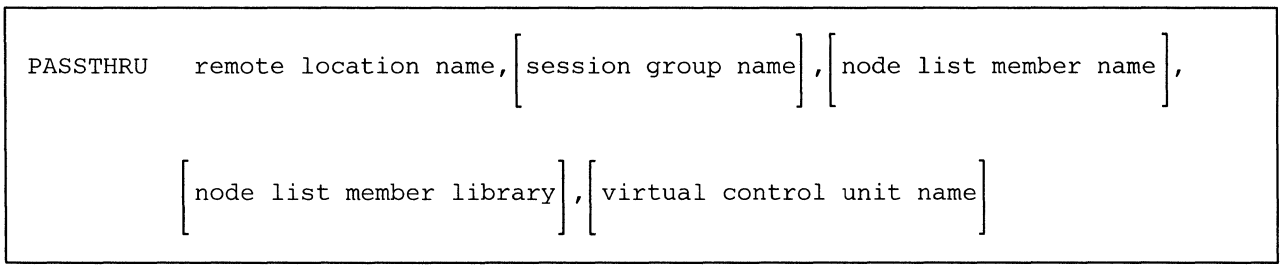
OVERRIDE Procedure

The OVERRIDE procedure is supported only for compatibility with the IBM System/34. See the "ALTERCOM Procedure" on page 4-11 for information about changing communications information.

PASSTHRU Procedure

The PASSTHRU procedure allows you to pass through from your system to a remote System/36 or System/38 where you can sign on as if your display station were attached to the remote system.

For more information about how to run the PASSTHRU procedure, see the manual *Using System/36 Communications*.



S9020510-2

remote location name specifies the remote location that you want to sign on to. This parameter is required.

session group name specifies the APPC session group to be used by display station pass-through. If you want blanks for a session group name, *BLANK must be specified. If this parameter is not specified, the APPC session group name configured as the default will be used.

node list member name specifies the name of a source member that contains a list of the System/38 location names that must act as intermediate nodes in order to complete the pass-through connection. If this parameter is not specified, the display station pass-through support assumes that no System/38 system needs to act as an intermediate node to complete the pass-through session.

node list member library specifies the library that contains the node list member. If this parameter is not specified, the current library is assumed.

virtual control unit name specifies the name of the virtual control unit that was created using the CRTCUD command on the System/38. This prompt is required if the remote system is a System/38.

Example

This example shows how to pass through to a System/36 with a location name of Chicago.

```
PASSTHRU CHICAGO
```

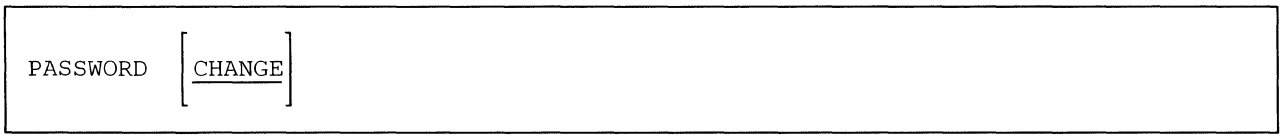
PASSWORD Procedure

The PASSWORD procedure allows you to change your password.

For more information about how to run the PASSWORD procedure, see the *System Security Guide*.

Password security must be active to run the PASSWORD procedure.

The PASSWORD procedure runs the $PRPWD utility program.



S9020511-1

CHANGE specifies that your password is to be changed. CHANGE is the default and need not be specified.

Example

This example shows how to change your password.

```
PASSWORD
```

PCEXCH Procedure

The PCEXCH procedure allows you to exchange data between a virtual disk or virtual diskette and a folder. The File Support Utility can create a virtual diskette on the System/36; PC Support/36 can create a virtual disk. Virtual disks or virtual diskettes contain data from an IBM Personal Computer.

To exchange data from a folder to a virtual disk or virtual diskette:

```
PCEXCH DOCUMENT,[document name],[folder name],[disk/diskette name],

      [file name],[disk subdirectory],[NOREPLACE / REPLACE],[NODELETE / DELETE],

      [folder subdirectory]
```

S9020482-1

To exchange data from a virtual disk or virtual diskette to a folder:

```
PCEXCH PCFILE,[disk/diskette name],[file name],[DATA / FINAL / REVISABLE],[disk subdirectory],

      [STORE / TEXT / PRINT],[document name],[folder name],[NOREPLACE / REPLACE],[retention date],

      [document description],[printer id],[NODELETE / DELETE],[folder subdirectory]
```

S9020483-1

DOCUMENT specifies that data is to be exchanged from a folder to a virtual disk or virtual diskette.

document name specifies the DW/36 document to be exchanged.

folder name specifies the folder that contains the document.

disk/diskette name specifies the virtual disk or virtual diskette that is to receive the exchanged data.

file name specifies the file on the virtual disk or virtual diskette that is to contain the exchanged data.

| **disk subdirectory** specifies the alternative to the master directory of the virtual disk or virtual diskette in which the file is to be listed. If no parameter is specified, the master directory is assumed.

NOREPLACE specifies that if a file with the same name already exists in the directory, an error message will be issued. If no parameter is specified, NOREPLACE is assumed.

REPLACE specifies that if a document with the same name already exists in the master directory, it will be replaced.

PCFILE specifies that data is to be exchanged from a Personal Computer virtual disk or virtual diskette to a folder.

disk/diskette name specifies the virtual disk or virtual diskette that contains the file.

file name specifies the file on the virtual disk or virtual diskette that is to be exchanged.

DATA specifies that the file to be exchanged contains data.

| **FINAL** specifies that the file to be exchanged contains a final form or resolved document.

REVISABLE specifies that the file to be exchanged contains a revisable form document.

| **disk subdirectory** specifies the alternative to the master directory of the virtual disk or virtual diskette in which the file is listed. If no parameter is specified, the master directory is assumed.

STORE specifies that the file is to be stored in its present format in the specified document.

TEXT specifies that the file is to be converted to a DW/36 format in the specified folder. If TEXT is specified, FINAL or REVISABLE must also be specified.

PRINT specifies that the file is to be printed. If PRINT is specified, FINAL must also be specified.

document name specifies the document that is to contain the exchanged data.

folder name specifies the folder that is to contain the document. If no parameter is specified, the current folder is assumed.

NOREPLACE specifies that if a document with the same name already exists in the specified folder, an error message will be issued. If no parameter is specified, NOREPLACE is assumed.

REPLACE specifies that if a document with the same name already exists in the specified folder, it will be replaced.

retention date specifies how long the document to be created is to be retained.

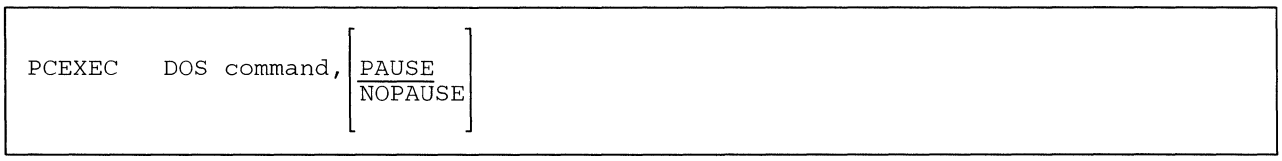
document description is a description of the document to be created. The description can be up to 35 characters long, and must be enclosed in single quotes.

printer id specifies the work station ID of the printer to be used if PRINT is specified. If no printer is specified, the system printer is assumed.

PCEXEC Procedure

The PCEXEC procedure allows the PC Support/36 Organizer user to issue commands on the personal computer.

If the PC Support/36 Organizer is not active and the PCEXEC procedure is used, a message appears telling you that PCO.EXE is not active; therefore, no communications with the personal computer can occur. You must press the enter key to resume.



S9020599-0

DOS command specifies the command you want to execute on the personal computer. If you enter COMMAND.COM, the DOS environment is assumed, and you will remain in DOS until you type EXIT to return to the System/36. If the DOS command you enter contains either blanks or commas, you must enclose the DOS command in single quotes (').

PAUSE specifies that you can press any key to return after the DOS command has been executed. If the first parameter is COMMAND.COM, this parameter must be blank or an error is issued.

NOPAUSE specifies that the PC Support/36 Organizer returns to the System/36 immediately after the DOS command has been executed. If the first parameter is COMMAND.COM, this parameter must be blank or an error is issued.

Example

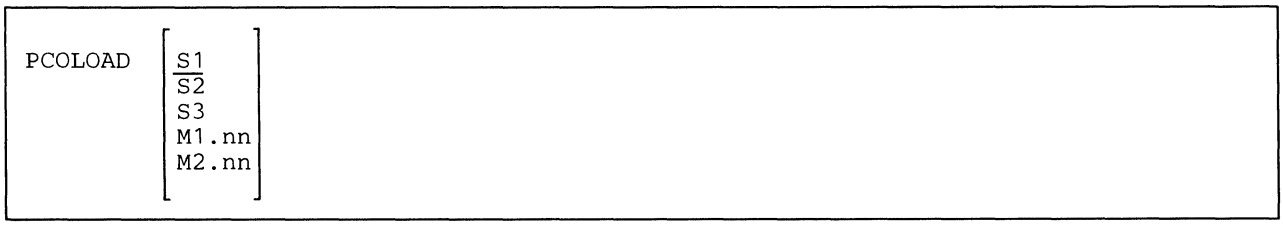
This example executes a DIR command on the personal computer while running under the PC Support/36 Organizer. The personal computer display remains on the screen until you press a key to return to the PC Support/36 Organizer.

```
PCEXEC 'DIR /W',PAUSE
```

PCOLOAD Procedure

The PCOLOAD procedure copies the PC Support/36 Organizer support from backup diskette to PC Support/36 and system libraries. The PCOLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the PCOSAVE procedure. See the "PCOSAVE Procedure" on page 4-337 for more information about how to save the PC Support/36 Organizer support on diskette.

Note: The PC Support/36 Organizer requires shared folders support to be on the system to load the PC Support/36 Organizer. If shared folders support is not on the system, you will be prompted to load shared folders support before the PC Support/36 Organizer can be loaded.



S9020600-0

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

PCOPROF Procedure

The PCOPROF procedure allows you to select a text editor. The PCOPROF procedure takes information from screens and stores that information into a profile. If there is no profile, the system will create one for you.

```
PCOPROF
```

S9020594-0

The PCOPROF procedure has no parameters.

PCOSAVE Procedure

The PCOSAVE procedure copies the PC Support/36 Organizer support from the libraries #IWLIB and #LIBRARY onto diskette. The PCOLOAD procedure should be used to load the PC Support/36 Organizer support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPIWS. The diskette must be in slot 1.

```
PCOSAVE
```

S9020601-0

The PCOSAVE procedure has no parameters.

PCU Procedure

The personal computer utility (PCU) procedure allows you to do the following, using PC Support/36:

- Copy all of the PC files within a virtual disk or diskette to a shared folder.

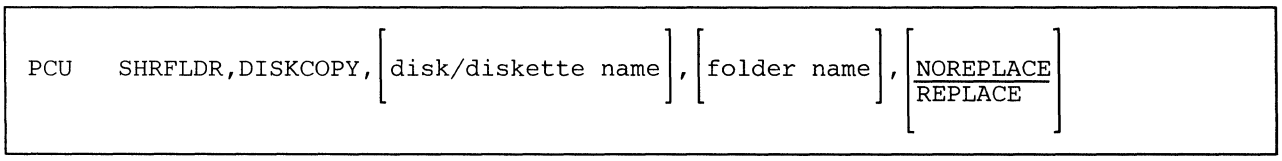
  Document-types with PC file extensions of:

  - RFT will default to RFTDCA (revisable form text document content architecture)
  - FFT will default to FFTDCA (final form text document content architecture)
  - All others will default to PCDATA.

  PC files with a PRF extension will be assumed to be profile files and will not be copied.
- Create a virtual disk.
- Delete a virtual disk.
- Copy System/36 files and library source and procedure members to a PC file stored on a virtual disk, and optionally translate characters from EBCDIC to ASCII.
- Copy PC files stored on a virtual disk to a System/36 file or library source or procedure member, and optionally translate characters from ASCII to EBCDIC.
- Copy PC files stored on a virtual disk to other PC files stored on the same or a different virtual disk.
- Copy PC files to a DisplayWrite/36 document (if DisplayWrite/36 is supported).
- Copy a DisplayWrite/36 document to a PC file (if DisplayWrite/36 is supported).
- Modify the ASCII to EBCDIC translation table.
- Modify the EBCDIC to ASCII translation table.

For more information about how to run the PCU procedure, see the *PC Support/36 Technical Reference.*

To copy all PC files within a virtual disk or diskette to a shared folder:



S9020595-0

DISKCOPY specifies that you want to copy all of the files and directories from the virtual disk to the folder.

disk specifies the name of the virtual disk that you want to copy.

diskette specifies the name of the virtual diskette that you want to copy.

folder name specifies the name of the folder that you want to copy all of the PC files into.

NOREPLACE specifies that like-named documents will not be replaced with the PC file information.

REPLACE specifies that existing documents with the same name as the PC files to be copied will be replaced with the PC file.

To create a virtual disk:

```
PCU   VIRTDISK,CREATE,{virtual disk name},{disk size},{directory size},
      ['description']
```

S9020536-0

To delete a virtual disk:

```
PCU   VIRTDISK,DELETE,{virtual disk name}
```

S9020537-0

To copy a virtual disk file to another virtual disk:

```
PCU   VIRTDISK,DISKDISK,{input PC file name},{input virtual disk name},
      [access path],{output PC file name},{output virtual disk name},
      [access path],[CREATE / REPLACE],[NOREADONLY / READONLY],[EXCLUSIVE / SHARE]
```

S9020538-0

To copy a virtual disk file to a System/36 file:

```
PCU   VIRTDISK,DISKFILE,{input PC file name},{input virtual disk name},
      [access path],{output S/36 file name},[CREATE / ADD / REPLACE],{maximum number of records},
      {record length for file},[XLATE / NOXLATE],[NOEND / END],[replacement character]
```

S9020539-0

To copy a System/36 file to a virtual disk file:

```
PCU   VIRTDISK,FILEDISK,{input S/36 file name},{output PC file name},
      {output virtual disk name},[access path],[CREATE / REPLACE],[NOREADONLY / READONLY],
      [EXCLUSIVE / SHARE],[XLATE / NOXLATE],[NOEND / END],[replacement character]
```

S9020540-0

To copy a virtual disk file to a System/36 library member:

```
PCU   VIRTDISK,DISKLIBR,{input PC file name},{input virtual disk name},
      [access path],{member name},[S / P],{library name},[CREATE / ADD / REPLACE],
      [member record length],[XLATE / NOXLATE],[NOEND / END],[replacement character]
```

S9020541-0

To copy a System/36 library member to a virtual disk file:

```
PCU   VIRTDISK,LIBRDISK,{member name},[S/P],{library name},{output PC file name},
      {output virtual disk name},[access path],[CREATE/ADD/REPLACE],[NOREADONLY/READONLY],
      [EXCLUSIVE/SHARE],[XLATE/NOXLATE],[NOEND/END],[replacement character]
```

S9020542-0

To copy a virtual disk file to a DisplayWrite/36 document (if DisplayWrite/36 is supported):

```
PCU   VIRTDISK,DISKDOC
```

S9020543-0

To copy a DisplayWrite/36 document to a virtual disk file (if DisplayWrite/36 is supported):

```
PCU   VIRTDISK,DOCDISK
```

S9020544-0

To modify the ASCII to EBCDIC translation table:

```
PCU   VIRTDISK,ASCII
```

S9020545-0

To modify the EBCDIC to ASCII translation table:

```
PCU   VIRTDISK,EBCDIC
```

S9020546-0

CREATE specifies that you want to create a virtual disk when using the CREATE and DISKDISK options. CREATE specifies that the System/36 file does not need to exist to copy to the data file on the DISKFILE option or the DISKLIBR option.

DELETE specifies that you want to delete a virtual disk.

DISKDISK specifies that you want to copy a virtual disk file to another virtual disk file.

DISKFILE specifies that you want to copy a virtual disk file to a System/36 file.

FILEDISK specifies that you want to copy a System/36 file to a virtual disk.

DISKLIBR specifies that you want to copy a virtual disk file to a System/36 library member.

LIBRDISK specifies that you want to copy a System/36 library member to a virtual disk.

DISKDOC specifies that you want to copy a virtual disk file to a DisplayWrite/36 document.

DOCDISK specifies that you want to copy a DisplayWrite/36 document to a virtual disk.

ASCII specifies that you want to modify an ASCII to EBCDIC translation table.

EBCDIC specifies that you want to modify an EBCDIC to ASCII translation table.

disk size specifies the capacity of the new virtual disk in K bytes.

directory size specifies the maximum number of files or subdirectories that can be contained in the root directory.

'description' specifies a description that can be up to 40 characters long. This description can contain any character except an apostrophe.

virtual disk name specifies the name of the virtual disk you want to create, delete, copy to, or copy from.

input PC file name specifies the name of the PC file to be copied.

input virtual disk name specifies the name of the virtual disk that contains the input file.

input S/36 file name specifies the name of the System/36 file that will be copied from.

output PC file name specifies the name of the PC file to receive the copy.

output virtual disk name specifies the name of the existing virtual disk that will receive the copied file.

output S/36 file name specifies the name to be given to the System/36 file that will contain the data copied from the personal computer.

access path specifies the name of one or more subdirectories.

READONLY specifies that you want PC programs to have an access level of read only to the virtual disk file.

NOREADONLY specifies that you want PC programs to be able to write or add data to the virtual disk file.

EXCLUSIVE specifies that you do not want other users to be able to read from the virtual disk while the virtual disk file is being copied to.

SHARE specifies that you will allow other users to be able to read from the virtual disk while the virtual disk file is being copied to.

ADD specifies that the System/36 file you are copying to must exist and you will be adding data to the end of the existing data file.

REPLACE specifies that the System/36 file or virtual disk file you are copying to with the same name will be replaced with the System/36 file or virtual disk file you are copying from.

maximum number of records specifies the maximum number of records the System/36 file can contain.

record length for file specifies the length of the records to be copied to the System/36 data file.

XLATE specifies that you want the date to be translated from ASCII to EBCDIC or EBCDIC to ASCII.

NOXLATE specifies that you do not want the data to be translated from ASCII to EBCDIC or EBCDIC to ASCII.

NOEND specifies that you want to substitute a replacement character for any characters that cannot be translated.

END specifies that you want to end the copy function when an untranslatable character is found. An error message is displayed.

replacement character specifies the character to be used when an untranslatable character is found.

member name specifies the name of the library member that is to be copied or that is to receive a copy.

S specifies that you want a source library member.

P specifies that you want a procedure library member.

library name specifies the name of an existing library that contains the library member.

member record length specifies the length of the records to be copied to the library member. The range is 40 through 120.

Example

This example shows how to start the PCU procedure to work with virtual disks.

```
PCU
```

A menu will be displayed and you can select one of the options from that menu.

POST Procedure

The POST procedure copies special E-format diskette files to disk files. The special E-format diskettes are created by the IBM 5260 Retail System. The POST procedure can also:

- Copy a special E-format diskette file or basic data exchange diskette file to a sequential or indexed disk file
- Add a special E-format diskette file or basic data exchange diskette file to an existing sequential disk file
- Copy a disk file to a basic data exchange diskette file
- Add records from a disk file to an existing basic data exchange diskette file

For information about the basic data exchange format, see the "TRANSFER Procedure" on page 4-542.

Because POST only processes special E-format or basic data exchange diskette files, a diskette used to receive POST output must be initialized in one of the following formats:

- A diskette 1 (one-sided) diskette initialized in the 128-byte per sector format.
- A diskette 2D (two-sided, double-density) diskette initialized in the 256-byte per sector format.

If the diskette format is not known, you can use the CATALOG procedure to list the diskette VTOC. This listing shows whether the diskette was initialized in the 128-, 256-, 512-, or 1024-byte format. See the "CATALOG Procedure" on page 4-73.

When a diskette file is copied to a disk sequential or indexed file, records are placed in the disk file sequentially, using the record length of the diskette file. When a diskette file is added to an existing disk sequential file, the records in the diskette file are either truncated or padded with zeros (x'00') to conform to the record length of the disk file.

When a sequential, indexed, or direct disk file is copied to a basic data exchange diskette file, the record length of the diskette file is set to the record length of the disk file, to 128 (for diskette 1), or to 256 (for diskette 2D), whichever is smaller. For example, if you were copying a sequential disk file with a record length of 300 to a diskette 2D, the record length of the basic data exchange diskette file would be 256, and record positions 257 through 300 would be truncated.

The POST procedure runs the $POST utility program.

To copy a special E-format diskette file or basic data exchange diskette file to a new sequential or indexed disk file:

```
POST     input file name,[I1],[mmddyy|ddmmyy|yymmdd],[NOADD],{,|key length,key position}

         {,|RECORDS,records|BLOCKS,blocks},[S1|S2|S3|M1.nn|M2.nn],[AUTO|NOAUTO],[NOEOD|EOD],[NODUPKEY|DUPKEY]
```

S9020180-0

To add a special E-format diskette file or basic data exchange diskette file to an existing sequential disk file:

```
POST     input file name,[I1],[mmddyy|ddmmyy|yymmdd],ADD,[output file name|input file name],[date],

         [S1|S2|S3|M1.nn|M2.nn],[AUTO|NOAUTO],[NOEOD|EOD]
```

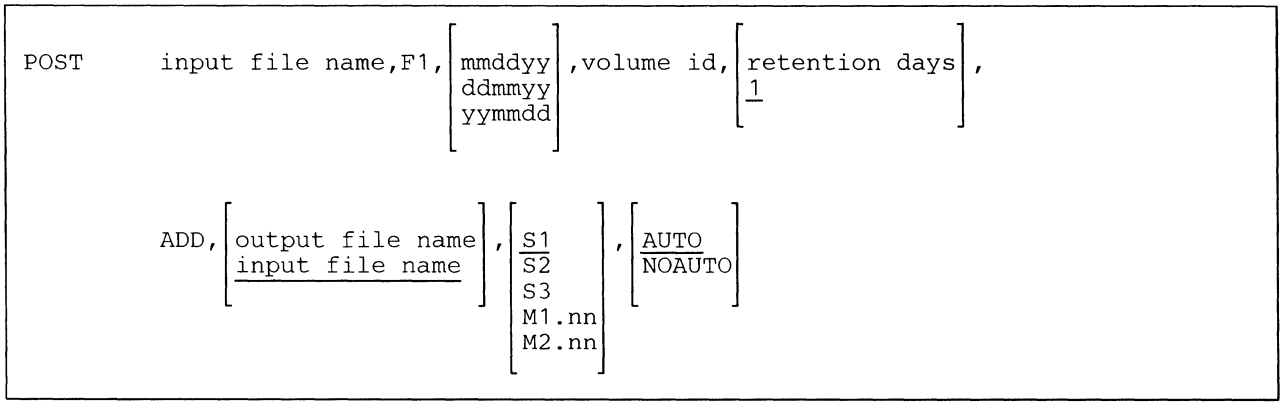
S9020181-0

To copy a disk file to a new basic data exchange diskette file:

```
POST     input file name,F1,[mmddyy|ddmmyy|yymmdd],volume id,[retention days|1],,,

         [S1|S2|S3|M1.nn|M2.nn],[AUTO|NOAUTO]
```

S9020182-0

To add a disk file to an existing basic data exchange diskette file:



S9020183-0

input file name specifies the file to be copied. If a new file is being created, it is given the name specified by this parameter.

I1 specifies that a diskette file is being copied to a sequential or indexed disk file. If the parameter is not specified, I1 is assumed.

F1 specifies that a disk file is being copied to a basic data exchange diskette file.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file being copied. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If more than one file with the specified name resides on disk and no date is specified, the file with the specified name and the most recent date is copied to diskette.

NOADD specifies that the diskette file being transferred will become a new sequential or indexed disk file with the name specified by input file name. If no parameter is entered, NOADD is assumed.

key length,key position specifies that an indexed file is to be created. If they are omitted, a sequential file is created.

key length specifies the key length for the indexed disk file that is to be created. The key length can be any number from 1 through 120.

key position specifies the starting position of the key in the record. The key position can be any number from 1 through 128 for diskette 1 or from 1 through 256 for diskette 2D.

RECORDS or BLOCKS specifies the size of the file. If neither RECORDS or BLOCKS is specified, the size of the diskette file is used for the size of the disk file. RECORDS or BLOCKS must be specified if:

- The diskette file is on more than one diskette.
- The created disk file is to be larger than the file being copied.

RECORDS specifies that the disk file being created must be large enough to contain the number of records specified.

records specifies the number of records to allocate for the disk file. Any number from 1 through 8000000 can be specified.

BLOCKS specifies that the disk file being created must be large enough to contain the number of blocks specified.

blocks specifies the number of blocks to allocate for the disk file. Any number from 1 through 312815 can be specified.

ADD specifies the following:

- If I1 is specified, ADD specifies that the records in a diskette file are to be added to the records in an existing sequential disk file.
- If F1 is specified, ADD specifies that the records in a disk file are to be added to the records in an existing basic data exchange diskette file. The first record from the disk file is placed after the last record in the diskette file.

output file name specifies the following:

- If I1 is specified, the output file name specifies the existing disk file to which a diskette file is to be added.
- If F1 is specified, the output file name specifies the existing diskette file to which a disk file is to be added.

The output file name is allowed only if ADD is specified. If no output file name is specified, the input file name is assumed.

date specifies the creation date of the existing disk file. The date is valid only if ADD is specified. The date must be specified in the same format as the session date.

volume id specifies the volume ID for the diskette. From 1 through 6 alphameric characters can be specified. The volume ID parameter indicates that a basic data exchange diskette file is to be created from a disk file.

retention days specifies the length of the retention period in days for the diskette file. If a retention period is not specified, 1 day is assumed. If a retention period of 999 days is specified, the diskette file is considered to be a permanent file. For more information on diskette file retention, see the "FILE OCL Statement (for Diskette Files)" on page 5-43.

S1, S2, or S3 specifies the diskette slot containing the first diskette from which members are to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette from which members are to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

NOEOD specifies that the special end-of-data condition (see the note that follows) is not to be detected when a diskette file is being copied or added to a disk file. If neither EOD nor NOEOD is specified, the NOEOD condition is assumed.

Note: The IBM 5260 Retail System indicates the end-of-data condition with a blank record. If EOD is specified when a basic exchange diskette file is being copied, data may be lost.

EOD specifies that the special end-of-data condition (see the note above) is to be detected when a diskette file is being copied or added to a disk file.

DUPKEY specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file is the attribute of the output file.

NODUPKEY specifies that duplicate keys are not to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file is the attribute of the output file.

Example 1

This example shows how to create an indexed disk file named FILE2 (with the key in positions 1 through 4) from a special E-format diskette file named FILE2, and check for special end-of-data.

```
POST FILE2,,,,4,1,,,,,EOD
```

Example 2

This example shows how to add a special E-format diskette file named FILE1 to an existing disk file named FILE1.

```
POST FILE1,,,ADD
```

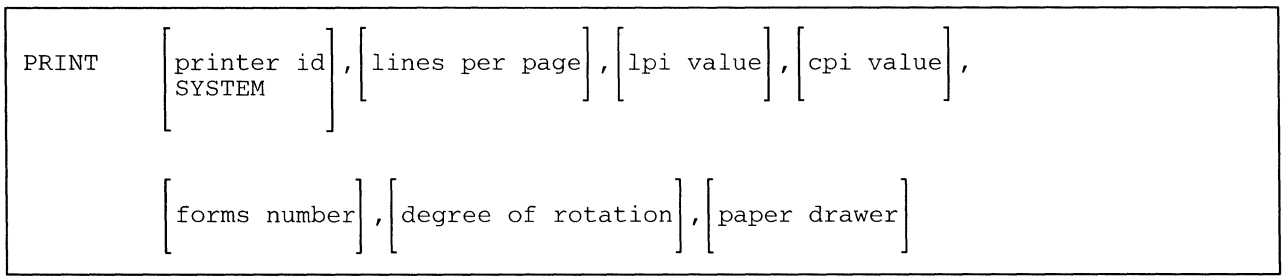
PRINT Procedure

The PRINT procedure specifies the following information about the printer to be used during a display station session unless a specific printer ID is specified in a job.

- The printer ID of the printer to be used for all printed output created during the display station session
- The number of lines to be printed per page
- The vertical print density (lines per inch)
- The horizontal print density (characters per inch)
- The forms number
- The orientation or size of printed output on the page (rotation or reduction)
- The printer drawer from which paper is to be used

A job placed on the job queue uses the values that were in effect when the job was placed on the queue. However, if a procedure running from the job queue contains a PRINT procedure, the PRINT procedure in that procedure is used.

The PRINT procedure processes a FORMS OCL statement.



S9020184-1

Although all parameters are optional, at least one parameter must be specified. Any parameters that are omitted cause the corresponding settings to remain unchanged.

printer id or SYSTEM specifies the printer to be used for printed output. All system list output, Print key output (unless otherwise indicated by the PRINTKEY procedure), and any other printed output from the session is printed on the specified printer. The PRINTER OCL statement can be used to specify a different printer for program output. If no parameter is specified, the printers are not changed.

printer id specifies the work station ID of the printer to be used. You can use the STATUS WORKSTN command to determine the printer IDs.

SYSTEM specifies that the system printer is to be used.

lines per page specifies the number of print lines per page. The maximum number of lines that can be specified per page is 112. This number is determined by multiplying the lines per inch setting times the height of the paper. For example, if the paper is 11 inches high and the lines per inch setting is 6, the lines per page setting should be 66 (11 x 6 = 66). The lines per page remains in effect until either the session ends, or a new lines per page value is specified by:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement
- The SET procedure or the $SETCF utility program

If no parameter is specified and the number of lines per page was not previously set during the session, the system uses the value specified for the display station during system configuration, by the SET procedure, or by the $SETCF utility program.

For IBM-supplied programs and most user-written programs, the printer skips to a new page when six less than the number of lines specified are printed. For example, if 66 is specified, the printer skips to a new page after printing line 60. Also, the printing starts on line 6 of the new page. Therefore, if you specify 66, you really get 54 (66 - 12 = 54) lines per page. If 13 is specified, one line is printed per page. When 12 or fewer lines are specified, printing occurs on every line of each page.

For print operations from your programs, the SSP indicates an overflow condition when six less than the number of lines specified (either in the program or in the PRINT procedure) are printed.

lpi value specifies the vertical print density (that is, lines per inch) to use for printed output from the display station session. The values that can be specified are 4, 6, and 8. If the output is printed on a printer that does not support vertical print density, the lines per inch value is ignored.

The lines per inch value remains in effect until either the session ends or a new lines per inch value is specified by:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement

If no lines per inch value is specified and the lines per inch was not previously set during the session, the system uses the value that was set when the printer was configured (either 4, 6, or 8). If no lines per inch value was specified when the printer was configured, the printer uses the setting at the printer (the set print density switch), and this parameter is ignored.

cpi value specifies the horizontal print density (that is, characters per inch) to use for printed output from the display station session. The values that can be specified are 10 or 15. The printer being used must support the horizontal print density specified. If the value specified is not supported by the printer, a message will appear and the operator controlling the printer can either continue or cancel the job.

The characters per inch value remains in effect until either the session ends or a new characters per inch value is specified by:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement

If no characters per inch value is specified and the characters per inch was not previously set during the session, the system uses the value that was set when the printer was configured (either 10 or 15). If no characters per inch value was specified when the printer was configured, the printer uses the setting at the printer (the set print density switch), and this parameter is ignored.

forms number specifies the forms number of the printer forms to be used for printed output from the display station session. (Each type of form should have a unique, user-assigned forms number.) The forms number can be any combination of up to 4 characters except commas (,), apostrophes ('), and blanks. Question marks (?), slashes (/), equal signs (=), greater than signs (>), plus signs (+), and hyphens (-) should be used with caution because they have special meanings within procedures.

If a forms number is specified, the operator who controls the printer is prompted to install the forms with the specified forms number in the printer if the specified forms are not already installed.

The forms number remains in effect until either the session ends or a new forms number is specified by:

- Another PRINT procedure
- The LINES procedure
- A PRINTER OCL statement (for that print file only)
- A FORMS OCL statement

degree of rotation specifies the degree of rotation that output is to be printed on the page, or that the output is to be reduced for printing on 8-1/2 x 11 inch paper. You can specify 0, 90, 180, or 270 degrees of rotation, or you can specify COR for computer output reduction. If you specify COR, output normally printed on 14-inch wide paper will print on 8-1/2 x 11 inch paper. This parameter is valid only for output printed on a 3812 Printer.

The **degree of rotation** value remains in effect until either the session ends or a new **degree of rotation** is specified by:

- Another PRINT procedure
- A PRINTER OCL statement ROTATE keyword parameter (for that print file only)
- A FORMS OCL statement ROTATE keyword parameter

paper drawer specifies the printer drawer from which paper is to be used. You can specify 1, 2, or 3 (for drawer 1, drawer 2, or drawer 3). The 3812 SNA Control Stream (SCS) and Intelligent Printer Data Stream (IPDS) Printers have two paper drawers, and the 4214 and 5219 Printers have three paper drawers. To select envelopes on the 5219 Printer, specify drawer 3.

The **paper drawer** value remains in effect until either the session ends or a new **paper drawer** is specified by:

- Another PRINT procedure
- A PRINTER OCL statement DRAWER keyword parameter (for that print file only)
- A FORMS OCL statement DRAWER keyword parameter

Example 1

The following procedure specifies that the printer with a printer ID of P3 is to be used for printed output during the session.

```
PRINT P3
```

Example 2

The following procedure specifies that the number of lines per page to be used is 51.

```
PRINT ,51
```

Example 3

The following procedure specifies that a printer with a printer ID of P2 is to be used for printed output during the session. The operator wants the printed output to be printed at 8 lines per inch and 15 characters per inch. Because the forms installed in the printer are 11 inches (27.9 cm) long, the number of lines per page is 88.

```
PRINT P2,88,8,15
```

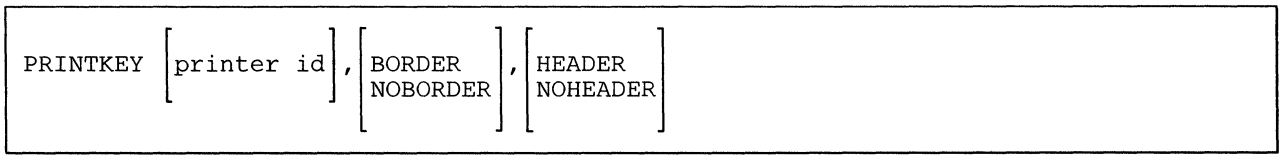
PRINTKEY Procedure

The PRINTKEY procedure allows you to specify the following:

- The printer to be used for the Print key output.
- Whether a border should be printed around the display image. This can be helpful in documenting your displays.
- Whether a heading should be printed before the display image. This can be helpful determining where the print request came from when no separator pages are being printed.

The changes made with the PRINTKEY procedure remain in effect during the session until changed by another PRINTKEY procedure, the SET procedure, the PRINT procedure, or a WORKSTN OCL statement.

The PRINTKEY procedure runs the $SETCF utility program.



S9020185-0

Although all parameters are optional, at least one parameter must be specified. Any parameters that are omitted cause the setting to remain unchanged.

printer id specifies the work station ID of the printer to be used to print the display image when the operator presses the Print key. The STATUS WORKSTN command can be used to determine the printer IDs.

BORDER specifies that a border should be printed around the display image. Figure 4-17 on page 4-355 shows a sample display image with a border.

NOBORDER specifies that no border should be printed around the display image.

HEADER specifies that a heading should be printed above the display image. Figure 4-17 on page 4-355 shows a sample display image with a header.

NOHEADER specifies that no heading should be printed above the display image.

Example 1

This example shows how to get a border printed around the displays when you press the Print key.

```
PRINTKEY ,BORDER
```

Example 2

This example shows how to get a border and a header printed, and how to get the printed output to go to printer P4.

```
PRINTKEY P4,BORDER,HEADER
```

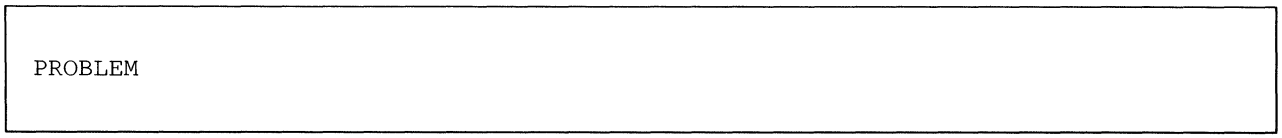
```
  ********************************************************************************
  **                                                                            **
  **           PRINT KEY FROM-W1        BY USER-MAT535     10/24/82    10.55.07  **
  **                                                                            **
  ********************************************************************************



    0   +    1    +    2    +    3    +    4    +    5    +    6    +    7    +    8
    1   +    0    +    0    +    0    +    0    +    0    +    0    +    0    +    0
    ********************************************************************************
01 * COMMAND                    Mike's General Menu                          W1 * 01
02 *                                                                              * 02
03 * Select one of the following:                                                * 03
04 *                                                                              * 04
05 *    1. Start BASIC                          10. Display system list output    * 05
06 *    2. Start SDA                            11. Print system list output      * 06
07 *    3. Start SEU                                                              * 07
08 *    4. Start Help                           12. Run display edit program      * 08
09 *                                            13. Run display spool program     * 09
10 *    5. List library information                                               * 10
11 *    6. List catalog (VTOC) information      14. List a data file              * 11
12 *                                            15. Delete a data file            * 12
13 *    7. Set Print key border/header                                            * 13
14 *    8. Restore Print key border/header      16. Save TEMLIB on diskette       * 14
15 *                                                                              * 15
16 *                                            24. Sign off                      * 16
17 *                                                                              * 17
18 *Cmd3-Previous menu   Help-More information                                    * 18
19 *                                                                              * 19
20 *                                                                              * 20
21 * Enter option number or command                                              * 21
22 *                                                                              * 22
23 *                                                                              * 23
24 *                                                                              * 24
    ********************************************************************************
    0   +    1    +    2    +    3    +    4    +    5    +    6    +    7    +    8
    1   +    0    +    0    +    0    +    0    +    0    +    0    +    0    +    0
```

Figure 4-17. Sample Print Key Output with Header and Border

PROBLEM Procedure

The PROBLEM procedure starts the online problem determination (OLPD) procedures. The PROBLEM procedure should be run from the physical system console at the system location. If the system console has been assigned to another display station, you should use the CONSOLE GIVE and CONSOLE TAKE commands to restore the physical system console.

OLPD may ask the operator to perform some tasks during the course of problem determination. The tasks may be mounting a tape or diskette, putting a modem in self test, or placing a wrap connector on a cable. Help is provided on the OLPD screens and is intended to be used by the person performing the tasks. The help contains graphics that are often not displayable on a Distributed Host Command Facility (DHCF) or other 3270 device.



S9020186-0

The PROBLEM procedure has no parameters.

Example

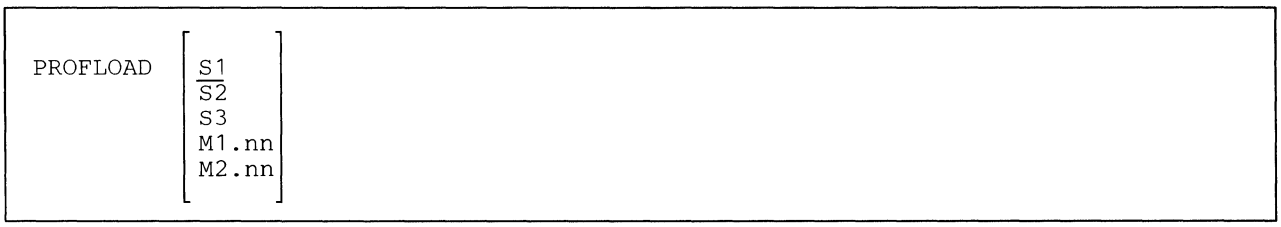
This example shows how to start the online problem determination procedures.

```
PROBLEM
```

PROFLOAD Procedure

The PROFLOAD procedure copies PROFS bridge support from a backup diskette to the Personal Services/36 library, #OFCLIB, and the system library, #LIBRARY. The PROFLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the PROFSAVE procedure. See the "PROFSAVE Procedure" on page 4-358 for more information about how to save PROFS bridge support on diskette.

Note: PROFS bridge support requires Personal Services/36 to be on the system to load PROFS bridge support. If Personal Services/36 is not on the system, you will be prompted to load Personal Services/36 before PROFS bridge support can be loaded.



S9020628-0

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

Example

Copy PROFS bridge support to #OFCLIB and #LIBRARY from a backup diskette.

```
PROFLOAD
```

PROFSAVE Procedure

The PROFSAVE procedure copies PROFS bridge support from the libraries #OFCLIB and #LIBRARY onto diskette. The PROFLOAD procedure should be used to load PROFS bridge support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPOFC. The diskette must be in slot 1.

```
PROFSAVE
```

S9020629-0

The PROFSAVE procedure has no parameters.

Example

Copy the PROFS bridge support onto diskette.

```
PROFSAVE
```

PRTGRAPH Procedure

The PRTGRAPH procedure is used to print graphic files on an intelligent printer data stream (IPDS) printer. Examples of graphics files are Business Graphics Utilities/36 (BGU/36) graphics files or graphics files built with the BLDGRAPH procedure. For more information, see the manual *Using the Intelligent Printer Data Stream Advanced Functions PRPQ*, GC21-9480.

The PRTGRAPH procedure runs the $DPGP utility.

For more information, see the manual *Concepts and Programmer's Guide.*

```
PRTGRAPH   prtid,FILE,graphics file name,width
```

S9020557-1

prtid specifies the ID of the IPDS printer on which the graphics file is to be printed. The default is the current session printer.

FILE specifies that the graphics data is in a disk file.

graphics file name specifies the name of the graphics object file to be printed.

width specifies the width, in inches, of the graphics area to be printed. This parameter cannot contain more than 5 characters specified in decimal numbers. The specified value cannot be more than 45.50. For example, if the area to be printed is 13 inches, you can specify either 13 or 13.0 for this parameter. If this parameter is not specified, 13.2 is assumed.

QRY Procedure

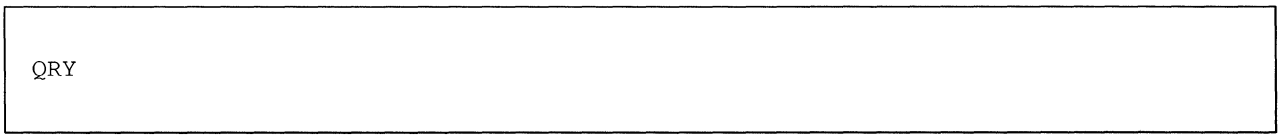
The QRY procedure calls Query/36 to allow you to produce a variety of reports using data from files. The query itself is a question, and the report created from the query is the answer to the question.

You can create a new query, and revise, copy, delete, browse, and run an existing query. A query report can be displayed or printed. You can also print a query definition. You can select and sort the data in the report and perform arithmetic operations on it. You can also write the query output to a file.

For more information about Query/36, see the online information. To read the online information:

- Run the READINFO procedure. For Query/36, specify:

  ```
  READINFO #QRYDOC,#QRYFLDR
  ```



S9020187-0

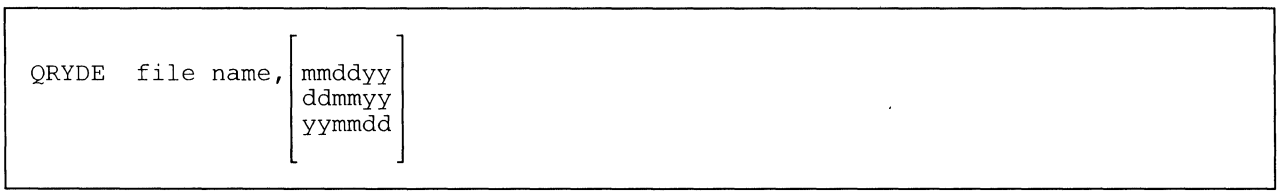
The QRY procedure has no parameters.

QRYDE Procedure

The QRYDE procedure allows you to enter new data in a file or existing data in a file to be updated one record at a time using the Query/36 data entry facility.

Notes:

1. *The program product Query/36 must be loaded (installed) on the system for this procedure to work.*
2. *The file you wish to enter data in or update must be linked to an IDDU file definition.*
3. *The date must be in the session format.*



S9020585-0

file name specifies the name of the disk file you want to enter or update. The file you specify must be linked to an IDDU file definition. See the "IDDULINK Procedure" on page 4-224 for more information.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file. The date, if specified, must be in the session date format. If a date is not specified and more than one file exists with the same file name, the most recent file created with that date is used.

QRYLOAD Procedure

The QRYLOAD procedure creates a library named #QRYLIB and copies the Query/36 support from diskette into that library. QRYLOAD copies additional support into the system library (#LIBRARY). The QRYLOAD procedure can copy diskettes obtained through software distribution diskettes created by the QRYSAVE procedure. See the "QRYSAVE Procedure" on page 4-367 for information about how to save the Query/36 support on diskette.

The QRYLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the QRYLOAD procedure to restore support that has been saved by QRYSAVE.

If Query/36 is not currently on the system, you must use the TOLIBR procedure to copy the diskette file QRY into #LIBRARY before running QRYLOAD.

If #LIBRARY was backed up with Query/36 on the system and then replaced before QRYLOAD is run, you do not have to copy the diskette file QRY using the TOLIBR procedure.

```
QRYLOAD [ A1 ] , [ S1 ]
        [ A2 ]   [ S2 ]
        [ A3 ]   [ S3 ]
        [ A4 ]   [ M1.nn ]
                 [ M2.nn ]
```

S9020188-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #QRYLIB is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, #QRYLIB is placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #QRYLIB on disk and copy the Query/36 support from diskette.

```
QRYLOAD
```

QRYRUN Procedure

The QRYRUN procedure runs a query that has already been defined and sends the report produced to the device specified. You can also display data in a file without defining a query.

For more information about Query/36, see the online information. To read the online information:

- Run the READINFO procedure. For Query/36, specify:

```
READINFO #QRYDOC,#QRYFLDR
```

To run a query and display the report:

```
QRYRUN query name,[library name],[file name],DISPLAY,,,,,,,,[NORECSEL/RECSEL],

      [DETAIL/SUMMARY]
```

S9020189-2

To run a query and print the report:

```
QRYRUN query name,[library name],[file name],PRINTER,[printer id],

      [form width],[line spacing],[copies],[forms number],,,[NOPRINT/PRINT],

      [NORECSEL/RECSEL],[DETAIL/SUMMARY]
```

S9020190-2

QRYRUN

To run a query and send the output to disk:

```
QRYRUN query name,[library name],[file name],DISK,,,,,[output file name],

      [NEW     ],[NOPRINT],[NORECSEL],[DETAIL ]
      [REPLACE ],[PRINT  ],[RECSEL  ],[SUMMARY]
```

S9020191-3

To display data in a file without defining a query:

```
QRYRUN ,,file name,DISPLAY,,,,,,,,,[NORECSEL],[DETAIL ]
                                   [RECSEL  ],[SUMMARY]
```

S9020484-2

To print a report without defining a query:

```
QRYRUN ,,file name,PRINTER,[printer id],[form width],[line spacing],[copies],

      [forms number],,,[NOPRINT],[NORECSEL],[DETAIL ]
                       [PRINT  ],[RECSEL  ],[SUMMARY]
```

S9020512-2

To send output to disk without defining a query:

```
QRYRUN ,,file name,DISK,,,,,,[output file name],[NEW     ],[NOPRINT],[NORECSEL],
                             [REPLACE ],[PRINT  ],[RECSEL  ]

      [DETAIL ]
      [SUMMARY]
```

S9020513-3

query name specifies the query to be run. The query must be previously defined. If no parameter is specified, you must specify a file name.

library name specifies the library that contains the query to be run. If no parameter is specified, the current library is assumed.

file name specifies the disk file to be queried. The file must be linked to the same file definition as the file name specified when the query was defined. If no parameter is specified, the file that was specified when the query was defined is assumed. If a file name is specified and no query name is specified, and the file is linked to a definition, the data in the file is displayed, printed, or sent to disk.

DISPLAY, PRINTER, or DISK specifies where the report or output produced by the query is to be sent. If no parameter is specified, the output device that was specified when the query was defined is assumed. If no output device was specified in the query, or if you do not specify a query name, DISPLAY is assumed.

DISPLAY specifies that the report or output produced by the query is to be sent to the display station that runs the procedure.

PRINTER specifies that the report or output produced by the query is to be printed.

DISK specifies that the output produced by the query is to be written to a disk file.

printer id specifies the work station ID of the printer on which the report is to be printed. If no parameter is specified, the printer that was specified when the query was defined is assumed. If no printer was specified in the query, or if you do not specify a query name, the session printer is assumed.

form width specifies the width of the forms on which the report is to be printed. You can enter a number from 60 to 198. If no parameter is specified, the form width that was specified in the query is assumed. If no form width was specified in the query, or if you do not specify a query name, 132 is assumed.

line spacing specifies the number of blank lines to leave between lines in the report. You can enter a number from 1 through 3.

- 1 indicates single spacing (no blank lines)
- 2 indicates double spacing (1 blank line)
- 3 indicates triple spacing (2 blank lines)

If no parameter is specified, the line spacing that was specified in the query is assumed. If no line spacing was specified in the query, or if you do not specify a query name, 1 is assumed.

copies specifies the number of copies of the report to be printed. You can enter a number from 1 through 255. If no parameter is specified, the number of copies that was specified in the query is assumed. If no number was specified in the query, or if you do not specify a query name, 1 is assumed.

forms number specifies the name of the particular form on which you want the report to be printed. You can enter from 1 through 4 characters. When the report is ready to print, a message at the subconsole asks if that particular form is on the printer. The operator can then check the printer and change forms, if necessary, before printing.

The forms number cannot contain the following characters: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), equal signs (=), and hyphens (-). If no parameter is specified, the forms number that was specified in the query is assumed. If no number was specified in the query, or if you do not specify a query name, no number is assumed.

output file name specifies the disk file that is to contain the query output. If no parameter is specified, the file that was specified in the query is assumed. If no file was specified in the query and the query output was sent to disk, #QRYOUT is assumed. If you do not specify a query name, no file is assumed.

NEW or REPLACE specifies whether the report is to be written to a new disk file or is to replace the contents of an existing file. If no parameter is specified, the NEW or REPLACE option that was specified in the query is assumed. If no option was specified in the query, or if you do not specify a query name, NEW is assumed.

NEW specifies that the report is to be written to a new disk file.

REPLACE specifies that the report is to replace the contents of a disk file that already exists on the system.

NOPRINT or PRINT specifies whether or not the query definition is to be printed when the query is run. If no parameter is specified, the NOPRINT or PRINT option that was specified in the query is assumed. If no option was specified in the query, or if a query name is not specified, NOPRINT is assumed.

NOPRINT specifies that the query definition is not to be printed when the query is run.

PRINT specifies that the query definition is to be printed when the query is run. PRINT cannot be specified if DISPLAY is the fourth parameter.

NORECSEL or RECSEL specifies whether or not the query is to be run with a run time selection test. If no parameter is specified, NORECSEL is assumed.

NORECSEL specifies that the query is to be run without a run time record selection test.

RECSEL specifies that record selection tests are to be changed for this run only. A display is shown on which you can change the record selection tests defined in the query or specify record selection tests if a query was not specified.

DETAIL or SUMMARY specifies the type of output to be produced by the query. If no parameter is specified, the DETAIL or SUMMARY option that was specified in the query is assumed. If no option was specified in the query, or if a query name is not specified, DETAIL is assumed.

DETAIL specifies that the output produced by the query is to be a report containing detail records and summary records if any exist.

SUMMARY specifies that the output produced by the query is to be a report containing summary records only.

QRYSAVE Procedure

The QRYSAVE procedure copies the Query/36 support to diskette. The Query/36 support from the libraries #QRYLIB and #LIBRARY is copied. You should use the "QRYLOAD Procedure" on page 4-362 to load the Query/36 support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPQRY.

```
QRYSAVE
```

S9020192-1

The QRYSAVE procedure has no parameters.

READINFO Procedure

The READINFO procedure displays an online document in its final form. The document may be supplied with an IBM program product, or created by you using DW/36.

For the procedure to function, the support to read online documents must be configured on your system.

```
READINFO document name,folder name,[help text label]
```

S9020193-0

document name specifies the document you want to display. The documents supplied by IBM program products are:

- #IDDDOC (for the interactive data definition utility)
- #WPDOC (for DisplayWrite/36)
- #QRYDOC (for Query/36)
- #OFCDOC (for Personal Services/36)
- #DSUDOC (for the development support utility)

folder name specifies the folder that contains the document to be displayed. The folders that contain the documents supplied by IBM program products are:

- #IDDFLDR (for the interactive data definition utility)
- #WPFLDR (for DisplayWrite/36)
- #QRYFLDR (for Query/36)
- #OFCFLDR (for Personal Services/36)
- #DSUFLDR (for the development support utility)

help text label specifics a label within the document where you want it to be opened. If no parameter is specified, the document is opened to the first page.

REBLD Procedure

The REBLD procedure is supported only for compatibility with the IBM System/34. (System/36 does not allow you to specify hexadecimal data in a procedure, however, so the ninth parameter in the REBLD procedure cannot contain hexadecimal data.) See the "COPYDATA Procedure" on page 4-111 for information on copying disk files.

RELOAD Procedure

The IBM System/34 RELOAD procedure is not supported. See the "RESTLIBR Procedure" on page 4-386 for information about restoring (reloading) the system library. See the "SAVELIBR Procedure" on page 4-431 for information about saving the system library.

REMOVE Procedure

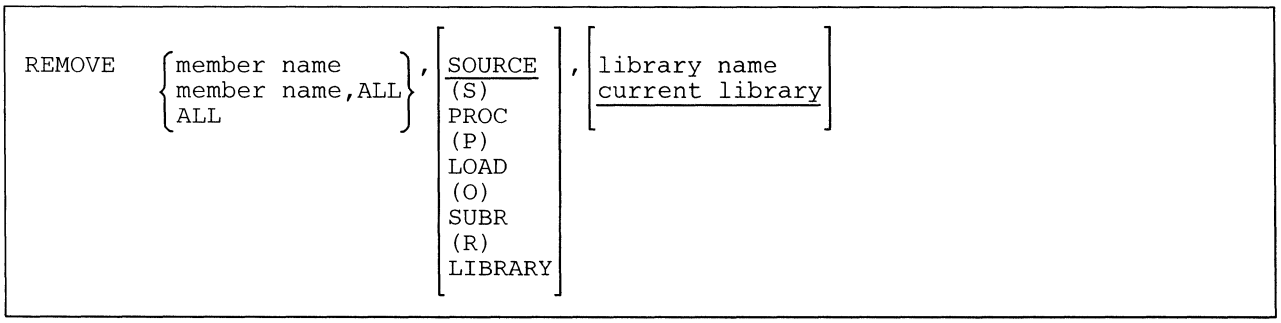
The REMOVE procedure removes one or more specified library members from a library. IBM-supplied library members cannot be removed using this procedure. To remove IBM-supplied library members, see "Remove Members from a Library (REMOVE Procedure)" on page A-78.

The space that was occupied by the removed members can be used again only if one of the following is true:

- The program attempting to use the space is the only program using the library, and no members physically follow the removed members in the library.
- The CONDENSE procedure is run to compress the library. See the "CONDENSE Procedure" on page 4-109.

To remove files or entire libraries from disk or diskette, see the "DELETE Procedure" on page 4-144.

The REMOVE procedure runs the $MAINT utility program.



member name specifies the library member to be removed.

member name,ALL specifies the beginning characters of the names of one or more library members to be removed. Up to 7 characters can be used.

ALL specifies that all library members are to be removed. When ALL and LIBRARY are both specified, no other operators can be using that library; if the library is the system library (#LIBRARY), no other jobs can be running.

SOURCE or S specifies that only library source members are to be removed. If no parameter is specified, SOURCE is assumed.

PROC or P specifies that only library procedure members are to be removed.

LOAD or O specifies that only library load members are to be removed.

SUBR or R specifies that only library subroutine members are to be removed.

LIBRARY specifies that all library types (SOURCE, PROC, LOAD, and SUBR) are to be removed.

library name specifies the library containing the members to be removed. If no library name is specified, the current library is assumed.

Example 1

Remove the procedure member named PAYROLL from the library called MYLIB.

```
REMOVE PAYROLL,PROC,MYLIB
```

Example 2

Remove all system library members that are named SAM.

```
REMOVE SAM,LIBRARY,#LIBRARY
```

Example 3

Remove all library members beginning with the characters PAY from the library named YOURLIB.

```
REMOVE PAY,ALL,LIBRARY,YOURLIB
```

Example 4

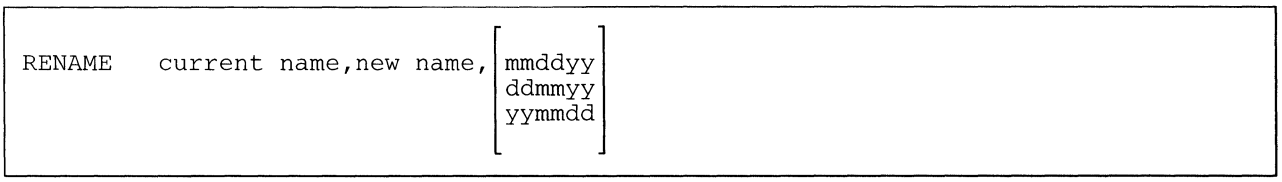
Remove a source member named THIS, a procedure member named THAT, and all source members beginning with THEM from the library MYLIB (which is the current library).

```
REMOVE THIS
REMOVE THAT,P
REMOVE THEM,ALL
```

RENAME Procedure

The RENAME procedure changes the name of an existing disk file, library, or folder. A disk file, library, or folder can be renamed only if it is not currently being used. The RENAME procedure can be used to rename files on a remote system by specifying the local file label as it is defined in your system's network resource directory. If RENAME is being used to rename a remote file, the new file label must be defined in the directory as well as the old file label. The entries for the two files must indicate that the files are at the same remote locations. The system library (#LIBRARY), the system files (such as the history file), job files, and scratch files cannot be renamed.

The RENAME procedure runs the $RENAM utility program.



S9020195-0

current name specifies the current name of the file, library, or folder.

new name specifies the new name for the file, library, or folder. The new name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because these have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), and hyphens (-).

The new name cannot be the name of another disk file, library, or folder, even if that file, library, or folder has a different creation date. Also, do not use ALL as a file name; ALL, DISK, F1, #LIBRARY, PRINT, TAPE, or READER as a library name; or ALL or #LIBRARY as a folder name.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file (libraries do not have a date). The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If the creation date is not specified and more than one file exists with the specified current name, the last file created is renamed. The creation date is not changed by the RENAME procedure.

Example 1

Change the name of an existing file from OLDPAY to NEWPAY.

```
RENAME OLDPAY,NEWPAY
```

Example 2

Change the name of an existing file from OLDPAY to NEWPAY, change THIS to THAT, and change MYLIB to YOURLIB.

```
RENAME OLDPAY,NEWPAY
RENAME THIS,THAT
RENAME MYLIB,YOURLIB
```

REQUESTX Procedure

The REQUESTX procedure allows you to request or cancel an available user facility in an X.21 public data network. Depending on the network, registration for a variety of facilities or services can be done by service order at subscription time, or over the network itself by using the REQUESTX procedure. The REQUESTX procedure is described in the manual *Using System/36 Communications*.

The REQUESTX procedure runs the #GCFR utility program.

To have the system prompt for the specifications:

```
REQUESTX
```

S9020196-0

To have the system process the specifications contained in a library source member:

```
REQUESTX [source member name] , [library name      ] , [NOENTRY]
                                [current library   ]   [ENTRY  ]
```

S9020197-1

source member name specifies the library source member that contains registration and cancelation specifications. If a source member name is not specified, the REQUESTX procedure will prompt for the specifications.

library name specifies the library that contains the source member. If no parameter is specified, the current library is assumed.

NOENTRY specifies that the source entry utility (SEU) is not to be called, and that the specifications in the indicated source member are to be transmitted.

ENTRY specifies that the source entry utility (SEU) is to be called to create or change the source member before the specifications are transmitted by the REQUESTX procedure.

Example 1

Start the REQUESTX procedure, which will then prompt for the requests.

```
REQUESTX
```

Example 2

Start the REQUESTX procedure, which will then process the library source member REQ from the current library (named MYLIB).

```
REQUESTX REQ
```

RESPONSE Procedure

The RESPONSE procedure updates the automatic response values and the severity levels for messages in a message load member. Both IBM-supplied messages and user-written messages can be assigned these values. Many IBM-supplied messages have severity levels, and many have automatic responses. These values are indicated with the message in the following message manuals:

- *System Messages*
- *RPG II Messages*
- *Utilities Messages*
- *COBOL Messages*
- *Assembler Messages*
- *BASIC Messages*
- *MSRJE Messages*
- *3270 Messages*
- *Character Generator Utility Guide*
- *Ideographic Sort Guide*

The severity level is used to determine whether the system should automatically respond to a displayed message.

The NOHALT procedure or OCL statement must be run in order for the RESPONSE procedure to be effective. The system, session, or job severity level at which the system automatically responds is set by the NOHALT procedure or OCL statement; see the "NOHALT Procedure" on page 4-305 and the "NOHALT OCL Statement" on page 5-79 for more information about setting the severity level.

The input to the RESPONSE procedure is an automatic response source member that contains, for each message to be responded to: the alpha code, the message identification code (MIC), the response, and the severity level. The source member can be created using the Source Entry Utility (SEU), the Development Support Utility (DSU), or the $MAINT utility program.

The changes made by the RESPONSE procedure remain in effect until changed by another RESPONSE procedure or until a new release is installed for IBM-supplied messages. Changes made to a user message member remain in effect until changed by another RESPONSE procedure or until the load member is replaced by the CREATE procedure. See the "NOHALT Procedure" on page 4-305 for information about how long the NOHALT procedure remains in effect. See "Automatic Response Programming Considerations" on page 4-378 for other considerations.

The automatic response source member contains three types of statements. See the "Automatic Response Source Statements" on page 4-376 for information about the format of these statements.

- The automatic response control statement
- One or more automatic response specification statements
- One or more comment statements (optional)

The RESPONSE procedure runs the $ARSP utility program.

```
RESPONSE source member name,[ library name      ]
                            [ current library   ]
```

S9020198-0

source member name specifies the source member that contains the automatic response control statements, specification statements, and comment statements.

library name specifies the library that contains the source member. If a library name is not specified, the current library is assumed.

Example

To apply the automatic responses contained in the library source member named AUTORESP (which is stored in a library named MYLIB).

```
RESPONSE AUTORESP,MYLIB
```

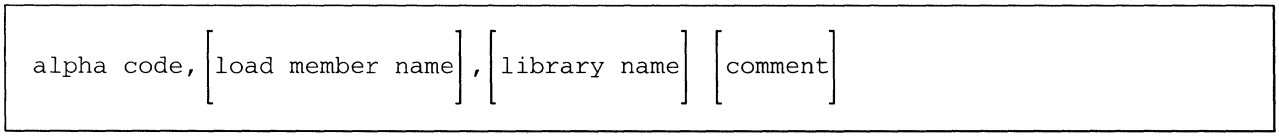
Automatic Response Source Statements

The Automatic Response Control Statement

The automatic response control statement specifies the alpha code for the automatic response specification statements that follow. The alpha codes are the two, three, or four characters that are displayed before the MIC number. For example, if message SYS-1395 is displayed, SYS is the alpha code and 1395 is the MIC number. This statement must come before the automatic response specification statements. A source member may contain more than one automatic response control statement and each one can have its own automatic response specification statement and comment statements.

The RESPONSE procedure will process both nonideographic and ideographic data in the case where a message may exist in both nonideographic and ideographic form.

The syntax of the automatic response control statement is:



S9020199-0

alpha code specifies the alpha code of the messages that are being given automatic response values. The alpha code must be one of the following:

ASM	DSU	IWS	SDA	USER
BAS	EMU	KBD	SEU	WSU
BGU	EP	NRD	SORT	
CBL	ESU	OFC	SRTX	
CGU	ET	QRY	SYS	
DFU	FORT	RJE	TTM	
DHCF	IDDU	RPG	TXT	

You should group the messages with the same alpha codes together in the source member. This allows you to update the source member more easily.

load member name specifies the message load member whose automatic response values are to be updated. If the alpha code is USER, the member name must be specified. If the alpha code is not USER, the member name parameter is ignored and the $ARSP utility determines the member name.

library name specifies the library that contains the message load member. If the alpha code is USER, the library name can either be specified or not specified; if it is not specified, the system library (#LIBRARY) is assumed. If the alpha code is not USER, the library name parameter is ignored and the $ARSP utility determines the library name.

comment specifies any information that may be helpful to identify the message or the response you want. This information is not used by the system. One or more blanks must be placed before the comment.

Automatic Response Specification Statement

The automatic response specification statement specifies the message identification code (MIC) of the message to be updated, and the automatic response and severity level values for that message.

If an automatic response specification statement is specified for a message that is not in the specified message member, an error message is displayed.

The format of the automatic response specification statement is:

```
mic   response,severity level  [comment]
```

S9020200-0

mic specifies the message identification code (MIC) of the message. The MIC must be a 4-digit number from 0000 through 9999; it must be placed in positions 1 through 4 of the message text statement. The MICs must be in **ascending numerical** order.

response specifies the automatic response value for the message. The allowed values are: 0, 1, 2, 3, D, or N. If N is specified, the IBM-supplied automatic response value is used for the specified MIC. The automatic response must be placed in position 6 in the statement. See the appropriate message manual for a description of the responses for the value you want to update.

When a job is automatically canceled by a 3 option, a message is displayed indicating the job was canceled by the system.

severity level specifies the severity level for the message. The allowed values are 1, 2, 3, 4, and 5. The value must be placed in position 8 in the statement.

You can specify that only messages with a specific severity level are to be automatically responded to; see the "NOHALT Procedure" on page 4-305 and the "NOHALT OCL Statement" on page 5-79 for how to specify this severity level for the system, your session, or a job.

The severity-level guidelines used for the IBM-supplied messages are as follows:

1 Informational messages (option 0 only).

2 Messages with one option; or messages with two options where one option is a retry option.

3 Program error messages; these usually have more than one option.

4 Messages for severe errors, such as hardware errors or permanent input/output errors.

5 No automatic response value is allowed for this message.

comment specifies any information that may be helpful to identify the message or the response you want. This information is not used by the system. One or more blanks must be placed before the comment.

Comment Statement

Specifies any information that may be helpful to identify the message or the response you want. This information is not used by the system. The format of the comment statement is as follows:

```
* comment
```

S9020201-0

The asterisk (*) must be the first character in the statement. Comment statements can be placed anywhere within the other statements.

Automatic Response Programming Considerations

You should not specify automatic response values to messages that:

- Indicate permanent input/output errors
- Indicate a dump has been taken
- Require an operator to do something before the message is responded to
- Indicate a retry option
- Are automatic response facility (the RESPONSE procedure) error messages

When you choose values, make sure that they do not result in lost data or fill the history file with repetitive messages.

Consider the following when you are assigning values to messages:

- Messages are automatically responded to only if the following is true:
  - The message requires a response.
  - The automatic response value is a value that the message allows.
- Both the message and the response are logged to the history file.
- You may want to maintain two source members. One of these would contain your automatic response values; the other would contain the response value N (in column 6 of all automatic response specification statements).

  By maintaining two source members, you can use one for your automatic response values and the other for the IBM-supplied automatic response values. To reset the IBM-supplied automatic response values, you would run the RESPONSE procedure using the source member that has the N in column 6.

Example Automatic Response Source Member

Assume a source member contains the following statements. Note how the statements are grouped by the alpha codes and the MICs are listed in numerical order.

```
* Automatic responses for SSP displayed messages
SYS
1051 3,3 Invalid LPI parameter
1063 3,3 Invalid library name
1272 3,3 File statement cannot have both RETAIN-J and JOB-YES
*
* Automatic responses for RPG displayed messages
RPG
9011 0,3  Square root of negative number
9013 0,3  Divide by zero
9016 0,3  No data found
9037 0,3  Index key not in sequence
*
* Automatic responses for USER displayed messages
* The message load member is named MESSAGES,
* and is contained in the library MYLIB
USER,MESSAGES,MYLIB
0001 2,3  Error in customer number
0012 0,2  Error in item number
```

You could use the following source member to return the responses back to the IBM-supplied responses:

```
* Automatic responses for SSP displayed messages
SYS
1051 N     Invalid LPI parameter
1063 N     Invalid library name
1272 N     File statement cannot have both RETAIN-J and JOB-YES
*
* Automatic responses for RPG displayed messages
RPG
9011 N      Square root of negative number
9013 N      Divide by zero
9016 N      No data found
9037 N      Index key not in sequence
```

RESTEXTN Procedure (for the Ideographic Version of the SSP)

The RESTEXTN procedure restores from diskette all or part of the extended character file. The restored file is placed in a disk file labeled either #EXT1818 or #EXT2424. The extended character file is divided into two parts, the IBM-supplied extended characters and the user-defined extended characters. You can restore either or both of these parts. You can also restore specified portions of the user-defined extended characters.

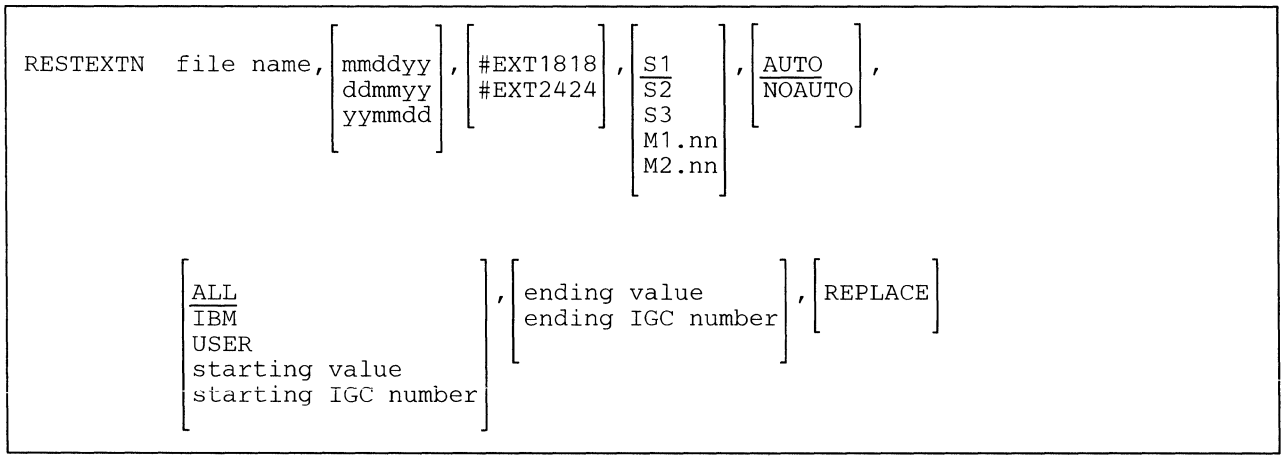
The size of the file is determined by what part of the extended character file is restored. If only IBM-supplied characters are requested to be restored, then the file is allocated large enough to hold those characters. If user-defined characters are requested to be restored, the file is allocated large enough to contain all user-defined characters or the user-defined characters specified. Also, you can specify the size for the file with the RECORDS or BLOCKS parameter on the FILE OCL statement. If the size you specify is not large enough to contain the characters, an error message is displayed.

The RESTEXTN procedure can be run on files saved by the SAVEEXTN procedure or any properly formatted I-exchange files from another system. Also, the RESTEXTN procedure can be run on files saved by the XSAVE procedure on a System/34. If password security is active, the RESTEXTN procedure can be run from any display station, but the operator must have system operator authority to run the RESTEXTN procedure.

You can use the character generator utility to enter ideographic characters in the extended character file. For information on how to use the character generator utility, see the *Character Generator Utility Guide.*

The RESTEXTN procedure runs the $XREST utility program.

Note: The RESTEXTN procedure supports only Japanese extended character files. Extended character files from Korea, Taiwan, or People's Republic of China are not supported.



S9020202-0

file name specifies the name of the diskette file containing the extended character file.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the diskette file. The date specified must be in the same format as the date that was specified when the diskette file was created.

#EXT1818 or #EXT2424 specifies the name for the file that will be created on disk. The name must be either #EXT1818 or #EXT2424. If this parameter is not specified, the file name from the first parameter is assumed. Therefore, an error message is issued if the file name in the first parameter is something other than #EXT1818 or #EXT2424.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

ALL specifies that the entire extended character file is to be restored to disk. If this parameter is not specified, ALL is assumed.

IBM specifies that only the IBM-supplied extended characters are to be restored to disk.

USER specifies that only the user-defined extended characters are to be restored to disk.

starting value specifies with which user-defined extended character to begin restoring to disk. The code point value must be for a user-defined extended character and not an IBM-supplied extended character. The value must be specified in the following format: Xdddd, where dddd is the code point value. For more information on code point values for extended characters, refer to the *Character Generator Utility Guide*.

starting IGC number specifies with which user-defined extended character to begin restoring to disk. IGC number is the 5-digit decimal number that was assigned when the character was created using CGU. For more information on the IGC number, refer to the *Character Generator Utility Guide*.

ending value Specifies with which user-defined extended character to stop restoring to disk. The code point value must be for a user-defined extended character and not a IBM-supplied extended character. The value must be specified in the following format: Xdddd, where dddd is the code point value. For more information about code point values for extended characters, refer to the *Character Generator Utility Guide.* If this parameter is not specified, and IBM was specified for the previous parameter, only the IBM-supplied extended characters will be restored. If this parameter is not specified, and any value other than IBM was specified for the previous parameter, the last user-defined extended character in the extended character file is assumed.

ending IGC number specifies with which user-defined extended character to stop restoring to disk. IGC number is the 5-digit decimal number that was assigned when the character was created using CGU. For more information on the IGC number, refer to the *Character Generator Utility Guide.* If this parameter is not specified, and IBM was specified for the previous parameter, only the IBM-supplied extended characters will be restored. If this parameter is not specified, and any value other than IBM was specified for the previous parameter, the last user-defined extended character in the extended character file is assumed.

REPLACE specifies that if extended characters already exist with the same code point value or IGC number, they are to be replaced. If REPLACE is specified, new extended characters replace existing extended characters with the same code point value or IGC number, and no messages regarding the replacements are displayed.

If REPLACE is not specified, extended characters are placed into the file until a character with the same code point value or IGC number is found, at which time the system displays a message telling the operator that a character has already been defined. In response to the message, the operator can either cancel the job, cancel the job step, continue processing without replacing the character, or continue processing by replacing the character. If other characters with the same code point value or IGC number are found during the job, the same message will be displayed.

Example

This example shows how to restore the entire extended character file, including user-defined extended characters, from diskette to disk. The name of the diskette file is #EXT2424. The same name is used for the disk file. The diskettes are located in magazine M1. Any duplicates encountered should be automatically replaced during the restore.

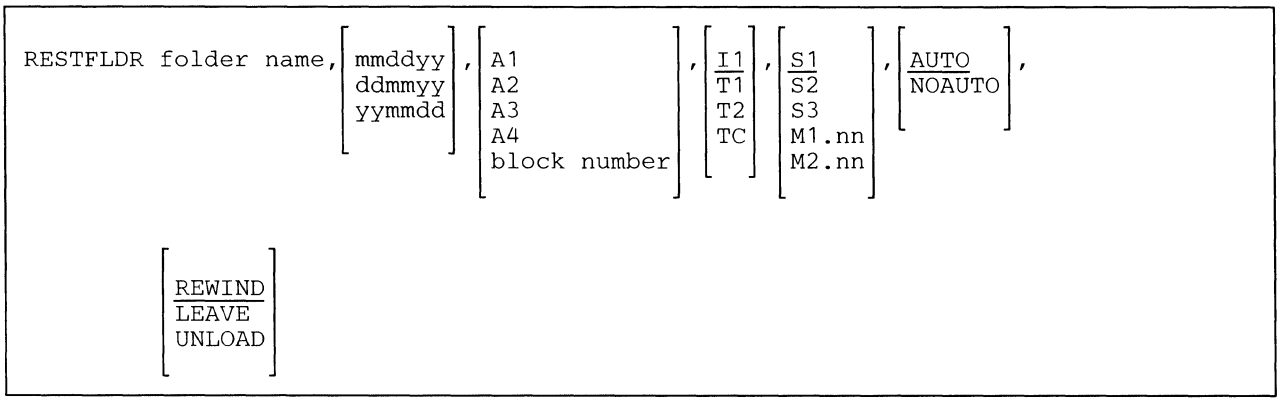
```
RESTEXTN #EXT2424,,,M1,,,,REPLACE
```

RESTFLDR Procedure

The RESTFLDR procedure restores a folder copied onto disk, diskette, tape, or tape cartridge by the SAVEFLDR procedure. You can also specify a preferred location on disk for the folder.

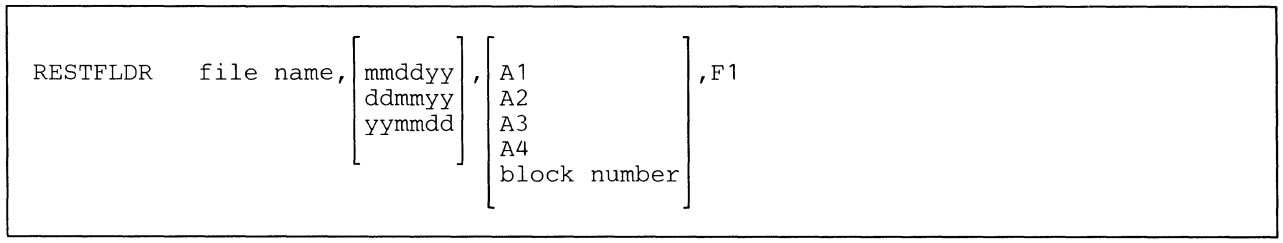
The RESTFLDR procedure runs the $TMSERV utility program.

To restore a folder from diskette, tape, or tape cartridge:



S9020485-1

To restore a folder from disk:



S9020563-0

folder name specifies the folder to be restored.

file name specifies the name of the file that contains the saved folder on disk.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the folder to be restored. The date must be specified in the same format as the creation date of the disk, diskette, or tape file.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, other units (if they exist) are checked, and the folder is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, the folder is placed on the least-used disk unit.

block number specifies the location of the first block of the folder. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk.

I1 specifies that the folder is to be restored from diskette. If no parameter is specified, I1 is assumed.

F1 specifies that the folder is to be restored from a disk file.

T1, T2, or TC specifies that the folder is to be restored from tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed. This parameter is not allowed if T1, T2, or TC is specified.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is not allowed if T1, T2, or TC is specified.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive is used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the RESTFLDR procedure has run. This parameter is valid only if T1, T2, or TC is specified. If no parameter is specified, REWIND is assumed.

LEAVE specifies that the tape should be left where it is after the RESTFLDR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is valid only if T1, T2, or TC is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the RESTFLDR procedure has run. This parameter is not valid if I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

Example 1

This example shows a folder named MYFLDR being restored from diskette.

```
RESTFLDR MYFLDR
```

Example 2

This example shows a folder named MYFLDR being restored from tape. The folder is stored on a tape reel that is mounted on tape drive 1. After the folder has been restored, the tape is rewound and unloaded.

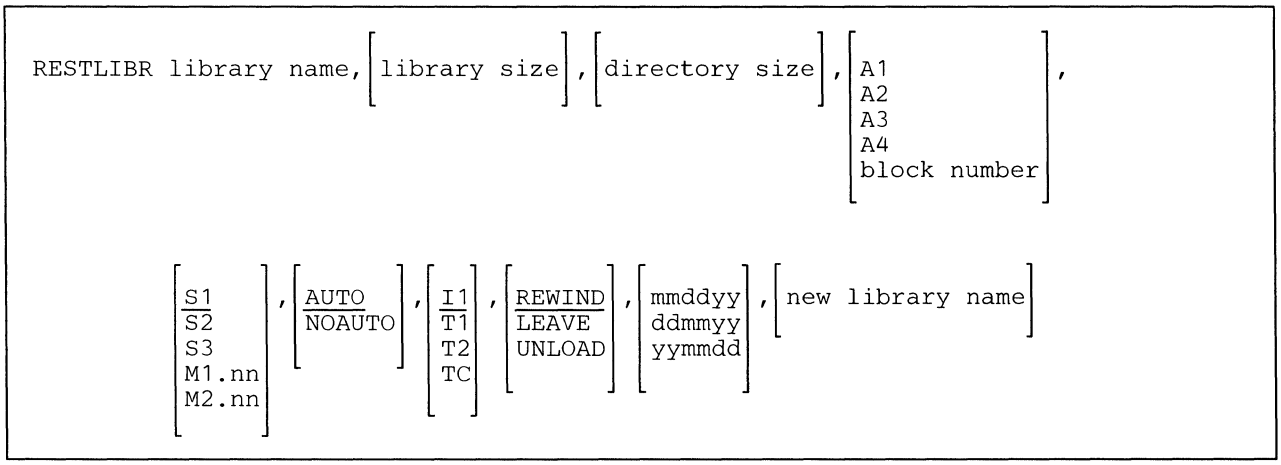
```
RESTFLDR MYFLDR,,,T1,,NOAUTO,UNLOAD
```

RESTLIBR Procedure

The RESTLIBR procedure restores a library copied onto diskette, tape, or tape cartridge by the SAVELIBR procedure. You can change the size of a user library, change the directory size of a user library, or specify a new library name that is different from the one on the diskette, tape, or tape cartridge. You cannot change the size or the directory size of the system library (#LIBRARY) using the RESTLIBR procedure. You cannot specify a preferred location on disk for #LIBRARY and you cannot specify a new name for #LIBRARY. For information about changing the size or directory size of #LIBRARY, see the manual *Changing Your System Configuration*.

To copy individual members to a library from a disk, diskette, or tape file, see the "TOLIBR Procedure" on page 4-537. (Files copied by TOLIBR must have been created either by the FROMLIBR procedure or by the $MAINT utility program.) To restore data files, see the "RESTORE Procedure" on page 4-392.

The RESTLIBR procedure runs the $MAINT utility program.



S9020203-2

library name specifies the library that is to be restored. A library name must be specified. If the system library (#LIBRARY) is being restored, the release level of the saved copy must match the copy on the system.

If the library specified already exists, a message is displayed that allows you to:

- Cancel the RESTLIBR procedure.
- Delete the library from the disk and proceed with the RESTLIBR procedure.

library size specifies the new size, in blocks, of the library. One block contains 2560 bytes. The maximum library size is 15000 blocks. If a size is not specified, the size that was saved with the library is used.

directory size specifies the new size, in sectors, of the directory for the library. The minimum directory size is 2 sectors; the maximum directory size is 2500 sectors. One library block contains 10 sectors. If a directory size is not specified, the size that was saved with the library is used.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, other units (if they exist) are checked, and the file is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, the library is placed on the least used disk unit.

block number specifies the location of the first block of the library. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed. This parameter is not allowed if T1, T2, or TC is specified.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is not allowed if T1, T2, or TC is specified.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive is used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

I1 specifies that the library is to be restored from diskette. If no parameter is specified, I1 is assumed.

T1, T2, or TC specifies that the library is to be restored from tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

REWIND specifies that the tape should be rewound after the RESTLIBR procedure has run. REWIND is assumed if this parameter is not specified. This parameter is not valid if I1 is specified.

LEAVE specifies that the tape should be left where it is after the RESTLIBR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is not valid if I1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the RESTLIBR procedure has run. This parameter is not valid if I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file to be restored. The date must be specified in the same format as the creation date of the diskette or tape file.

new library name specifies the name of the library to be restored if you change the library name from the one specified on the diskette, tape, or tape cartridge. A library name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because these have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), minus signs (-), and equal signs (=). Do not user #LIBRARY, F1, READER, PRINT, DISK, TAPE, or ALL as a library name.

Example 1

This example shows the system library being restored.

```
RESTLIBR #LIBRARY
```

Example 2

This example shows a library named PAYLIB being restored. The size of the library and the size of the directory are not being changed.

```
RESTLIBR PAYLIB
```

Example 3

This example shows a library named PAYLIB being restored. The size of the library is being changed to 300 blocks. The size of the directory is being changed to 30 sectors.

```
RESTLIBR PAYLIB,300,30
```

Example 4

This example shows a library named PAYLIB being restored, preferably to disk A2. The size of the library and the size of the directory are not being changed. The library is stored on a tape reel that is mounted on tape drive 1. After the library has been restored, the tape is rewound and unloaded.

```
RESTLIBR PAYLIB,,,A2,,,T1,UNLOAD
```

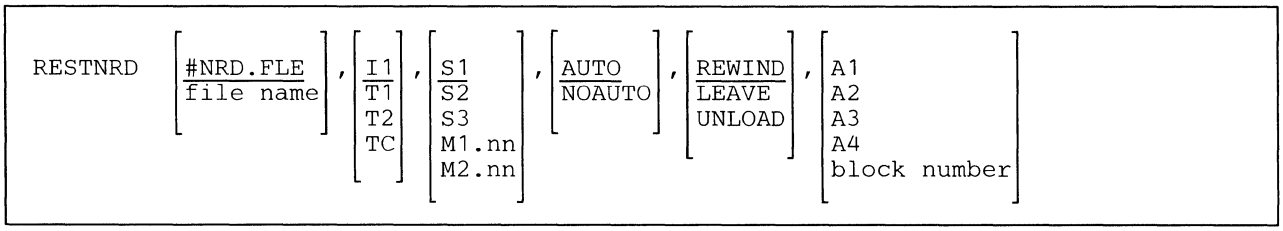
RESTNRD Procedure

The RESTNRD procedure restores a version of the network resource directory from diskette, tape, or tape cartridge to disk. The version to be restored must have been saved (as a file) by the SAVENRD procedure.

The system assigns the name #NRD.FLE to the directory file when it is restored to disk. If the directory file already exists on disk, you must use the DELNRD procedure to remove it before you can restore another version of the directory on the system.

For more information about the network resource directory, see the *Distributed Data Management Guide.*

The RESTNRD procedure runs the $COPY and $SINCT utility programs.



S9020204-1

#NRD.FLE specifies that the file named #NRD.FLE on diskette or tape, which is the default file name assigned by the SAVENRD procedure, is to be restored to disk. If no parameter is specified, #NRD.FLE is assumed.

file name specifies the name of the file on diskette or tape that is to be restored to disk. This is the name under which the directory was saved. If more than one file exists with the specified name, the system restores the first file on the diskette or tape that it finds with that name.

I1 specifies that the file is to be restored from diskette. If no parameter is specified, I1 is assumed.

T1, T2, or TC specifies that the file is to be restored from tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates the tape is a tape cartridge.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed. This parameter is not allowed if T1, T2, or TC is specified.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is not allowed if T1, T2, or TC is specified.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01 and continues through M2.10.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive is used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the RESTNRD procedure has run. This parameter is valid only if T1, T2, or TC is specified. If no parameter is specified, REWIND is assumed.

LEAVE specifies that the tape should be left where it is after the RESTNRD procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is valid only if T1, T2, or TC is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the RESTNRD procedure has run. This parameter is not valid if I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, other units (if they exist) are checked, and the file is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, the file is placed on the least-used disk unit.

block number specifies the location of the first block of the file. Up to six digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk.

Example 1

This example restores the file named #NRD.FLE from the diskette located in slot S2 to whichever disk the system determines to use. (The file must have been saved using the SAVENRD procedure.) No network resource directory can be on the system when this file is restored.

```
RESTNRD ,,S2
```

Example 2

This example restores the backup file NRD.BKUP from tape to disk. The tape reel containing the backup file must be mounted on tape drive T1, and is to be left where it is when the directory has been restored. The directory is to be placed on the second disk if space is available.

```
RESTNRD NRD.BKUP,T1,,,LEAVE,A2
```

RESTORE Procedure

The RESTORE procedure restores a saved file, a set of saved files, or part of a set of saved files from diskette, tape, or tape cartridge to disk. One of the following must have created the diskette, tape, or tape cartridge file:

- The SAVE procedure
- The $COPY utility program

When only one file is to be restored, you can change the size of the file by specifying the RECORDS or BLOCKS parameter. You can also specify the preferred disk placement or the block number where the file is to begin by specifying the LOCATION parameter.

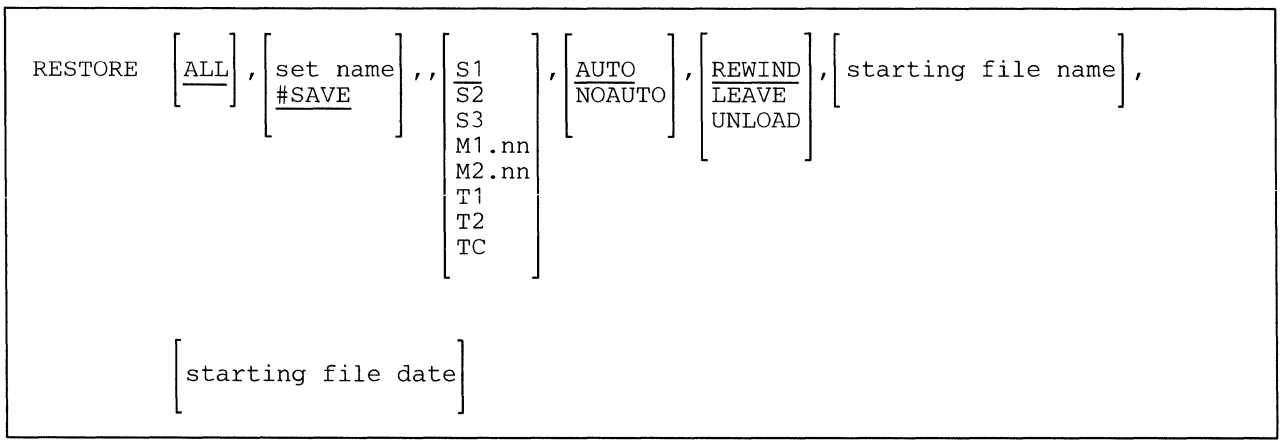
The RESTORE procedure can be used to restore local diskette files individually to a remote system. See the *Distributed Data Management Guide.*

The RESTORE procedure runs the $COPY utility program. $COPY will not process a library, a folder, system files such as the spool file, or a diskette, tape, or tape cartridge file that was not created by $COPY.

To restore libraries, see the "RESTLIBR Procedure" on page 4-386. To restore folders, see the "RESTFLDR Procedure" on page 4-383. To restore the network resource directory, see the "RESTNRD Procedure" on page 4-389. To copy basic data or I exchange diskette files to the disk, see the "TRANSFER Procedure" on page 4-542. To copy special E-format diskettes (those diskettes created by the 5260 Retail System), see the "POST Procedure" on page 4-345. To copy exchange tape files to the disk, see the "TAPECOPY Procedure" on page 4-500.

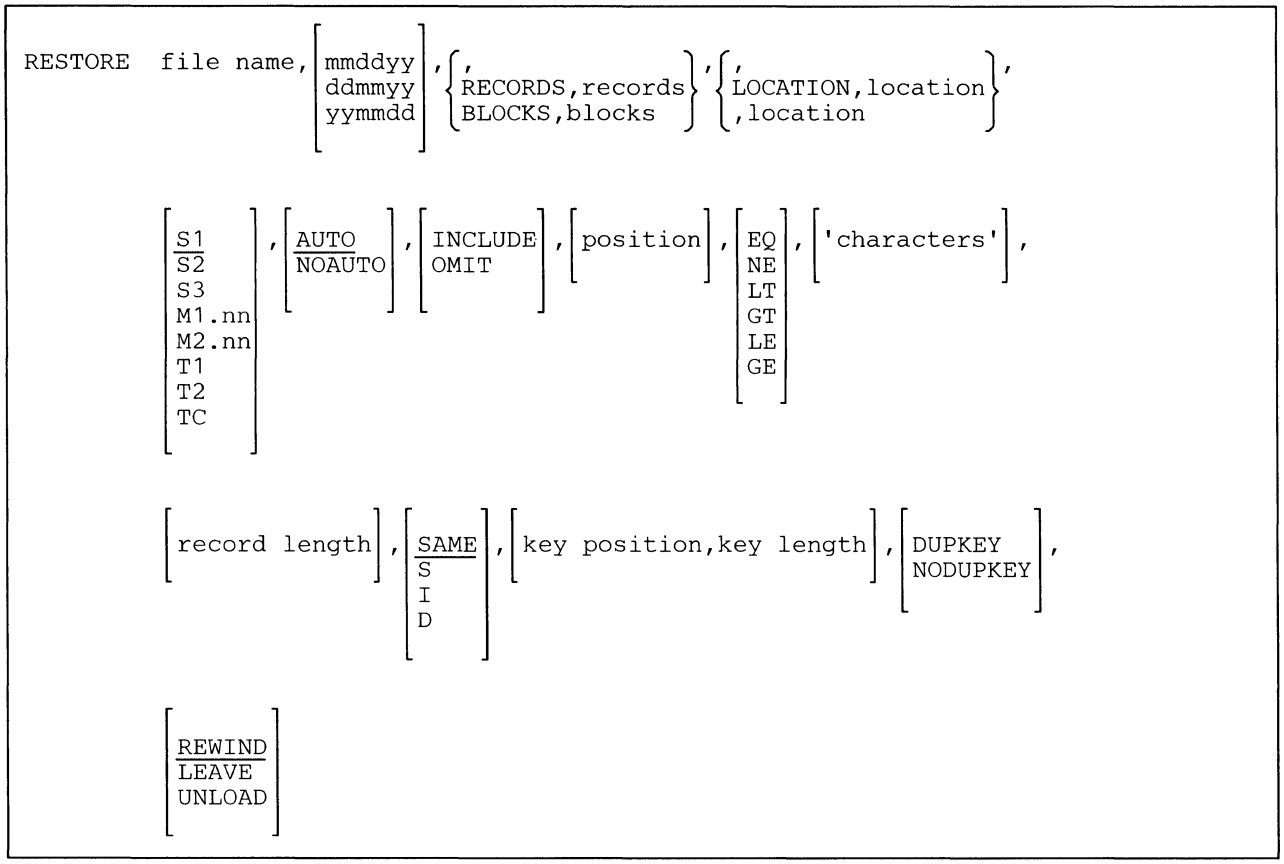
When you are copying all files within a set from diskette, tape, or tape cartridge back to disk, the data begins with the first file within the set on the first diskette, tape reel, or tape cartridge unless a different starting file (an optional starting file date) is specified. On diskette, the first file within the set is also the first file on the diskette because a set of files can only be saved on a diskette containing no unexpired files.

For restoring all previously saved data files:



S9020205-1

For restoring a single previously saved data file:



S9020206-1

Parameters 9 (INCLUDE/OMIT) through 17 (DUPKEY/NODUPKEY) are not valid when restoring from tape or tape cartridge.

ALL specifies that all files previously saved in a set are to be restored to the disk. If the first parameter is not specified, ALL is assumed.

set name specifies the name used for the entire set of files that was saved on diskette, tape, or tape cartridge by the SAVE (SAVE ALL) procedure. If no set name is specified on a RESTORE ALL statement, a set name of #SAVE is assumed.

file name specifies a single diskette, tape, or tape cartridge file that is to be restored to the disk. If more than one file exists with the specified name and if the creation date is not specified, the system restores the first file it finds on the diskette, tape, or tape cartridge with the specified name.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the diskette, tape, or tape cartridge file. The date must be in the same format as the session date. Use the STATUS SESSION control command to determine the date format.

RECORDS or BLOCKS specifies the size for the file. If BLOCKS or RECORDS and the associated size is not specified, the file is restored to its original size. If the files organization and/or record length is being changed, the restored file is made large enough to hold the total number of records in the diskette, tape, or tape cartridge file.

Note: When a single disk file is added to a previously saved diskette file and BLOCKS or RECORDS is not specified, the restore may not take place as a result of the add. This happens because you tried to restore more data into the size of the disk space allocated for the previously saved file before the add.

RECORDS specifies that the disk file is to be made large enough to contain the number of records specified. **records** can be any number from 1 through 8000000.

BLOCKS specifies that the disk file is to be made large enough to contain the number of blocks specified. **blocks** can be any number from 1 through the maximum number of blocks of disk storage configured on the system.

LOCATION specifies the preferred disk placement or the first block number for the file. If you do not specify LOCATION and an associated location, the SSP places the file on the least used disk.

location can be:

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and the file is placed on the least used disk unit. If no location is specified, the file is restored according to the disk preference of the file before it was saved if a disk preference existed. Otherwise, the file is placed on the least used disk unit.

block number specifies the location of the first block of the file. Up to 6 digits can be specified. You can use the CATALOG procedure to determine where blocks of disk space are available on disk.

For more information about file locations, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

T1, T2, or TC specifies the tape drive containing the first tape to be processed. T1 indicates the first tape drive, T2 indicates the second tape drive, and TC indicates the tape cartridge.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01 and continues through M2.10.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

INCLUDE or OMIT specifies whether specific records in the file to be copied are to be included in or omitted from the restored file. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters.

If only 'characters' or position is specified, INCLUDE and EQ are assumed.

position specifies, for each record, the first character to be compared with the comparison characters. The position can be any number from 1 through 4096. If a position is not specified, then every position in the record is compared with the comparison characters until the specified condition is met.

EQ specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is to be included in or omitted from the restored file.

NE specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record is to be included in or omitted from the restored file.

LT specifies that if the characters in the record indicated by position are less than the comparison characters, the record is to be included in or omitted from the restored file.

LE specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record is to be included in or omitted from the restored file.

GT specifies that if the characters in the record indicated by position are greater than the comparison characters, the record is to be included in or omitted from the restored file.

GE specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record is to be included in or omitted from the restored file.

'characters' specifies the comparison characters. Up to 30 characters can be specified, and they must be enclosed by apostrophes ('). Blanks and commas (,) can be included, but apostrophes cannot be included as data.

record length specifies the record length of the restored file, and can be any number from 1 through 4096. If this parameter is not entered, the record length of the file on diskette is used for the record length of the restored file.

If the record length of the file on diskette is less than the specified record length, the additional record positions in the restored file are filled with blanks. If the record length of the file on diskette is greater than the specified record length, the extra positions in the diskette file are truncated. If the restored file is an indexed file and the key field is truncated, a message is displayed to allow the operator to cancel the job.

SAME specifies that the restored file is to have the same organization as when the file was saved. If a parameter is not specified, SAME is assumed.

S specifies that the restored file is to be organized as a sequential file.

I specifies that the restored file is to be organized as an indexed file.

D specifies that the restored file is to be organized as a direct file.

key position specifies the starting position of the key for the restored file. The key position must be specified if the restored file is to be changed into an indexed file from a sequential or direct file (that is, when I is specified). Also, if you are restoring an indexed file, you can specify another field for the key. The key position can be any number from 1 through 4096. The entire key, defined by the key position and key length, must be within the record.

If a value is not specified, and the file you are restoring is an indexed file, the key position of the saved indexed file is assumed. If a key position is specified, the key length must also be specified.

key length specifies the length of the key for the restored file. The key length must be specified if the restored file is to be changed into an indexed file from a sequential or direct file (that is, when I is specified). Also, if you are restoring an indexed file, you can specify another field for the key. The key length can be any number from 1 through 120. The entire key, defined by the key position and key length, must be within the record.

If a value is not specified, and the file you are restoring is an indexed file, the key length of the saved indexed file is assumed. If a key length is specified, the key position must also be specified.

DUPKEY specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file will be the attribute of the output file.

NODUPKEY specifies that duplicate keys are not to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file will be the attribute of the output file.

REWIND specifies that the tape or tape cartridge should be rewound to the beginning after the RESTORE procedure has run. REWIND is assumed if this parameter is not specified.

LEAVE specifies that the tape or tape cartridge should be left where it is after the RESTORE procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the RESTORE procedure has run. If UNLOAD and TC are specified, the tape will be rewound after processing.

starting file name specifies the diskette, tape, or tape cartridge file within a set of files where the system should begin restoring files. This file and all files located after it are restored. If no parameter is specified, the system begins restoring files with the first file of the set.

starting file date specifies the creation date of the starting diskette, tape, or tape cartridge file specified by the starting file name. The date must be in the same format as the session date; you can use the STATUS SESSION control command to determine the date format. If a starting file date is specified, a starting file name must also be specified.

Example 1

This example shows how to restore all the files that were saved on diskette using the SAVE ALL procedure. The diskettes are in magazine slots M1 and M2, and the diskette drive is to automatically advance to M2 after processing the diskettes in M1.

```
RESTORE ALL,,,M1
```

Example 2

This example shows how to restore a file named PAYROLL to disk from diskette. The file is located on a diskette in slot S1.

```
RESTORE PAYROLL
```

Example 3

This example shows how to restore part of a set of files that was saved on diskette using the SAVE ALL procedure. All of the files on diskette after and including the file named FILEA are to be restored.

```
RESTORE ALL,,,,,,FILEA
```

Example 4

This example shows how to restore a file named PAYROLL to disk from tape. The tape is mounted on tape drive 1.

```
RESTORE PAYROLL,,,,,,T1
```

Example 5

This example shows how to restore a file named PAYROLL from tape cartridge to disk.

```
RESTORE PAYROLL,,,,,,TC
```

Example 6

The example that follows is based on a diskette file named FILE1 that contains the following five records. The file is an indexed file, the file's record length is 60, and the index key is in positions 1 through 4.

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE1       10/24/82      INDEXED         4        1         60              I1   10/24/82   11.03.50

REC NO. LINE NO. 1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

   1       1    0001SAMPLE DATA         4512 - 18TH AVE NW  NEW YORK, NY

   2       1    0002MORE SAMPLE DATA    100 - MAIN ST N     CHICAGO, IL

   3       1    0003CUSTOMER 1          3214 - 5TH ST NE    NEW YORK, NY

   4       1    0004CUSTOMER 2          2014 - 4TH ST NE    NEW YORK, NY

   5       1    0005CUSTOMER 3          3018 - 5TH ST NE    CHICAGO, IL

   TOTAL RECORDS PRINTED - 5
```

This example shows how to restore the file named FILE1. The key is to be changed from positions 1 through 4 to positions 5 through 24, and only those records that contain the phrase 'NEW' anywhere in the record are to be copied.

```
RESTORE FILE1,,,,,,S1,NOAUTO,INCLUDE,,EQ,'NEW',,I,5,20
```

FILE1 now contains the following records:

```
File name   File date   Organization  Key length Key loc  Record length    Unit    Date       Time       Page - 1
FILE1       10/24/82      INDEXED         20       5         60             F1   10/24/82   11.04.19

               1... ...10.... ...20.... ...30.... ...40.... ...50.... ...60.... ...70.... ...80.... ...90.... ..100

 KEY - CUSTOMER 1
LINE NO -   1    0003CUSTOMER 1          3214 - 5TH ST NE    NEW YORK, NY

 KEY - CUSTOMER 2
LINE NO -   1    0004CUSTOMER 2          2014 - 4TH ST NE    NEW YORK, NY

 KEY - SAMPLE DATA
LINE NO -   1    0001SAMPLE DATA         4512 - 18TH AVE NW  NEW YORK, NY

    TOTAL RECORDS PRINTED - 3
```

RETRIEVE Procedure

The RETRIEVE procedure restores a folder member that was copied onto disk, diskette, tape, or tape cartridge by the ARCHIVE procedure to a folder on disk. The archived member can be restored to the same folder or a different folder. If the member is restored to the same folder, the folder directory is updated to reflect the retrieved member. If the member is restored to a different folder or renamed when restored to the same folder, the directory is not updated.

If a folder member already exists with the same name as the member being retrieved or the new member name, a message is displayed that allows you to:

- Cancel the RETRIEVE procedure
- Delete the old member from the folder and continue with the RETRIEVE procedure

If the member to be retrieved is to be created with a subdirectory, the subdirectory must be specified on the RETRIEVE procedure.

The RETRIEVE procedure runs the $TMSERV utility program.

To retrieve a folder member from diskette:

```
RETRIEVE folder name,file name,I1,[date archived],[new member name],

        [S1  ],[AUTO  ],,[subdirectory]
        [S2  ]
        [S3  ]
        [M1  ]
        [M2  ]
        [M1.nn]
        [M2.nn]
```

S9020207-1

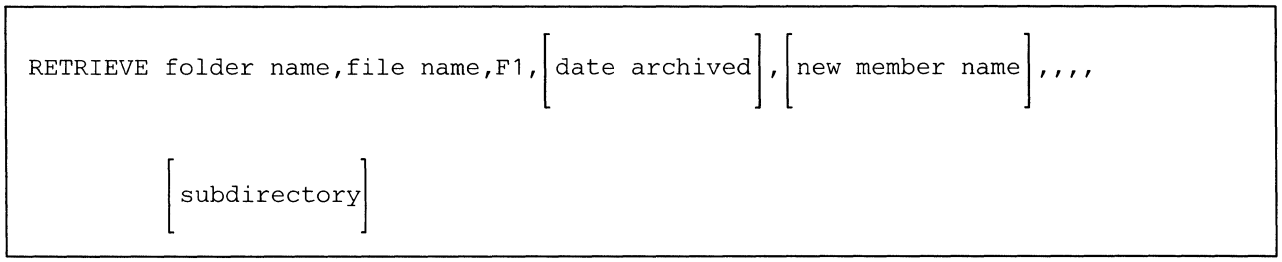
To retrieve a folder member from tape or tape cartridge:

```
RETRIEVE folder name,file name,[T1],[date archived],[new member name],,
                               [T2]
                               [TC]

        [AUTO  ],[REWIND],[subdirectory]
        [NOAUTO]
                 [LEAVE ]
                 [UNLOAD]
```

S9020208-2

RETRIEVE

To retrieve a folder member from disk:



S9020564-1

folder name specifies the name of the folder that is to contain the retrieved member.

file name specifies the name of the file from which the member is to be retrieved. If the file is located on diskette, tape, or tape cartridge, the file name may be a maximum of 12 characters long. If the file is located on disk, the file name may be a maximum of 8 characters long.

Note: If you are retrieving members from tape, duplicate file names with the same date may exist on the tape because there may be duplicate member names within subdirectories in the same folder. You will have to use OCL statements and sequence numbers to retrieve a member that exists in a tape file that has the same name and date as another tape file.

I1 specifies that the file containing the archived member is on diskette. If no parameter is specified, I1 is assumed.

T1, T2, or TC specifies that the file containing the archived member is to be retrieved from tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

F1 specifies that the file containing the archived member is on disk.

date archived specifies the date that the member was archived. The date was stored in the folder directory when the member was archived; you should specify this parameter to make sure that the correct member is retrieved. If no parameter is specified, the system retrieves the first file with the name specified in the second parameter.

new member name specifies the name of the member after it has been retrieved, if it is to be different from the current name of the member. A valid member name is from 1 to 12 characters. It is made up of a 1 to 8 character document name and may be optionally followed by a period and a 1 to 3 character extension.

S1, S2, or S3 specifies the diskette slot containing the diskette to be used. If no parameter is specified and the third parameter is I1, S1 is assumed. This parameter cannot be specified if you are retrieving from a disk file.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. This parameter cannot be specified if you are retrieving from a disk file.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

This parameter cannot be specified if you are retrieving from a disk file.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

This parameter cannot be specified if you are retrieving from a disk file.

REWIND specifies that the tape should be rewound after the member is retrieved. This parameter is valid only if T1, T2, or TC is specified. If no parameter is specified, REWIND is assumed. This parameter cannot be specified if you are retrieving from a disk or diskette file.

LEAVE specifies that the tape should be left where it is after the member is retrieved. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is valid only if T1, T2, or TC is specified. This parameter cannot be specified if you are retrieving from a disk or diskette file.

UNLOAD specifies that the tape should be rewound and unloaded after the member is retrieved. This parameter is valid only if T1 or T2 is specified. This parameter is not valid if I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing. This parameter cannot be specified if retrieving from a disk or diskette file.

subdirectory specifies the subdirectory path that will lead to the member that is to be retrieved and must be used for any future reference to this member. The subdirectory is a list of subdirectory names separated by forward slashes (/). Subdirectory names can be from 1 to 12 characters long, a 1 to 8 character name optionally followed by a period, and a 1 to 3 character extension. The total length of the specified subdirectory cannot exceed 62 characters.

If not specified, the member is placed in the base folder, and a subdirectory is not needed for future reference to the member.

Example 1

This example shows how to retrieve a file from a diskette named FILE1 to a folder named FLDR1.

```
RETRIEVE FLDR1,FILE1,I1
```

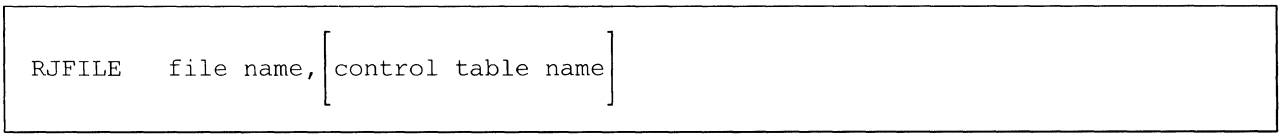
Example 2

This example shows how to retrieve an archived member from a tape cartridge file with a name of ARCHDMBRFILE to folder MYFOLDER with a subdirectory of SUB1.DIR/SUB2.DIR. The member is to be given a new name of MYMEMBER.NEW when placed in the folder.

```
RETRIEVE MYFOLDER,ARCHDMBRFILE,TC,,MYMEMBER.NEW,,,,,
SUB1.DIR/SUB2DIR
```

RJFILE Procedure

The RJFILE procedure is used to start the multiple session remote job entry (MSRJE) disk file utility. This utility allows you to convert punch or print data that was written to disk as 256-character unprocessed records by MSRJE to files that you define. You can also use RJFILE to print data that was written to disk as 256-character unprocessed records by MSRJE. For information on MSRJE and the RJFILE procedure, see the *MSRJE Guide*.



S9020209-0

file name specifies the file to be used by the utility.

control table name specifies the control table where the entry is located that is to be used by the utility. This control table is created by the RJTABLE procedure. If this parameter is not specified, the utility uses values supplied by the procedure. See the "RJTABLE Procedure."

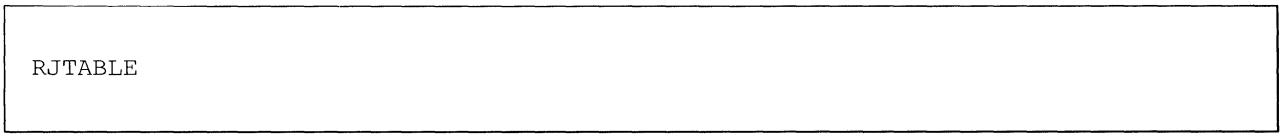
Example

This example shows the RJFILE procedure processing a file named DATAFILE using a control table named CONTROLS.

```
RJFILE DATAFILE,CONTROLS
```

RJTABLE Procedure

The RJTABLE procedure creates a control table for multiple session remote job entry (MSRJE). For information on MSRJE and the RJTABLE procedure, see the *MSRJE Guide*.



S9020210-0

The RJTABLE procedure has no parameters.

Example

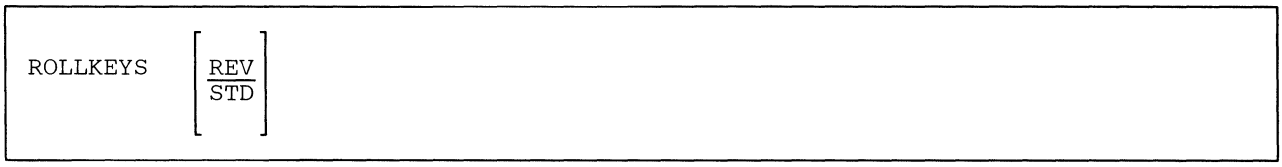
This example shows how to start the RJTABLE procedure.

```
RJTABLE
```

ROLLKEYS Procedure

The ROLLKEYS procedure allows you to set the direction that the roll keys move information on your display. For example, the current Roll Up key moves information from the bottom of the display toward the top. You can change the roll key direction so that when you press the roll Up key, the information from the top of the display moves toward the bottom. This procedure is only valid during the current session. If a user signs on to a pass-through session, the roll keys have their default values.

Note: This procedure is not recommended for users who are running 5250 emulation with the personal computer keyboard profile or using the IBM enhanced keyboard because the keys have already been remapped.



S9020555-0

REV specifies that the Roll Up key is to move information from the top of the display toward the bottom. The Roll Down key will move information from the bottom of the display toward the top. If this parameter is not specified, REV is assumed.

STD specifies that the Roll Up key will move information from the bottom of the display toward the top. The Roll Down key will move information from the top of the display toward the bottom.

Example

This example shows how to set the Roll Up key to move information from the top of the display toward the bottom. Because the default is *REV*, you do not have to specify it.

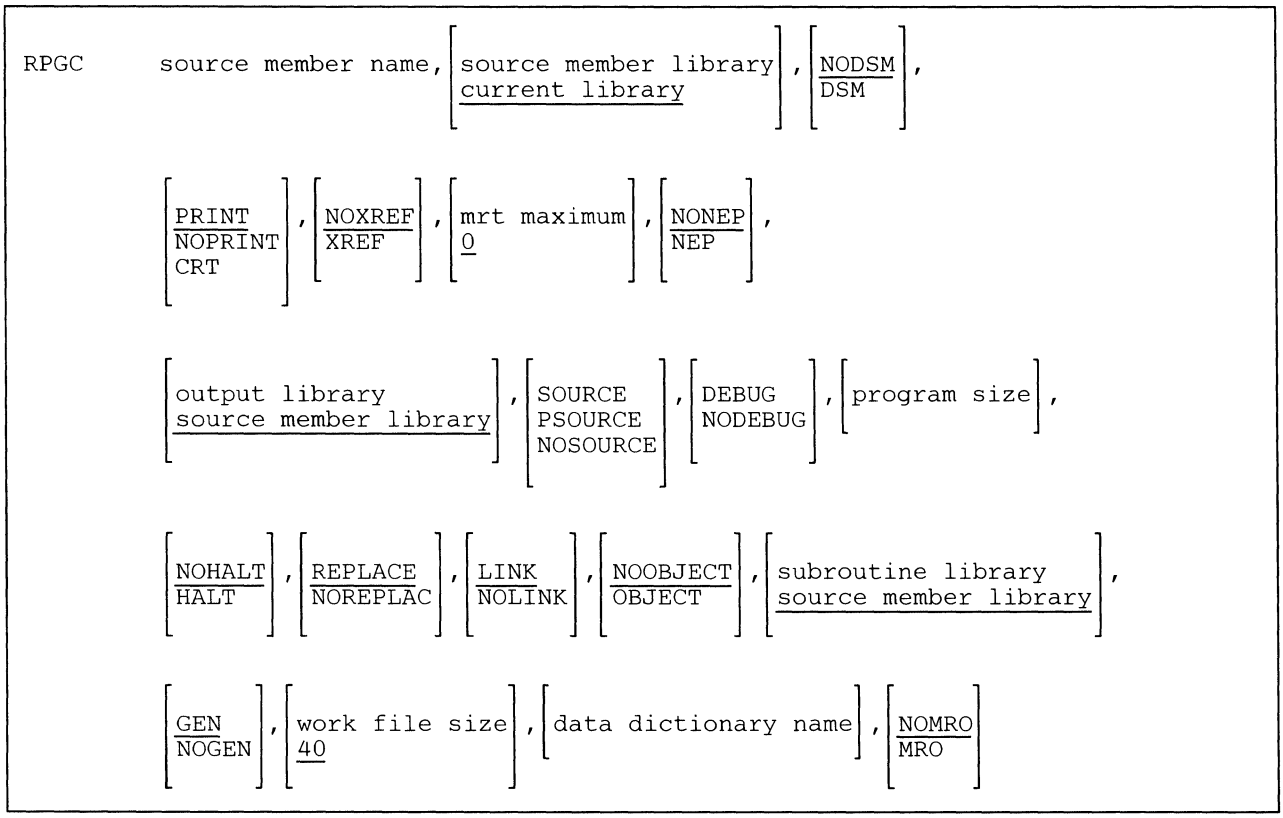
```
ROLLKEYS REV
```

RPG Procedure

The RPG procedure is supported only for compatibility with the IBM System/34. See the "RPGC Procedure" on page 4-407 for how to compile RPG programs.

RPGC Procedure

The RPGC procedure compiles an RPG II program. For information on RPG II, see the manual *Programming with RPG II*.



S9020211-1

source member name specifies the library source member that contains the RPG II program specifications.

source member library specifies the library that contains the source member to be compiled. If this parameter is not specified, the current library is assumed.

NODSM specifies that no diagnosed source member is to be created. If no parameter is specified, NODSM is assumed.

DSM specifies that a diagnosed source member is to be created. See the manual *Programming with RPG II* for more information about the diagnosed source member.

PRINT specifies that the compiler listings created by the RPGC procedure should be printed. If no parameter is specified, PRINT is assumed.

NOPRINT specifies that no compiler listings are to be printed or displayed.

CRT specifies that the compiler listings created by the RPGC procedure are to be displayed at the display station that is running the RPGC procedure.

NOXREF specifies that the RPGC procedure should not produce a cross-reference listing of the RPG II program. If no parameter is entered, NOXREF is assumed.

XREF specifies that a cross-reference listing is to be produced.

mrt maximum specifies the maximum number of display stations that can use the program at any one time. You can enter any number from 0 through 99. If the parameter is not entered, a value of 0 is assumed. A value of 0 indicates that the program is a single requester terminal (SRT) program; that is, each display station that runs the program is running its own copy of the program. A value of 1 or more indicates that the program is a multiple requester terminal (MRT) program.

For more information on MRT programs, see the *Concepts and Programmer's Guide*.

NONEP specifies that the program is not a never-ending program. If no parameter is entered, NONEP is assumed.

NEP specifies that the program is to be a never-ending program. See the NEP parameter of the "ATTR OCL Statement" on page 5-11 for more information on never-ending programs.

output library specifies the name of the library that is to contain the compiled program. If this parameter is not specified, the source member library is assumed.

SOURCE, PSOURCE, or NOSOURCE specifies the print option to be used instead of the entry in column 11 of the control specification in the source program. If no parameter is specified, the entry in column 11 of the control specification is used.

SOURCE specifies that the RPGC procedure is to create a full compiler listing.

PSOURCE specifies that the RPGC procedure is to create a partial compiler listing.

NOSOURCE specifies that the RPGC procedure is to create no compiler listing.

DEBUG or NODEBUG specifies the debug option to be used instead of the entry in column 15 of the control specification in the source program. If no parameter is specified, the value specified in column 15 of the control specification is used.

DEBUG specifies that the debug operation in the source program is to be used.

NODEBUG specifies that the debug operation in the source program is not to be used.

program size specifies the program size in K bytes (one K byte equals 1024 bytes) to be used instead of the program size entered in columns 12 through 14 of the control specification in the source program. The size can be an even number from 2 through 64. If no size is specified, the size specified in columns 12 through 14 of the control specification is used.

NOHALT specifies that the compiler should not stop and display an error message if a warning or terminating error is found in the program. If no parameter is specified, NOHALT is assumed.

HALT specifies that an error message should be displayed if a warning or terminating error is found in the program.

REPLACE specifies that if a load member or subroutine member is being created, and that if a load member or subroutine member with the same name as your program already exists in the output library, the newly compiled program is to replace the existing load or subroutine member. No message is displayed indicating the replace. If no parameter is specified, REPLACE is assumed.

NOREPLAC specifies that if a load member or subroutine member is being created, and that if a load member or subroutine member with the same name as your program already exists in the output library, a message is to be displayed and you can either replace the member or cancel the procedure.

LINK specifies that a load member is to be created. This parameter causes the overlay linkage editor to be run, which creates a program that can be run. If no parameter is entered, LINK is assumed.

NOLINK specifies that no load member is to be created.

NOOBJECT specifies that a subroutine member is not to be created. If no parameter is specified, NOOBJECT is assumed.

OBJECT specifies that a subroutine member is to be created. This member must be link-edited using the OLINK procedure or the overlay linkage editor before it can be run.

subroutine library specifies the library that contains one or more assembler subroutines to be combined with the program being compiled. If no parameter is specified, the source member library is assumed.

GEN specifies that if the RPG program being compiled contains a CONSOLE file, the display formats for that CONSOLE file are to be generated as part of the compile process. If no parameter is specified, GEN is assumed.

NOGEN specifies that no display formats for a CONSOLE file are to be generated.

work file size specifies the size (in blocks) for the RPGC work files. If no size is specified, 40 blocks is assumed.

data dictionary name specifies the data dictionary that contains the communications file definition to be used with the program being compiled.

MRO or NOMRO specifies whether the compiler is to use memory resident overlays. If no parameter is specified, NOMRO is assumed.

MRO specifies that the compiler is to use memory resident overlays.

NOMRO specifies that the compiler is not to use memory resident overlays.

Example

This example shows how to compile an RPG II program named PAYROLL. The program is contained in the current library; the compiled and link-edited load member is to be placed in the current library. A source listing and a cross-reference are to be generated, and the DEBUG operation is to be used.

```
RPGC PAYROLL,,,,XREF,,,,SOURCE,DEBUG
```

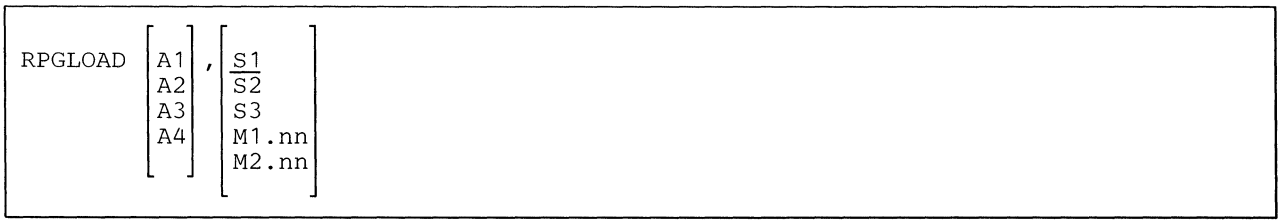
RPGLOAD Procedure

The RPGLOAD procedure creates a library named #RPGLIB and copies the RPG support from diskette into that library. RPGLOAD copies additional support into the system library (#LIBRARY). The RPGLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the RPGSAVE procedure. See the "RPGSAVE Procedure" on page 4-413 for information about how to save the RPG support on diskette.

The RPGLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the RPGLOAD procedure to restore support that has been saved by RPGSAVE.

If RPG is not currently on the system, you must use the TOLIBR procedure to copy the diskette file RPG into #LIBRARY before running RPGLOAD.

If #LIBRARY was backed up with RPG on the system and then replaced before RPGLOAD is run, you do not have to copy the diskette file RPG using the TOLIBR procedure.



S9020212-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #RPGLIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #RPGLIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #RPGLIB on disk and copy the RPG support from diskette.

```
RPGLOAD
```

RPGONL Procedure

The RPGONL procedure allows you to develop an RPG program. You can create a new program, or make changes to an existing program, and then compile and correct errors in the program by using the diagnosed source member function of the RPG compiler.

The RPGONL procedure shows displays that allow you to control how the procedure runs. You can end the RPGONL procedure from any of these displays.

See the manual *Programming with RPG II* for more information about this procedure and about RPG.

```
RPGONL
```

S9020213-0

The RPGONL procedure has no parameters.

Example

To start the RPGONL procedure, enter:

```
RPGONL
```

RPGP Procedure

The RPGP procedure displays a menu that allows you to select the RPG II task you want to perform. You can enter, change, or compile an RPG II program. You can also request a cross-reference listing, or create or change display formats used with RPG II programs.

See the manual *Programming with RPG II* for more information about this procedure and about RPG.

```
RPGP
```

S9020214-0

The RPGP procedure has no parameters.

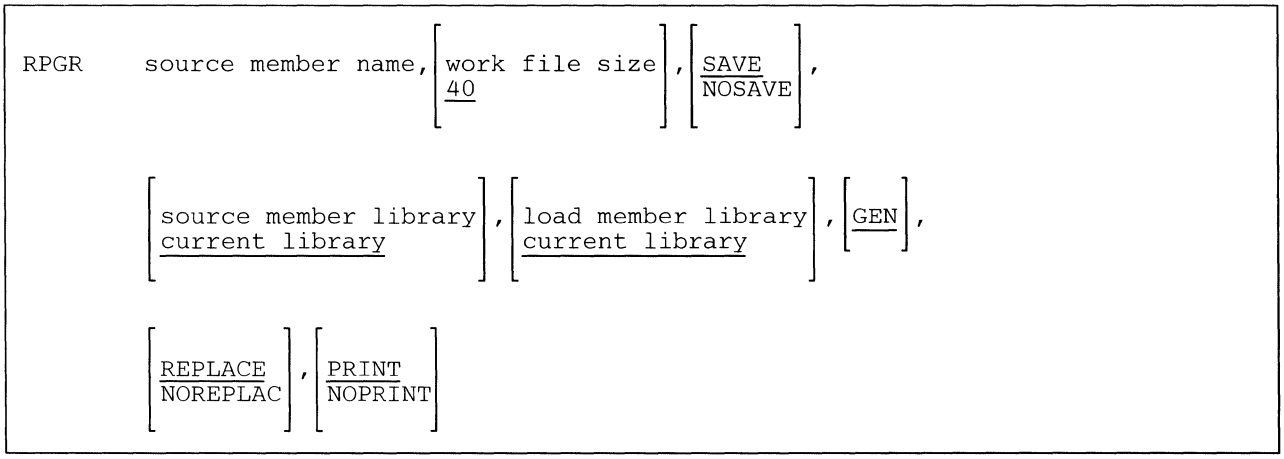
Example

To display a menu of RPG procedures, enter:

```
RPGP
```

RPGR Procedure

The RPGR procedure generates display format source and load members for a CONSOLE file in a RPG II source program. For information on RPG II, see the manual *Programming with RPG II*.



S9020215-0

source member name specifies the library source member that contains the RPG II program specifications.

work file size specifies the size of the $SOURCE work file in blocks, and can be any number from 1 through 999. If no parameter is entered, 40 blocks is assumed.

SAVE indicates that the source statements for the display format member are to be saved. If no parameter is entered, SAVE is assumed.

NOSAVE indicates that the source statements for the display format member are not to be saved. Only the display format load member will be created.

source member library specifies the library that contains the RPG source member. If no library name is specified, the current library is assumed.

load member library specifies the library that is to contain the display format load member. If no library name is specified, the current library is assumed.

GEN specifies that the display formats for the CONSOLE file are to be created. If no parameter is specified, GEN is assumed.

REPLACE specifies that an existing display format load member with the same name as **member name** is to be replaced. If no parameter is specified, REPLACE is assumed.

NOREPLAC specifies that if a load member already exists with the same name, a message is to be displayed and you can either continue or cancel the procedure.

PRINT specifies that a listing of the created display formats is to be printed. If no parameter is specified, PRINT is assumed.

NOPRINT specifies that the listing is not to be printed.

Example

This example shows how to generate the CONSOLE file display formats for an RPG II program named PAYROLL.

```
RPGR PAYROLL
```

RPGSAVE Procedure

The RPGSAVE procedure copies the RPG support to diskette. The RPG support from the libraries #RPGLIB and #LIBRARY is copied. You should use the "RPGLOAD Procedure" on page 4-410 to load the RPG support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPRPG and be located in diskette slot S1.

```
RPGSAVE
```

S9020216-0

The RPGSAVE procedure has no parameters.

Example

Copy the RPG support to diskette.

```
RPGSAVE
```

RPGSDA Procedure

The RPGSDA procedure starts the screen design aid (SDA) procedure. See the manual *Creating Displays* for more information about how to use SDA and for information about display formats.

See the manual *Programming with RPG II* for more information about this procedure and about RPG.

```
RPGSDA
```

S9020217-0

The RPGSDA procedure has no parameters.

Example

To start the RPGSDA procedure, enter:

```
RPGSDA
```

RPGSEU Procedure

The RPGSEU procedure allows you to create or change an RPG II program or procedure using the source entry utility (SEU). For information on RPG II, see the manual *Programming with RPG II*. For more information on SEU, see the manual *SEU Guide*.

```
RPGSEU   member name,[R],[seu format member],[statement length],
                      [A]  [#SE@XTRA       ]
                      [P]

         [library name  ]
         [current library]
```

S9020218-0

member name specifies the library member to be created or changed.

R specifies an RPG II source member. If no parameter is specified, R is assumed.

A specifies an RPG II source member containing auto report specifications.

P specifies a procedure member.

seu format member specifies the name of the load member that contains SEU formats. If no parameter is specified, #SE@XTRA is assumed.

statement length specifies the length for each source or procedure statement. This can be any number from 40 to 120. If the member exists, the statement length of the member is assumed. If the member is being created, the values you can specify and the values that are assumed by SEU if no statement length is specified are as follows:

Member Type	Allowed Statement Length	Assumed Statement Length
R	80 to 96	96
A	80 to 96	96
P	40 to 120	120

library name specifies the library that contains or will contain the member being changed or created. If no library name is specified, the current library is assumed.

Example

This example shows how to use the RPGSEU procedure to change an RPG II source member named PAYROLL. The current library contains the source member.

```
RPGSEU PAYROLL
```

RPGX Procedure

The RPGX procedure creates a cross-reference listing for an RPG II program (without compiling the program). For information on RPG II, see the manual *Programming with RPG II*.

```
RPGX    source member name,[work file size ],[source member library   ]
                           [40             ] [current library         ]
```

S9020219-0

If you enter the RPGX procedure with no parameters, a display allows you to enter the parameters.

source member name specifies the source member that contains the RPG II program specifications.

work file size specifies the size of the $SOURCE work file in blocks, and can be any number from 1 through 999. If no parameter is entered, 40 blocks is assumed.

source member library specifies the name of the library that contains the source member to be cross-referenced. If this parameter is not specified, the current library is assumed.

Example

This example shows how to create a cross-reference listing for an RPG II program named PAYROLL.

```
RPGX PAYROLL
```

SAVE Procedure

The SAVE procedure allows you to do the following:

- Save a single disk file on diskette, tape, or tape cartridge. The SAVE procedure can also be used to select records to be saved (only when saving a single file to diskette with no ADD operations).
- Add another disk file to a disk file already saved on diskette.
- Save a specified file group on diskette, tape, or tape cartridge.
- Save all user disk files on diskette, tape, or tape cartridge (including file groups).
- Save all user disk files on diskette, tape, or tape cartridge (except file groups).
- Save files on diskette in a compressed format by replacing repetitive characters with control characters.
- Save multiple file sets on tape or tape cartridge.

The SAVE procedure saves only user disk data files. The SAVE procedure cannot be used to save a library or folder

To save a library on diskette, tape, or tape cartridge, see the "SAVELIBR Procedure" on page 4-431.

To save a folder on diskette, tape, or tape cartridge, see the "SAVEFLDR Procedure" on page 4-428.

To save the network resource directory on diskette, tape, or tape cartridge, see the "SAVENRD Procedure" on page 4-434.

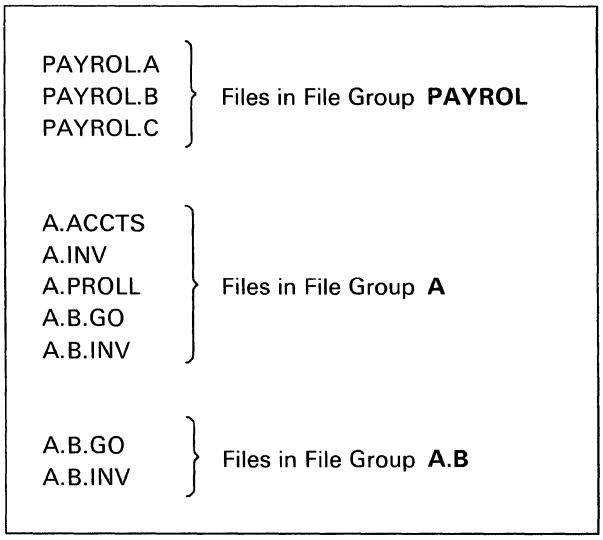
To save the system history file on diskette, tape, or tape cartridge, first run the HISTORY procedure to copy the history file to a disk file, and then run the SAVE procedure to save that disk file on diskette, tape, or tape cartridge. See the "HISTORY Procedure" on page 4-209 for more information.

The SAVE procedure can be used to save remote files individually to local diskettes. See the *Distributed Data Management Guide*.

When you use the SAVE procedure to save an indexed file that has no alternative indexes, the system saves the data and a description of the index, not the index itself.

When you save an alternative index, the system saves only a description of the index. Saving the physical file does not save the alternative indexes. However, you can save a physical file and all indexes for that file by using the file group-naming convention and specifying that group name for the file name parameter on the SAVE procedure.

File groups are defined by file names that contain a period. The characters preceding the period identify the file group, and the characters following the period identify the file within the group. As for all file names, the maximum number of characters is eight including the period. Files with names that do not contain a period are not part of a file group. Examples of names of files within a file group are:



S9020222-0

SAVE

The SAVE procedure can be used to save a file that is being used by another job on the system, if that job's FILE statement specifies DISP-SHRRR or DISP-SHRRM. See the "FILE OCL Statement (for Disk Files)" on page 5-32 for more information about file sharing.

The SAVE procedure can be run from any display station. If the SAVE procedure is used to copy all disk files (ALL is specified as both the first and the fifth parameters), the procedure can be run only if no jobs are adding to or modifying existing resident disk files (other jobs can be reading or creating new files). If any files are being added to or modified, a message is displayed and you can either try the SAVE procedure again or cancel the SAVE procedure.

All files in existence before entering the SAVE procedure can be saved. However, any files created during the save operation may or may not be saved depending on whether they are finished being created when their time to be saved occurs.

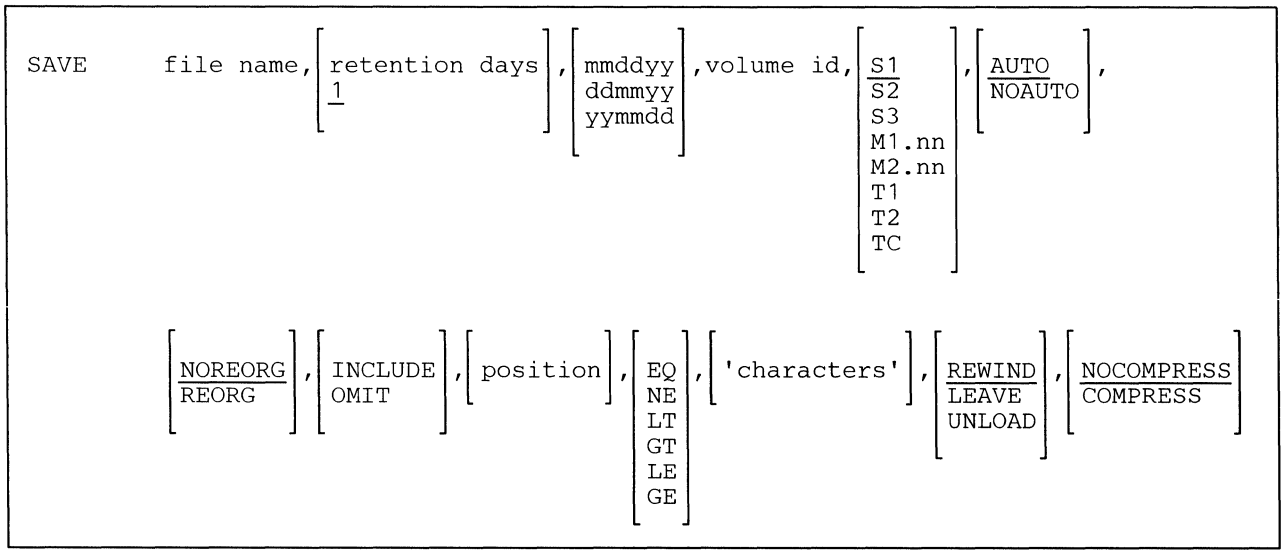
When you copy to diskette, tape, or tape cartridge all members of a specified file group, all disk files including file groups, or all disk files except file groups, you create a set of files on the diskette, tape, or tape cartridge. When you create this set of files on diskette, the receiving diskettes must be completely empty; they must not contain unexpired files. When you create this set of files on tape or tape cartridge, the first tape or tape cartridge may contain previously saved unexpired files or sets of files; therefore, the user should be cautious of duplicate file or set names.

| *Note: Expired tape files cannot be written over using the SAVE procedure; the tape must be initialized first.*

The SAVE procedure runs the $COPY utility program. $COPY will not process a library, a folder, or system files.

You can copy to following tape reels or cartridges if the first file of each reel or cartridge has expired.

For saving one disk file on diskette, tape, or tape cartridge:



S9020220-1

For adding a disk file to a diskette file:

```
SAVE     file name,ADD,[ mmddyy ],volume id,[ S1 ],[ AUTO ]
                       [ ddmmyy ]            [ S2 ]  [ NOAUTO ]
                       [ yymmdd ]            [ S3 ]
                                             [ M1.nn ]
                                             [ M2.nn ]
```

S9020221-0

For saving only disk files from a file group:

```
SAVE     [ ALL ],[ retention days ],[ set name ],volume id,file group,[ S1 ],
         [     ]  [ 1             ]  [ #SAVE    ]                     [ S2 ]
                                                                      [ S3 ]
                                                                      [ M1.nn ]
                                                                      [ M2.nn ]
                                                                      [ T1 ]
                                                                      [ T2 ]
                                                                      [ TC ]

         [ AUTO ],[ REWIND ],[ NOCOMPRESS ]
         [      ]  [       ]  [          ]
         [ NOAUTO ]  [ LEAVE ]  [ COMPRESS ]
                     [ UNLOAD ]
```

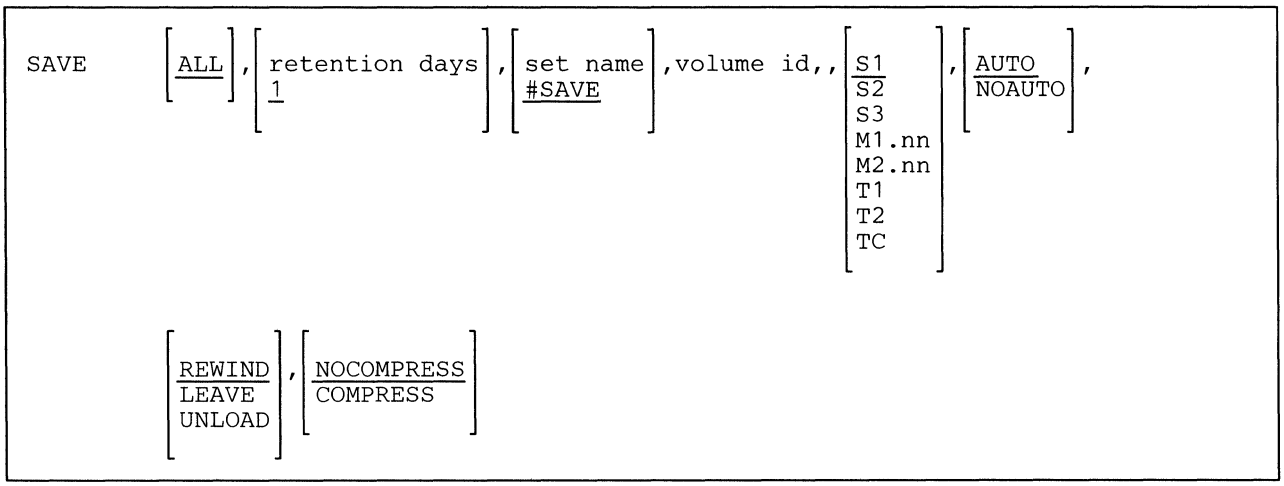
S9020223-1

For saving all disk files including files that belong to a file group:

```
SAVE     [ ALL ],[ retention days ],[ set name ],volume id,ALL,[ S1 ],[ AUTO ],
         [     ]  [ 1             ]  [ #SAVE    ]             [ S2 ]  [      ]
                                                              [ S3 ]  [ NOAUTO ]
                                                              [ M1.nn ]
                                                              [ M2.nn ]
                                                              [ T1 ]
                                                              [ T2 ]
                                                              [ TC ]

         [ REWIND ],[ NOCOMPRESS ]
         [        ]  [          ]
         [ LEAVE  ]  [ COMPRESS ]
         [ UNLOAD ]
```

S9020224-1

For saving all disk files except files that belong to a file group:



S9020225-1

file name specifies one disk file to be saved. The diskette, tape, or tape cartridge file name will be the same as the disk file name.

Note: When adding a disk file to a diskette file, both file names must be the same.

ALL specifies the following:

- If ALL is specified as both the first and the fifth parameters, all disk files are copied, whether or not file groups exist.
- If ALL is specified as the first parameter and a file group is specified as the fifth parameter, all members of that file group are saved.
- If ALL is specified as the first parameter and nothing is specified as the fifth parameter, all files that are not members of file groups are saved.

If no first parameter is specified, ALL is assumed.

If the first parameter is ALL or is not specified, the diskette, tape, or tape cartridge to which files are being copied cannot contain any active files.

retention days specifies how long the file is to be retained in days, and can be any number from 0 through 999. If a retention period is not specified, 1 day is assumed. If a retention period of 999 days is specified, the diskette, tape, or tape cartridge file is a permanent file. For more information on diskette, tape, or tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-43 or "FILE OCL Statement (for Tape Files)" on page 5-48.

ADD specifies that a single disk file is to be added to a file previously saved on diskette. The diskette file being added to must be the last file on the diskette. You can use the CATALOG procedure (to list the files on a diskette by location) to determine whether the file you want to add to is the last file on the diskette.

Note that if ADD is specified, parameters 7 through 13 cannot be specified. Also, ADD can be specified only for diskette, not for tape or tape cartridge.

set name specifies the name associated with the entire set of saved files. If no name is specified, the set name #SAVE is used.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the disk file. The date must be in the same format as the session date; use the STATUS SESSION control command to determine the date format. If the creation date is not specified and more than one file exists with the specified name, the most recent file is saved.

volume id specifies the volume ID of the diskette, tape reel, or tape cartridge. From 1 through 6 alphameric characters can be specified.

file group specifies the name of the file group to be saved. The period (.) that indicates a file group name must not be specified. For example, to save files belonging to the file group that includes PAYROL.A, PAYROL.B, and PAYROL.C, you would enter PAYROL for this parameter.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

T1, T2, or TC specifies the tape drive containing the first tape to be processed. T1 indicates the first tape drive, T2 indicates the second tape drive, and TC indicates the tape cartridge.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.
- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.
- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.
- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

Each time the SAVE procedure is run, the first diskette or tape to receive output must be in the specified location. If the diskette or tape in that location is not capable of receiving the output, the system will not search the following slots or tape drive for a valid diskette. Instead, an error message will be displayed, and the operator will have to either insert, into the specified slot or tape drive, a diskette or tape that is capable of receiving the output or cancel the procedure.

NOREORG specifies that the file is not to be reorganized. That is, for sequential and indexed files, any records that were deleted are saved along with the records that contain data. For direct files, deleted records are always saved. Also, for indexed files, the records are to be saved in the order in which they occur in the file.

REORG specifies that the file is to be reorganized. That is, for sequential and indexed files, any records that were deleted are not saved. Only the records that contain data are saved. For direct files, deleted records are always saved. Also, for indexed files, the records are to be saved sequentially by key. If T1, T2, or TC is specified, REORG is invalid.

INCLUDE or OMIT specifies whether specific records in the file are to be included in or omitted from the save. The INCLUDE and OMIT parameters work with the position, EQ, NE, LT, GT, LE, GE, and 'characters' parameters. If only the 'characters' or position is specified, INCLUDE and EQ are assumed. INCLUDE and OMIT are not valid if the save is to tape or tape cartridge.

position specifies, for each record, the first character to be compared with the starting position of the comparison characters. The position can be any number from 1 through 4096. If a position is not specified, then every position in the record is compared with the comparison characters. This parameter is not valid if the save is to tape or tape cartridge.

EQ specifies that if the characters in the record indicated by position are the same as the comparison characters, the record is to be included in or omitted from the save. EQ is not valid if the save is to tape or tape cartridge.

NE specifies that if the characters in the record indicated by position are not the same as the comparison characters, the record is to be included in or omitted from the save. NE is not valid if the save is to tape or tape cartridge.

LT specifies that if the characters in the record indicated by position are less than the comparison characters, the record is to be included in or omitted from the save. LT is not valid if the save is to tape or tape cartridge.

LE specifies that if the characters in the record indicated by position are less than or the same as the comparison characters, the record is to be included in or omitted from the save. LE is not valid if the save is to tape or tape cartridge.

GT specifies that if the characters in the record indicated by position are greater than the comparison characters, the record is to be included in or omitted from the save. GT is not valid if the save is to tape or tape cartridge.

GE specifies that if the characters in the record indicated by position are greater than or the same as the comparison characters, the record is to be included in or omitted from the save. GE is not valid if the save is to tape or tape cartridge.

'characters' specifies the comparison characters. Up to 30 characters can be specified, and they should be enclosed by apostrophes ('). You can specify any characters except apostrophes in the character string. This parameter is not valid if the save is to tape or tape cartridge.

REWIND specifies that the tape or tape cartridge should be rewound after the SAVE procedure has run. REWIND is assumed if this parameter is not specified.

LEAVE specifies that the tape or tape cartridge should not be rewound after the SAVE procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the SAVE procedure has run. If UNLOAD and TC are specified, the tape will be rewound after processing.

NOCOMPRESS specifies that data is not compressed when a disk file is saved on diskette. The diskette file appears in the same format as the disk file from which it was copied.

COMPRESS specifies that data is compressed when a disk file is saved on diskette. Repetitive characters in the file are replaced by control characters. When the file is restored, the data is put back in its original format. Compressing a file will usually save space on diskette, unless the file has few or no repetitive characters.

If COMPRESS is specified, the following restrictions apply:

- COMPRESS is not allowed if T1, T2, or TC is specified.
- The file must be saved on a diskette 2D diskette initialized to FORMAT2 (1024 bytes per sector).
- Records cannot be selected or reorganized.
- A disk file being added to a diskette file cannot be compressed, and a disk file cannot be added to a diskette file that has been compressed.
- A file that has been compressed cannot be restored on a System/32 or System/34.
- A file that has been compressed cannot be displayed using the LISTDATA or LISTFILE procedure.

Example 1

Save all disk files on diskette for a period of seven days. The diskettes have volume IDs of VOL001, and are located in magazines M1 and M2, starting in position M1.01.

```
SAVE ALL,7,,VOL001,ALL,M1,AUTO
```

Example 2

Save a file named FILE1 and add this file to an existing diskette file named FILE1. The volume ID of the diskette is VOL001.

```
SAVE FILE1,ADD,,VOL001
```

Example 3

Save all files belonging to file group PAYROL. The name to be associated with the set of saved files is PAYROL. The volume ID of the diskette is VOL002, and the file will be saved for at least 1 month (33 days). The files are compressed.

```
SAVE ALL,33,PAYROL,VOL002,PAYROL,,,,COMPRESS
```

Example 4

Save specific records from a single file named FILE3 on a diskette. The only records to be saved contain the word SAVE in positions 10 through 13 of the record.

```
SAVE FILE3,,,VOL001,,,,INCLUDE,10,EQ,'SAVE'
```

Example 5

Save all files that are not group files on tape reel VOL001, which is mounted on tape drive T1. When the reel on T1 is filled, the reel on T2 is to be used (if the tape reel name is VOL001 and no files exist on it).

```
SAVE ALL,,,VOL001,,T1,AUTO,REWIND
```

Example 6

Save a file named FILE1 on tape reel VOL001 mounted on tape drive T2. If additional tape reels are required to hold the file, use only tape drive 2. Also, do not rewind the tape after saving the file.

```
SAVE FILE1,,,VOL001,T2,NOAUTO,,,,,,LEAVE
```

Example 7

Save a disk file named FILE1 to tape cartridge volume TEST1. The tape file will be a permanent file.

```
SAVE FILE1,999,,TEST1,TC
```

SAVEEXTN Procedure (for the Ideographic Version of the SSP)

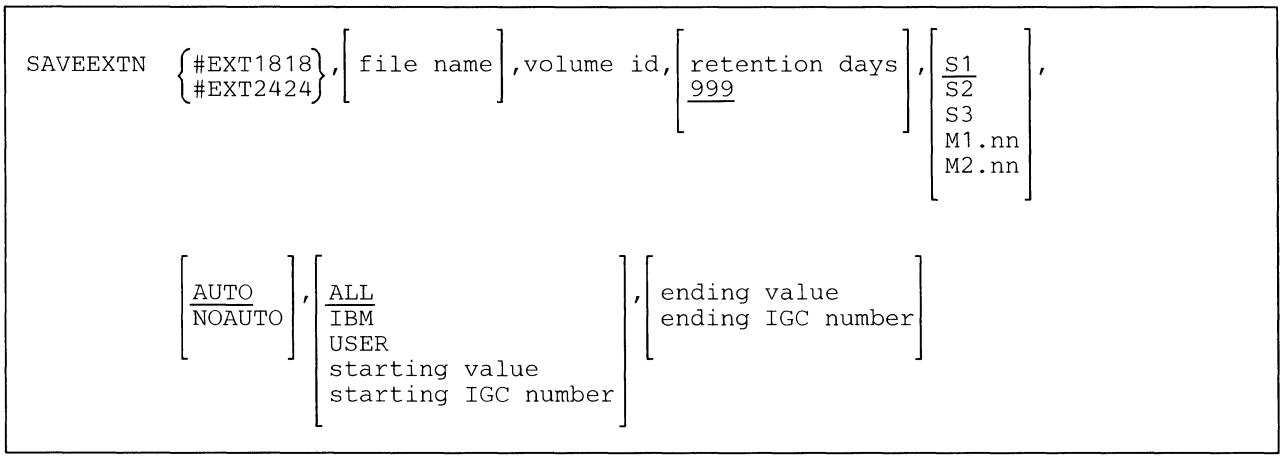
The SAVEEXTN procedure saves from disk all or part of the extended character file to diskette. The extended character file is divided into two parts, the IBM-supplied extended characters and the user-defined extended characters. You can save either or both of these parts. You can also save specified portions of the user-defined extended characters.

If password security is active, an operator with system operator authority can run the SAVEEXTN procedure from any display station.

You can use the character generator utility to enter ideographic characters in the extended character file. For information on how to use the character generator utility, see the *Character Generator Utility Guide.*

The SAVEEXTN procedure runs the $XSAVE utility program.

Note: The SAVEEXTN procedure supports only Japanese extended character files. Extended character files from Korea, Taiwan, or People's Republic of China are not supported.



S9020226-0

#EXT1818 or #EXT2424 specifies the name of the extended character file on disk to be saved. The name must be either #EXT1818 or #EXT2424.

file name specifies the name of the diskette file to contain the extended character file. If no name is specified, the name specified in the first parameter is assumed.

volume id specifies the volume ID of the diskette. From 1 through 6 alphameric characters can be specified.

retention days specifies how long (in days) the diskette file containing the extended character file is to be retained, and can be any number from 1 through 999. If a retention period is not specified, 999 days are assumed. If a retention period of 999 days is specified, the diskette file is a permanent file. For more information on diskette file retention, see *FILE OCL Statement (for Diskette Files)* on page 5-37.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

ALL specifies that the entire extended character file is to be saved to diskette. If this parameter is not specified, ALL is assumed.

IBM specifies that only the IBM-supplied extended characters are to be saved to diskette.

USER specifies that only the user-defined extended characters are to be saved to diskette.

starting value specifies with which user-defined extended character to begin saving to diskette. The code point value must be for a user-defined extended character and not an IBM-supplied extended character. The value must be specified in the following format: Xdddd, where dddd is the code point value. For more information about code point values for extended characters, refer to the *Character Generator Utility Guide*.

starting IGC number specifies with which user-defined extended character to begin saving to diskette. The IGC number is the 5-digit decimal number that was assigned when the character was created using CGU. For more information on the IGC number, refer to the *Character Generator Utility Guide*.

ending value specifies with which user-defined extended character to stop saving to diskette. The code point value must be for a user-defined extended character and not an IBM-supplied extended character. The value must be specified in the following format: Xdddd, where dddd is the code point value. For more information about code point values for extended characters, refer to the *Character Generator Utility Guide.* If this parameter is not specified, and IBM was specified for the previous parameter, only the IBM-supplied extended characters will be saved. If this parameter is not specified, and any value other than IBM was specified for the previous parameter, the last user-defined extended character in the extended character file is assumed.

ending IGC number specifies with which user-defined extended character to stop restoring to disk. The IGC number is the 5-digit decimal number that was assigned when the character was created using CGU. For more information on the IGC number, refer to the *Character Generator Utility Guide.* If this parameter is not specified, and IBM was specified for the previous parameter, only the IBM-supplied extended characters will be saved. If this parameter is not specified, and any value other than IBM was specified for the previous parameter, the last user-defined extended character in the extended character file is assumed.

Example

This example shows how to save the entire extended character file, including user-defined extended characters, from disk to diskette. The name of the disk file is #EXT2424. The same name is used for the diskette file. The volume ID of the diskettes are IGC. The diskettes are located in magazine M1. Make the diskette file a permanent file.

```
SAVEEXTN #EXT2424,,IGC,,M1
```

SAVEFLDR Procedure

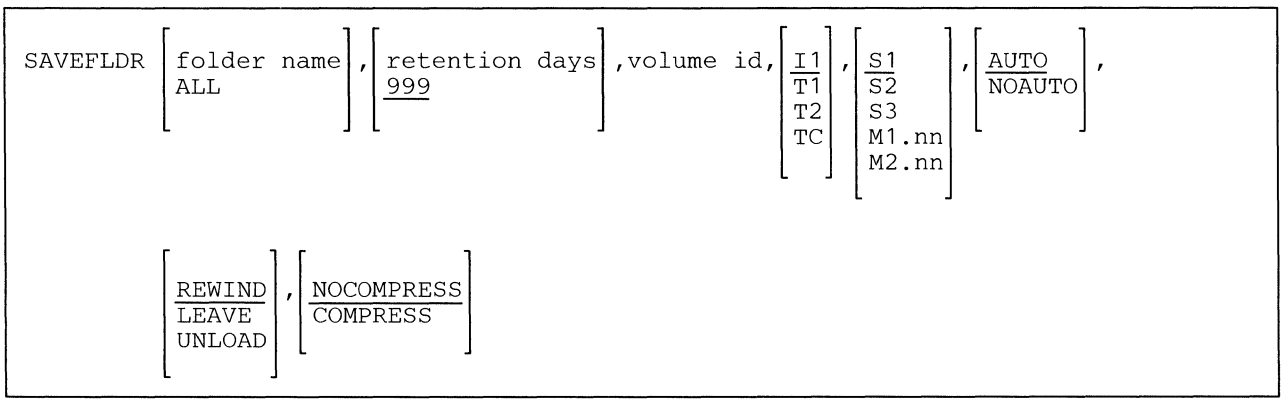
The SAVEFLDR procedure saves the entire contents of a folder or all folders on the system on diskette, tape, or tape cartridge. The SAVEFLDR procedure also saves the entire contents of a single folder on disk.

To determine the number of diskettes required to contain SSP folders, see the manual *Changing Your System Configuration*, SC21-9052.

To restore a folder copied on disk, diskette, tape, or tape cartridge by the SAVEFLDR procedure, see the "RESTFLDR Procedure" on page 4-383.

The SAVEFLDR procedure runs the $TMSERV utility program.

To save a single folder or all folders on diskette, tape, or tape cartridge.



S9020486-2

To save a single folder on disk:

```
SAVEFLDR   folder name,,,F1,,,,,file name
```

S9020565-0

folder name specifies the folder to save on disk, diskette, tape, or tape cartridge. The diskette, tape, or tape cartridge file containing the folder is given the same name as the folder. The name of the disk file to contain the folder must be different from the name of the folder. For an explanation of the convention that the system uses to name the folder and its extents, see "SAVEFLDR Files on Diskette" in the *Concepts and Programmer's Guide*.

ALL specifies that all folders that currently exist on the system are to be saved. ALL is valid only if you are saving on diskette, tape, or tape cartridge.

retention days specifies how long (in days) the diskette, tape, or tape cartridge file containing the folder is to be retained and can be any number from 0 through 999. If a retention period is not specified, 999 days are assumed. If a retention period of 999 days is used, the diskette, tape, or tape cartridge file is a permanent file. For more information on diskette, tape, or tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

volume id specifies the volume ID of the diskette, tape, or tape cartridge. From 1 through 6 alphameric characters can be specified. If no volume ID is specified, you are prompted to enter a volume ID.

I1 specifies that the folder is to be saved to diskette.

F1 specifies that the folder is to be saved to disk.

T1, T2, or TC specifies that the folder is to be saved to tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed. This parameter is valid only if parameter 4 is I1.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive is used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.
- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.
- If T1 or T2 is specified, only the specified tape drive will be used for all tape volumes.
- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the SAVEFLDR procedure has run. This parameter is valid only if T1, T2, or TC is specified. If no parameter is specified, REWIND is assumed.

LEAVE specifies that the tape should be left where it is after the SAVEFLDR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is valid only if T1, T2, or TC is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the SAVEFLDR procedure has run. This parameter is not valid if I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

NOCOMPRESS specifies that the contents of a folder are not compressed when the folder is saved on diskette. The folder on diskette appears in the same format as the folder on disk from which it was copied.

COMPRESS specifies that the contents of a folder are compressed when the folder is saved on diskette. Repetitive characters in the folder are replaced by control characters. When the folder is restored, its contents are put back in their original format. Compressing a folder usually saves space on diskette, unless the folder has few or no repetitive characters.

If COMPRESS is specified, the following restrictions apply:

- COMPRESS is not allowed if F1, T1, T2, or TC is specified.
- The folder must be saved on a 2D diskette initialized to FORMAT2 (1024 bytes per sector).

file name specifies the name of the disk file that the folder is to be saved on. A file name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, @, or $). The remaining characters can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. Do not use ALL as a file name. The name specified must be different from the folder name.

Example

Save a folder named MYFLDR as a permanent file on diskette with a volume ID of IBMIRD. The folder is compressed when saved.

```
SAVEFLDR MYFLDR,999,IBMIRD,,,,,COMPRESS
```

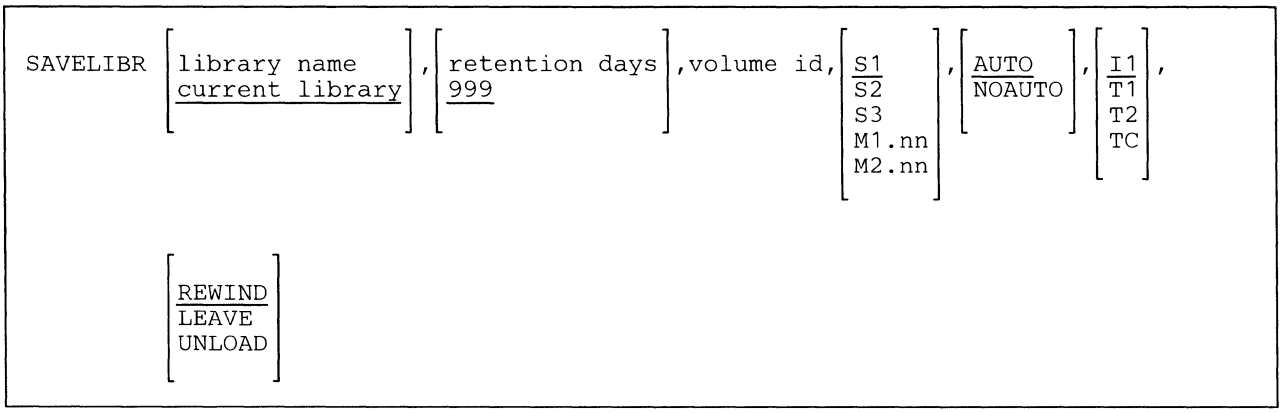
SAVELIBR Procedure

The SAVELIBR procedure saves the entire contents of a library on diskette, tape, or tape cartridge. All the members of the library are saved, together with the size of the library and the size of the library's directory. If the library contains IBM-supplied members, those members are also saved.

To restore a library copied onto diskette, tape, or tape cartridge by the SAVELIBR procedure, see the "RESTLIBR Procedure" on page 4-386.

You can copy one or more library members to diskette, tape, or tape cartridge by using the FROMLIBR procedure, but only the members are copied; the library and directory sizes are not saved. See the "FROMLIBR Procedure" on page 4-193 for more information about copying only library members.

The SAVELIBR procedure runs the $MAINT utility program.



S9020227-1

library name specifies the library to save on diskette, tape, or tape cartridge. The diskette, tape or tape cartridge file containing the library is given the same name as the library. If the library name is not specified, the current library is assumed. To save the system library to diskette, tape, or tape cartridge, enter #LIBRARY. (If you save #LIBRARY, no files can precede it on the diskette, tape, or tape cartridge.)

retention days specifies how long (in days) the diskette, tape, or tape cartridge file containing the library is to be retained and can be any number from 0 through 999. If a retention period is not specified, 999 days are assumed. If a retention period of 999 days is used, the diskette, tape, or tape cartridge file is a permanent file. For more information on diskette, tape, or tape cartridge file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

volume id specifies the volume ID of the diskette, tape reel, or tape cartridge. From 1 through 6 alphameric characters can be specified. If no volume ID is specified, you are prompted to enter a volume ID.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed. This parameter is not allowed if T1, T2, or TC is specified.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is not allowed if T1, T2, or TC is specified.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only the specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

I1 specifies that the library is to be saved to diskette.

T1, T2 or TC specifies that the library is to be saved to tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

REWIND specifies that the tape should be rewound after the SAVELIBR procedure has run. REWIND is assumed if this parameter is not specified. This parameter is not valid if I1 is specified.

LEAVE specifies that the tape should be left where it is after the SAVELIBR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is not valid if I1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the SAVELIBR procedure has run. This parameter is not valid if I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

Example 1

Save a library named MYLIB permanently on a diskette with a volume ID of VOL001. The diskette to contain the library is located in diskette slot S1.

```
SAVELIBR MYLIB,999,VOL001
```

Example 2

Save the system library on diskette with a volume ID of SYSTEM. The diskettes to contain the library are located in magazine slot M1, starting in position M1.01.

```
SAVELIBR #LIBRARY,,SYSTEM,M1
```

Example 3

Save the library named PAYLIB on tape with a volume ID of PAYROL. The tapes to contain the library are located on tape drives 1 and 2, starting with tape drive 2. After the library has been saved, the tape should be unloaded.

```
SAVELIBR PAYLIB,,PAYROL,,AUTO,T2,UNLOAD
```

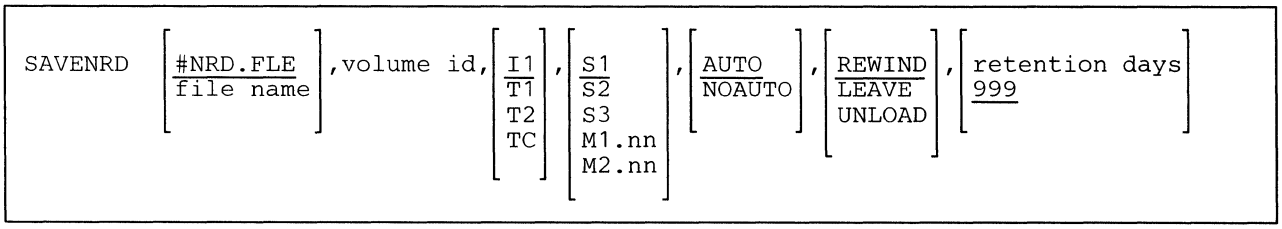
SAVENRD Procedure

The SAVENRD procedure saves the network resource directory on diskette, tape, or tape cartridge. If the directory is saved on tape or tape cartridge, the tape or tape cartridge must have a standard label. The directory can be saved as #NRD.FLE (the default directory file name) or with another file name.

The SAVENRD procedure can be run from any display station. The directory cannot be saved while it is being used by or edited by another user or job.

For more information about the network resource directory, see the *Distributed Data Management Guide.*

The SAVENRD procedure runs the $COPY utility program.



S9020228-1

#NRD.FLE specifies that the name of the directory (#NRD.FLE) is to be the same when saved as on disk. If no parameter is specified, #NRD.FLE is assumed.

file name specifies the name under which the directory is to be saved on diskette or tape.

volume id specifies the volume ID of the diskette or tape reel. From 1 through 6 alphameric characters can be specified.

I1 specifies that the directory is to be saved on diskette. If no parameter is specified, I1 is assumed.

T1, T2, or TC specifies that the directory is to be saved on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed. This parameter is not allowed if T1, T2, or TC is specified

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is not allowed if T1, T2, or TC is specified

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

Each time the SAVENRD procedure is run, the first diskette to receive output must be in the specified location. If the diskette in that location is not capable of receiving the output, the system will not search the following slots for a valid diskette. Instead, an error message will be displayed, and the operator will have to either insert a diskette that is capable of receiving the output into the specified slot or cancel the procedure.

REWIND specifies that the tape should be rewound after the SAVENRD procedure has run. This parameter is valid only if T1, T2, or TC is specified. If no parameter is specified, REWIND is assumed.

LEAVE specifies that the tape should be left where it is after the SAVENRD procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is valid only if T1, T2, or TC is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the SAVENRD procedure has run. This parameter is not valid I1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

retention days specifies how long the file is to be retained in days, and can be any number from 1 through 999. If a retention period is not specified, 999 days are assumed. If a retention period of 999 days is specified, the diskette or tape file is a permanent file. For more information on diskette or tape file retention, see "FILE OCL Statement (for Diskette Files)" on page 5-43 or "FILE OCL Statement (for Tape Files)" on page 5-48.

Example 1

This example saves the directory as #NRD.FLE on the diskette located in slot S2. The volume ID of the diskette is VOL001. The directory is saved as a permanent file.

```
SAVENRD #NRD.FLE,VOL001,,S2
```

Example 2

This example saves the directory as NRD.BKUP on tape reel BKUP04, which is mounted on tape drive T1. When the directory has been saved, the tape reel will be rewound and unloaded. The file will be retained for one year.

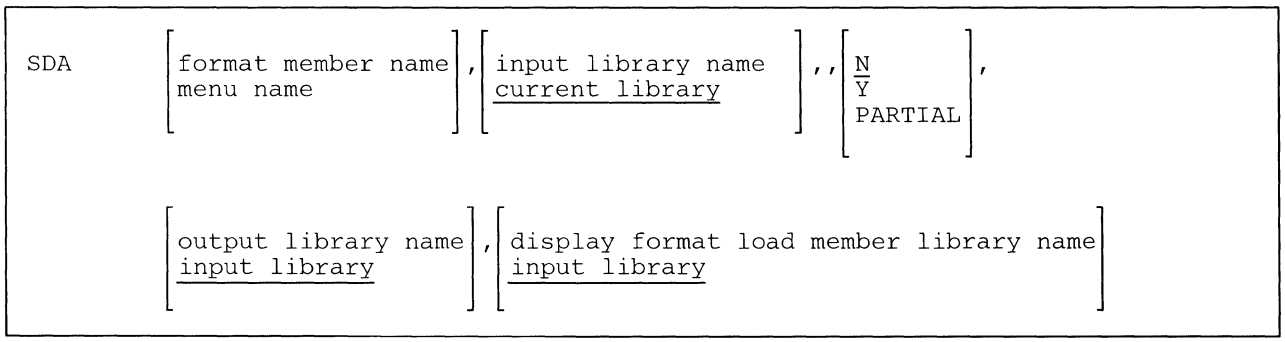
```
SAVENRD NRD.BKUP,BKUP04,T1,,,UNLOAD,365
```

SDA Procedure

The SDA procedure starts the screen design aid (SDA) program. You can use SDA to:

- Create or change menus or display formats
- Help create work station utility (WSU) and RPG programs
- Edit library members using source entry utility (SEU) or development support utility (DSU)
- View display formats
- Print display formats
- Generate display format load members using the $SFGR utility program

SDA is part of the Utilities Program Product. For more information on how to use SDA and about display formats and menus, see the manual *Creating Displays*.



S9020229-0

All the parameters for the SDA procedure are optional and are used to establish the default values that appear on following displays that are shown by SDA.

format member name or menu name specifies the display format source member or menu that you want to create or change. The format member name can be up to 8 characters long; the menu name can be up to 6 characters long.

input library specifies the name of the library that contains or will contain the display format or menu source members. This parameter also specifies the library to be searched for display format load members when you display formats contained in a load member. If this parameter is not specified, the current library is assumed.

Position 3 is for compatibility with the IBM System/34. This position is ignored.

N specifies that only error messages, and the lines containing the error, are to be listed when display formats are generated. If no parameter is specified, N is assumed.

Y specifies that the entire display format source member, together with any errors, is to be listed when display formats are generated.

PARTIAL specifies that a partial listing of the display format source member is to be listed when display formats are generated.

output library specifies the name of the library that is to contain the created or changed menu, or the source for the created or changed display format. If no library name is specified, the input library is assumed.

display format load member library specifies the name of the library that is to contain the created or changed display format load member. If no library name is specified, the input library is assumed.

Example

This example shows how to start SDA.

```
SDA
```

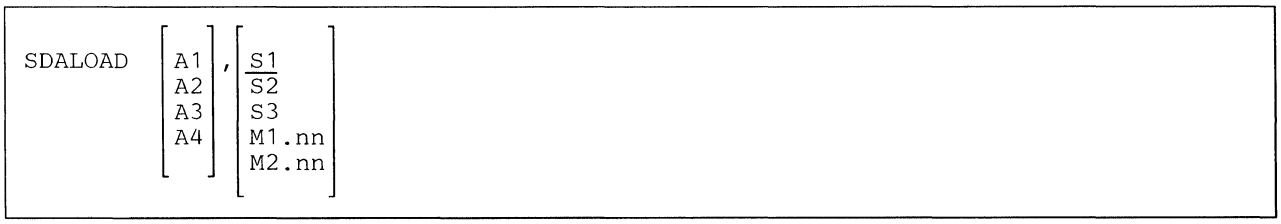
SDALOAD Procedure

The SDALOAD procedure creates a library named #SDALIB and copies the screen design aid (SDA) support from diskette into that library. SDALOAD copies additional support into the system library (#LIBRARY). The SDALOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the SDASAVE procedure. See the "SDASAVE Procedure" on page 4-440 for information about how to save the SDA support on diskette.

The SDALOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the SDALOAD procedure to restore support that has been saved by SDASAVE.

If SDA is not currently on the system, you must use the TOLIBR procedure to copy the diskette file SDA into #LIBRARY before running SDALOAD.

If #LIBRARY was backed up with SDA on the system and then replaced before SDALOAD is run, you do not have to copy the diskette file SDA using the TOLIBR procedure.



S9020230-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #SDALIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #SDALIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #SDALIB on disk and copy the SDA support from diskette.

```
SDALOAD
```

SDASAVE Procedure

The SDASAVE procedure copies the screen design aid (SDA) support to diskette. The SDA support from the libraries #SDALIB and #LIBRARY is copied. You should use the "SDALOAD Procedure" on page 4-439 to load the SDA support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPUTIL and be located in diskette slot S1.

```
SDASAVE
```

S9020231-0

The SDASAVE procedure has no parameters.

Example

Copy the SDA support to diskette.

```
SDASAVE
```

SECDEF Procedure

The SECDEF procedure allows you to do the following:

- Create or remove the user identification file
- Create or remove the resource security file
- Activate or deactivate password security
- Activate or deactivate badge security
- Activate or deactivate resource security
- Start or stop password date checking

For more information about these files and how to run the SECDEF procedure, see the *System Security Guide*.

If password security is active, only the master security officer can run the SECDEF procedure. If password security is not active, the SECDEF procedure can only be run from the system console.

The SECDEF procedure runs the $PRUID or $RRESC utility program, depending on the parameters entered.

To define password security:

```
SECDEF   [USERID],[CREATE
                   DELETE
                   ACTPW
                   DEACTPW
                   ACTBDG
                   DEACTBDG
                   STRTDATE
                   STOPDATE]
```

S9020232-1

To define resource security:

```
SECDEF   RESOURCE,[CREATE
                   DELETE
                   ACTRES
                   DEACTRES]
```

S9020233-0

USERID specifies that the user identification file is to be defined, or that password or badge security is to be activated or deactivated. If no parameter is specified, USERID is assumed.

If no second parameter is entered, a help menu is displayed that lists the options for SECDEF USERID. If a second parameter is entered, a prompt display is shown.

CREATE displays a prompt that allows you to create the user ID file.

DELETE displays a prompt that allows you to remove the user ID file.

ACTPW displays a prompt that allows you to activate password security.

DEACTPW displays a prompt that allows you to deactivate password security.

ACTBDG displays a prompt that allows you to activate badge security.

DEACTBDG displays a prompt that allows you to deactivate badge security.

STRTDATE displays a prompt that allows you to start the password date checking function.

STOPDATE displays a prompt that allows you to stop the password date checking function.

RESOURCE specifies that the resource security file is to be defined, or that resource security is to be activated or deactivated.

If no second parameter is entered, a help menu is displayed that lists the options for SECDEF RESOURCE. If a second parameter is entered, a prompt display is shown.

CREATE displays a prompt that allows you to create the resource security file.

DELETE displays a prompt that allows you to remove the resource security file.

ACTRES displays a prompt that allows you to activate resource security.

DEACTRES displays a prompt that allows you to deactivate resource security.

Example

This example shows how to start the SECDEF procedure to work with the user identification file.

```
SECDEF
```

You will then see a help menu, and you can select one of the options from that menu.

SECEDIT Procedure

The SECEDIT procedure allows you to add, remove, or update entries in the user identification file or in the resource security file.

For User Profiles in the User Identification File: If password security is active, only the master security officer or security officer can run the SECEDIT procedure with the USERID parameter specified to add, remove, or update user profiles in the user identification file.

If password security is active, any user can run the SECEDIT procedure with the USERID parameter specified to change his or her user profile. If password security is not active, the SECEDIT procedure with the USERID parameter specified can be run only from the system console.

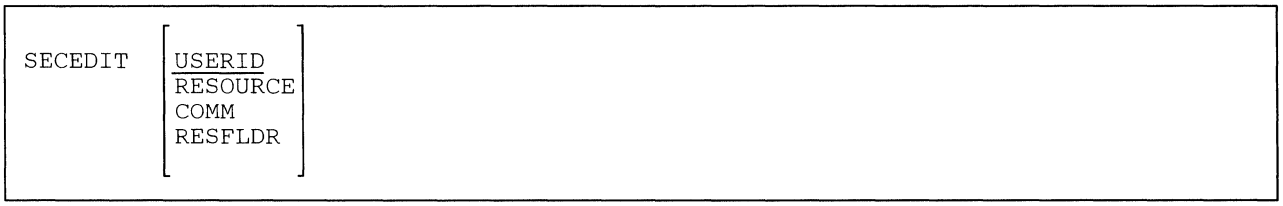
For Location Profiles in the User Identification File: If password security is active, only the master security officer can run the SECEDIT procedure with the COMM parameter specified to add, remove, or update location profiles in the user ID file. If password security is not active, the SECEDIT procedure cannot be run with the COMM parameter specified.

For the Resource Security File: If password security is active, only the master security officer or security officer can run the SECEDIT procedure to add, remove, or update all the information in the resource security file. If password security is not active, the SECEDIT procedure can be run only from the system console to add, remove, or update all entries in the resource security file. Owners of resources can update information for all their own resources. Any user can add information for an unsecured resource.

For more information about these files and how to use the SECEDIT procedure, see the *System Security Guide*.

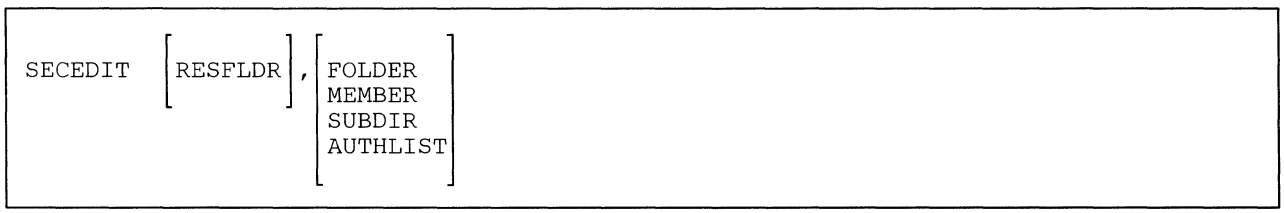
The SECEDIT procedure runs the $PRCED, $PRUED, $RREDT, or $RRTED utility program, depending on the parameters entered.

To add or change records in the user identification file or the resource security file:



S9020234-1

To work with authorization lists and security information for folders, members, and subdirectories:



S9020592-0

USERID specifies that you want to add, remove, or update user profiles in the user identification file. If no parameter is specified, USERID is assumed.

RESOURCE specifies that you want to add, remove, or update entries in the resource security file for files, libraries, and groups.

COMM specifies that you want to add, remove, or update location profiles in the user identification file.

RESFLDR specifies that you want to add, remove, or update authorization lists or security information for folders or subdirectories in the resource security file, or security information for folder members.

FOLDER specifies that you want to work with security information for folders.

MEMBER specifies that you want to work with security information for members.

SUBDIR specifies that you want to work with security information for subdirectories.

AUTHLIST specifies that you want to work with authorization lists.

Example

This example shows how to start the SECEDIT procedure to change an entry in the user identification file.

```
SECEDIT
```

SECLIST Procedure

The SECLIST procedure provides a listing of the user identification file and a listing of the resource security file.

For User Profiles in the User Identification File: If password security is active, only the master security officer or security officer can run the SECLIST procedure with the USERID parameter specified to list user profiles in the user ID file. If password security is not active, the SECLIST procedure with the USERID parameter specified can be run only from the system console.

For Location Profiles in the User Identification File: If password security is active, only the master security officer can run the SECLIST procedure with the COMM parameter specified to list location profiles in the user ID file. If password security is not active, the SECLIST procedure cannot be run with the COMM parameter specified.

For the Resource Security File: If password security is active, only the master security officer or security officer can run the SECLIST procedure to list all information in the resource security file. If password security is not active, the SECLIST procedure can be run only from the system console to list all information in the resource security file. Owners of resources and authorized users can list information for all their own resources at any time.

If you try to print with ideographic headings at a nonideographic printer, blanks are printed in place of the ideographic characters.

For information about security and for descriptions of the listings created by this procedure, see the *System Security Guide*.

The SECLIST procedure runs the $PRCLT, $PRLST, $RRLST, or $RRTLT utility program, depending on the parameters entered.

To list user profiles in the user identification file:

```
SECLIST  USERID,[PRINTER/CRT],[NOPW/PW],[USERID/CLASS]
```

S9020235-1

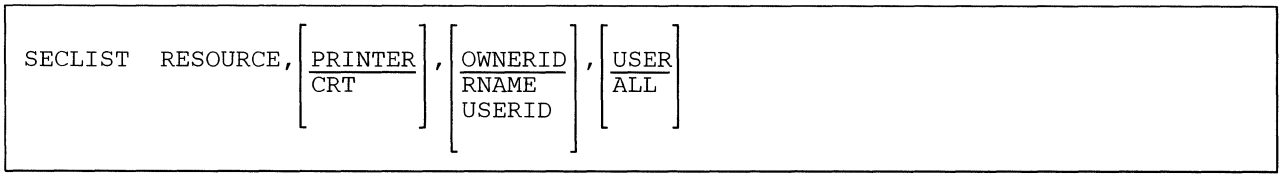
To list location profiles in the user identification file:

```
SECLIST  COMM,[PRINTER/CRT]
```

S9020237-0

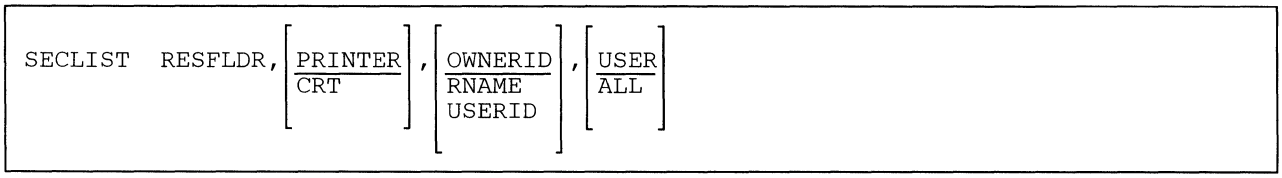
SECLIST

To list information about files, libraries, and groups in the resource security file:



S9020236-0

To list information about folders, authorization lists, and subdirectories in the resource security file:



S9020238-1

USERID specifies that user profiles in the user identification file are to be listed.

COMM specifies that location profiles in the user identification file are to be listed.

RESOURCE specifies that information about files, libraries, and groups in the resource security file is to be listed.

RESFLDR specifies that information about folders, subdirectories, and authorization lists in the resource security file is to be listed.

PRINTER specifies that the information is to be printed on your session printer. If no parameter is specified, PRINTER is assumed.

CRT specifies that the information is to be displayed.

NOPW specifies that no passwords are to be listed. If no parameter is specified, NOPW is assumed.

PW specifies that user passwords can be listed. When password security is active, a master security officer can list all the passwords in the file. A security officer can list only the passwords of operators with a security classification of system operator or lower. Passwords cannot be listed if password security is not active.

OWNERID specifies that resources are to be listed alphabetically by owner ID. If no parameter is specified, OWNERID is assumed.

RNAME specifies that resources are to be listed by resource name. If RNAME is specified, USER cannot be specified; ALL must be specified.

USERID specifies that output or resources are to be listed alphabetically by user ID. If SECLIST USERID is specified, output is listed alphabetically by user ID. If SECLIST RESOURCE is specified, resources are listed alphabetically by user ID. If no parameter is specified, USERID is assumed.

CLASS specifies that the output is to be listed in security classification order starting with master security officers. User IDs within each security classification are listed in alphabetical order.

USER specifies that only the resource security entries for the operator that runs the procedure are to be listed. If no parameter is specified, USER is assumed.

ALL specifies that all resource security file entries are to be listed. The operator must be a master security officer or a security officer to use this parameter when password security is active. If password security is not active, the operator must be running the procedure from the system console.

Example 1

This example shows how the master security officer or security officer would print all the resource security entries in alphabetic order by resource name.

```
SECLIST RESOURCE,,RNAME,ALL
```

Example 2

This example shows how the master security officer or security officer would print the entries in the user ID file with the passwords of the users.

```
SECLIST USERID,,PW
```

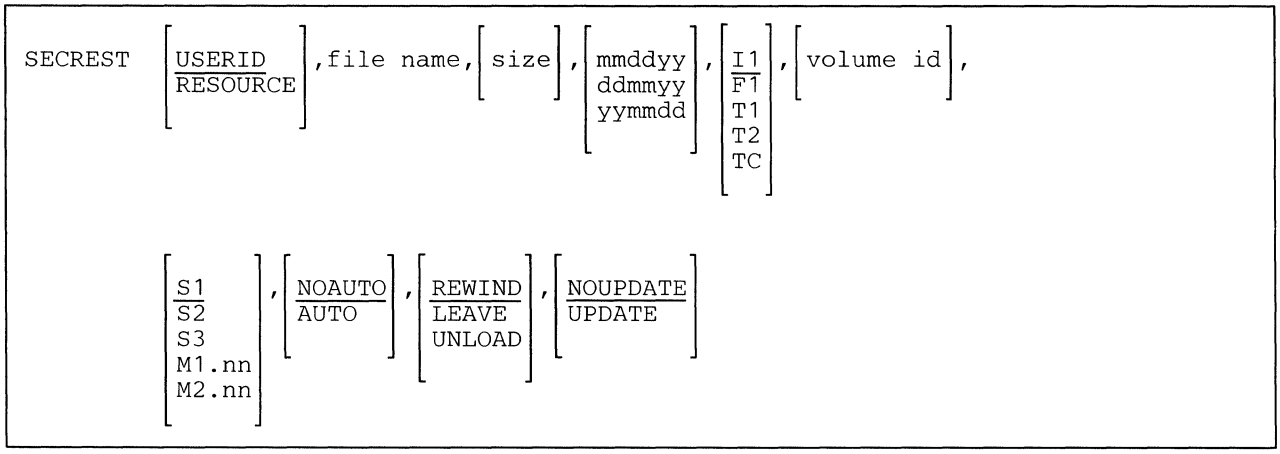
SECREST Procedure

The SECREST procedure restores the user identification file or the resource security file with the copy that was saved by the SECSAVE procedure.

If password security is active, only the master security officer can run the SECREST procedure. If password security is not active, the SECREST procedure can only be run from the system console.

For more information about these files and about security, see the *System Security Guide*.

The SECREST procedure runs the $PRURS or $RRSTR utility program, depending on the parameters entered.



S9020239-1

USERID specifies that you want to restore the user identification file. If no parameter is specified, USERID is assumed.

RESOURCE specifies that you want to restore the resource security file.

file name specifies the diskette, disk, tape, or tape cartridge file that contains the saved copy of the user identification file or resource security file. This file is to replace the current file.

size specifies the maximum number of records to allow in the user identification or resource security file. For the user identification file, this can be any number from 18 through 3998. For the resource security file, this can be any number from 72 through 63992. If no number is specified, the maximum number of records saved in the copy is used. If the number entered is too large for the system, or too small to contain all the copied records, an error message is displayed.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the diskette, disk, tape, or tape cartridge file that contains the saved copy. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format.

If no date is specified and more than one diskette, disk, tape, or tape cartridge file exists with the specified file name, the most recent file is used.

I1 specifies that the file is to be restored from diskette. If no parameter is specified, I1 is assumed.

F1 specifies that the file is to be restored from a disk file.

T1, T2, or TC specifies that the file is to be restored from tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

volume id specifies the volume ID of the diskette, tape reel, or tape cartridge that contains the copy. If I1, T1, T2, or TC is specified, and no volume ID is specified, the SSP copies the file from the diskette file in the specified location.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. This parameter is valid only if I1 is specified. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is valid only if I1 is specified.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.
- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.
- If T1 or T2 is specified, only the specified tape drive will be used for all tape volumes.
- If TC is specified, the NOAUTO/AUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, NOAUTO is assumed.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used.

- If TC is specified, the NOAUTO/AUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound to the beginning after the SECREST procedure has run. REWIND is assumed if this parameter is not specified.

LEAVE specifies that the tape should be left where it is after the SECREST procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the SECREST procedure has run. This parameter is not valid if I1, or F1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

NOUPDATE specifies that the date-password-last-changed field in those user profiles with a date of zeroes is to be set to the current system date. Other user profiles are not changed.

UPDATE specifies that the date-password-last-changed field in every user profile is to be set to the current system date.

Example 1

This example shows how to restore the user identification file from a copy on diskette. The volume ID of the diskette is VOL001, and the diskette is located in diskette slot S1. The name of the diskette file is USERFILE.

```
SECREST USERID,USERFILE,,,I1,VOL001,,,,UPDATE
```

Example 2

This example shows how to restore the user identification file from a copy on tape. The volume ID of the tape reel is VOL001, and the tape is mounted on tape drive 1. The tape is rewound and unloaded after the file has been restored.

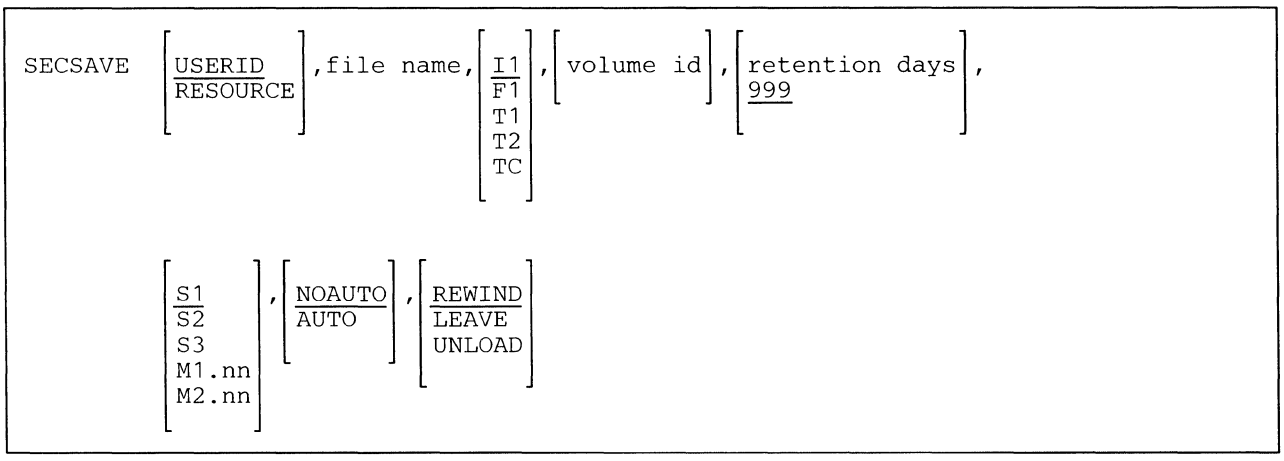
```
SECREST USERID,USERFILE,,,T1,VOL001,,,UNLOAD,NOUPDATE
```

SECSAVE Procedure

The SECSAVE procedure saves the user identification file or the resource security file on diskette, disk, tape, or tape cartridge. For more information about these files, see the *System Security Guide*.

If password security is active, only the master security officer can run the SECSAVE procedure. If password security is not active, the SECSAVE procedure can be run only from the system console.

The SECSAVE procedure runs the $PRUSV or $RRSAV utility program, depending on the parameters entered.



S9020240-1

USERID specifies that you want to save the user identification file. If no parameter is specified, USERID is assumed.

RESOURCE specifies that you want to save the resource security file.

file name specifies the name of the file to be created.

I1 specifies that the file is to be saved on a diskette. If no parameter is specified, I1 is assumed.

F1 specifies that the file is to be copied to another disk file.

T1, T2, or TC specifies that the file is to be saved on a tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

volume id specifies the volume ID of the diskette or tape that is to receive the saved copy. If I1, T1, T2, or TC is specified, a volume ID must be specified.

retention days specifies the number of days the diskette, tape, or tape cartridge is to be retained. Any decimal number from 1 through 999 can be entered. If no parameter is specified, 999 days are assumed. If 999 days is entered or assumed, the file is a permanent diskette file. See the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48 for more information about diskette file retention.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. This parameter is valid only if I1 is specified. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is valid only if I1 is specified.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.
- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.
- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.
- If TC is specified, the NOAUTO/AUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, NOAUTO is assumed.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.
- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.
- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used.
- If TC is specified, the NOAUTO/AUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the SECSAVE procedure has run. REWIND is assumed if this parameter is not specified.

LEAVE specifies that the tape should be left where it is after the SECSAVE procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the SECSAVE procedure has run. This parameter is not valid if I1 or F1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

Example 1

This example shows how to copy the user identification file to diskette. The volume ID of the diskette is VOL001, and the diskette is located in diskette slot S1. The name of the diskette file will be USERFILE.

```
SECSAVE ,USERFILE,,VOL001
```

Example 2

This example shows how to copy the resource security file to diskette. The volume ID of the diskette is VOL001, and the diskette is located in diskette slot S1. The name of the diskette file will be RESFILE.

```
SECSAVE RESOURCE,RESFILE,,VOL001
```

Example 3

This example shows how to copy the user identification file to tape. The volume ID of the tape reel is VOL001, and the tape is mounted on tape drive 1. The tape is positioned where it stops when the file has been copied.

```
SECSAVE ,USERFILE,T1,VOL001,,,,LEAVE
```

SET Procedure

The SET procedure establishes the following display station environment items:

- Number of lines printed per page
- Print belt image
- Session date format
- Session date
- Default region size
- Library assigned to the display station
- Printer for printed output
- Forms number
- Printer for Print key output
- Whether a border or header should be printed with Print key output

The items specified are placed in the **display station configuration record**, which defines display station characteristics. This causes your changes to remain in effect after you sign off. This information (except the session date) remains unchanged until another SET procedure is run from the display station, the system library is restored, or the system is configured again.

If the SET procedure is run during an inquiry request (the operator pressed the Attn key), changes are in effect only during the inquiry request and will not affect the interrupted job. Information displayed using the STATUS command will not reflect changes made during the inquiry request.

The SET procedure cannot be run from the job queue, or started by the EVOKE OCL statement.

The SET procedure runs the $SETCF utility program.

```
SET     [lines per page],[image member],[MDY|DMY|YMD],[mmddyy|ddmmyy|yymmdd],[region size],

        [library name|#LIBRARY|0],[printer id|SYSTEM],[forms number],

        [print key printer id],[BORDER|NOBORDER],[HEADER|NOHEADER]
```

S9020241-0

lines per page specifies the number of lines to be printed per page. The maximum number of lines that can be specified is 112; the minimum value is 1.

For information about how the value specified determines the actual number of lines printed per page, see the "FORMS OCL Statement" on page 5-55.

image member specifies the name of the source member that contains the print belt characters for the line printer. The SSP includes the following print belt members:

Member Name
BELT48
BELT48HN
BELT64B
BELT64C
BELT96
BELT188B

For information about the characters available in these print belts, see "3262 Print Belts" on page E-1. The print belt member name is also called the universal character set buffer (UCSB) in the *IBM 3262 Printer Models A1 and B1 Component Description and Operator's Guide*, GA33-1530.

MDY specifies that the session date format is month-day-year.

DMY specifies that the session date format is day-month-year.

YMD specifies that the session date format is year-month-day.

mmddyy, ddmmyy, or yymmdd specifies the session date. The date must be specified in the session date format.

region size specifies the region size to be used for programs run from this display station. (This value will be overridden if a different region size is specified for a job by a REGION OCL statement.) The size is specified in K-bytes (one K-byte contains 1024 bytes).

library name specifies the sign-on library to be assigned to the display station. The system library (#LIBRARY) can be assigned. If 0 is specified, no library is active.

The specified library does not become active until the **next** time an operator signs on at the display station.

printer id or SYSTEM specifies the default sign-on printer to be used for system list output created from the display station.

You can use the PRINT or SYSLIST procedure to immediately change the printer used for system list output. See the "PRINT Procedure" on page 4-350 or the "SYSLIST Procedure" on page 4-498 for information about running these procedures.

printer id specifies the 2-character ID of the printer to be used as the sign-on printer for the display station.

SYSTEM or SYS specifies that the system printer is to be used as the sign-on printer for the display station.

forms number specifies the forms number to be used for display station output. The forms number can also be changed by:

- A FORMS statement
- A PRINTER statement (for that print step only)
- The PRINT procedure

print key printer id specifies the 2-character work station ID of the printer to be used for Print key output.

BORDER specifies that a border indicating display positions is to be printed around the Print key output. See the "PRINTKEY Procedure" on page 4-354 for an example of the border that is printed.

NOBORDER specifies that no border is to be printed around the Print key output.

HEADER specifies that a header indicating from where the Print key was pressed is to be printed above the Print key output. See the "PRINTKEY Procedure" on page 4-354 for an example of the header that is printed.

NOHEADER specifies that no header is to be printed above the Print key output.

Example 1

Change the number of lines per page to 66.

```
SET 66
```

Example 2

Change the region size to be used for programs run from this display station.

```
SET ,,,,48
```

SETALERT Procedure

The SETALERT procedure changes the alert indicators for messages in a message load member. Both IBM-supplied messages and user-defined messages can be assigned these values. The setting of the alert indicator determines whether an alert will be sent when the message is issued. For more information about alerts, see the *Communications and Systems Management Guide*.

The input to the SETALERT procedure is an alert source member that contains, for each message that causes an alert to be built and sent: the alpha code, the message identification code (MIC), and the alert indicator. The source member can be created using the Source Entry Utility (SEU), the Development Support Utility (DSU), or the $MAINT utility program.

The changes made by the SETALERT procedure remain in effect until changed by another SETALERT procedure or until a new release is installed for IBM-supplied messages.

The alert source member contains three types of statements. See the "Alert Source Statements" on page 4-458 for information about the format of these statements.

- The alert control statement
- One or more alert specification statements
- One or more comment statements (optional)

The SETALERT procedure runs the $ARSP utility program.

```
SETALERT source member name [,library name      ]
                            [ current library   ]
```

S9020608-0

source member name specifies the source member that contains the alert control statements, specification statements, and comment statements.

library name specifies the library that contains the source member. If a library name is not specified, the current library is assumed.

Example

This example shows how to update the alert indicators for the messages specified in the source member named PROGALRT (which is stored in a library named MYLIB).

```
SETALERT PROGALRT,MYLIB
```

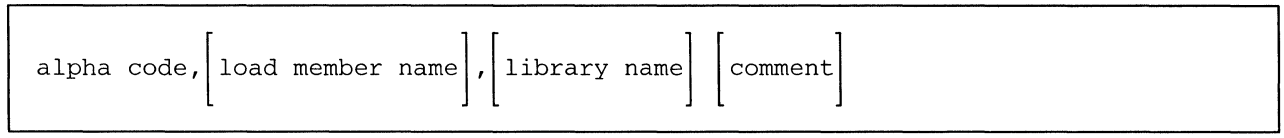
Alert Source Statements

The Alert Control Statement

The alert control statement specifies the alpha code for the alert specification statements that follow. The alpha codes are two, three, or four characters that are displayed before the MIC number. For example, if message SYS-1395 is displayed, SYS is the alpha code and 1395 is the MIC number.

The SETALERT procedure will process both nonideographic and ideographic data in the case where a message may exist in both nonideographic and ideographic form.

The syntax of the alert control statement is:



S9020609-0

alpha code specifies the alpha code of the messages that the user specifies for generating alerts. The alpha code must be one of the following:

ASM	DSU	IWS	SDA	USER
BAS	EMU	KBD	SEU	WSU
BGU	EP	NRD	SORT	
CBL	ESU	OFC	SRTX	
CGU	ET	QRY	SYS	
DFU	FORT	RJE	TTM	
DHCF	IDDU	RPG	TXT	

You should group the messages with the same alpha codes together in the source member. This allows you to update the source member more easily.

load member name specifies the message load member whose alert values are to be updated. If the alpha code is USER, the member name must be specified. If the alpha code is not USER, the member name parameter is ignored and the $ARSP utility determines the member name.

library name specifies the library that contains the message load member. If the alpha code is USER, the library name can either be specified or not specified; if it is not specified, the system library (#LIBRARY) is assumed. If the alpha code is not USER, the library name parameter is ignored and the $ARSP utility determines the library name.

comment specifies any information that may be helpful to identify the message or the response you want. This information is not used by the system. One or more blanks must be placed before the comment.

Alert Specification Statement

The alert specification statement specifies the MIC of the message to be updated and the alert setting.

If an alert specification statement is specified for a message that is not in the specified message member, an error message is displayed.

The syntax of the alert specification statement is:

```
mic  alert generation status indicator [comment]
```

S9020610-1

mic specifies the message identification code (MIC) of the message. The MIC must be a 4-digit number from 0000 to 9999; it must be placed in positions 1 through 4 of the alert specification statement. The MICs must be in ascending numerical order.

alert generation status indicator specifies whether this MIC should cause an alert to be built and sent. If you specify Y, the message member will be updated and an alert will be sent when the message is issued. If you specify N, the message member will be updated, and an alert will not be sent when the message is issued. The Y or N must be placed in position 6 of the source statement.

comment specifies any information that may be helpful to identify the message or the alert setting you want. This information is not used by the system. One or more blanks must be placed before the comment.

Alert Comment Statement

The alert comment statement specifies any information that may be helpful to identify the message or the response you want. This information is not used by the system.

The syntax of the comment statement is:

```
* comment
```

S9020611-0

The asterisk (*) must be the first character in the statement. Comment statements can be placed anywhere within the other statements.

Alert Programming Considerations

If an alert source member (created via the ALERT procedure) is used as the source member with the SETALERT procedure, the procedure will fail. In this case,

```
SYS-8999 Invalid source record format
```

will be displayed, and the procedure will terminate.

Example Alert Source Member

Assume an alert source member contains the following statements. Note how the statements are grouped by the alpha codes and the MICs are listed in numerical order.

```
CBL
3301 Y Send alert for message CBL-3301
3314 N Do not send alert for message CBL-3314
*Change alert setting for the following FORTRAN messages
FORT
4360 Y
4361 Y
```

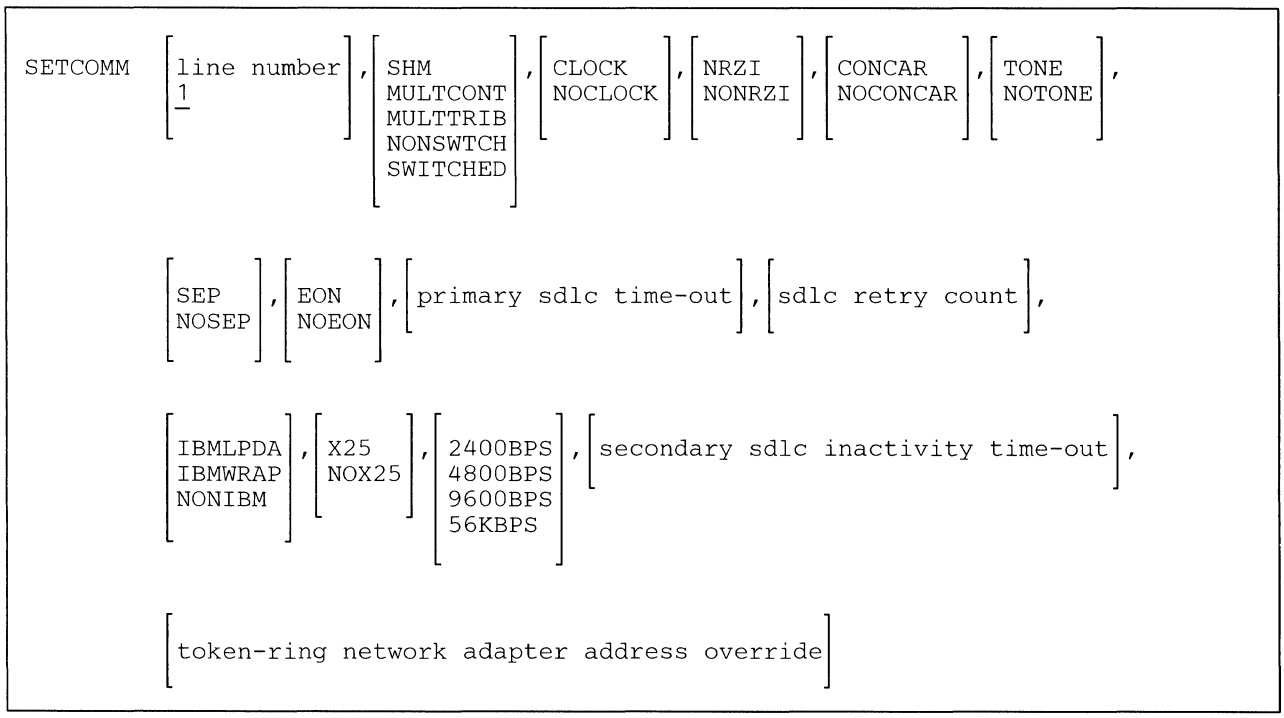
SETCOMM Procedure

The SETCOMM procedure allows you to set certain communications items in the communications configuration record. You can use the STATUS COMCNFIG control command to display the current settings. For more information on the SETCOMM procedure, see the manual *Using System/36 Communications*.

The changes you make using the SETCOMM procedure remain in effect until changed by a subsequent SETCOMM procedure.

After you have run the SETCOMM procedure, you **must** perform an initial program load (IPL) on the system. This causes the changes you made to become effective. See the manual *Operating Your System* for your system unit for information about how to perform an IPL. Any values set for a display station by the ALTERCOM procedure will be lost and the new values specified in the SETCOMM procedure will become effective. After the IPL, use the STATUS COMCNFIG command to ensure that the changes are correct.

The SETCOMM procedure runs the $SETCP utility program.



S9020242-5

Although each parameter is optional, at least one parameter, other than line number, must be specified. If a parameter is not specified, the preset values remain unchanged.

line number specifies the number of the communications line for which the settings are to be changed. If no parameter is specified, 1 is assumed.

NONSWTCH specifies that the line is a point-to-point nonswitched line.

SWITCHED specifies that the line is a point-to-point switched line.

MULTTRIB specifies that the System/36 is a multipoint tributary station.

MULTCONT specifies that the System/36 is a multipoint control station. This parameter cannot be used with BSC.

SHM specifies that the line uses X.21 short hold mode, which helps reduce line usage by disconnecting the circuit-switched line when there is no line activity. See "DEFINX21 Procedure" on page 4-141 for more information. See the manual *Using System/36 Communications* for details of short hold mode.

CLOCK specifies that the system must provide business machine clocking for data communications.

NOCLOCK specifies that the modem or another external source has the clocking facility.

NRZI specifies that NRZI data encoding is performed whenever the system is using SDLC protocol for modems that are sensitive to certain bit patterns in the data stream. NRZI can only be used if the System/36 modem, the remote system modem, and the remote system or device are also using NRZI.

NONRZI specifies that the line is non-NRZI.

CONCAR specifies continuous carrier. This is a feature that holds the 'request-to-send' signal active. Specify CONCAR when modems or modem eliminators are used and the multipoint control station is on a four-wire nonswitched communications facility, or when the interface is a four-wire nonswitched point-to-point communications facility and the modem does not support switched network backup. Also, specify CONCAR if an X.25 network is used.

NOCONCAR specifies that the continuous carrier feature will not be used. Specify NOCONCAR for a multipoint tributary station, for 2-wire point-to-point networks, for a switched public telephone line, or for an interface that uses a Digital Data Service Adapter (DDSA) or uses an X.21 interface adapter on a switched network.

TONE specifies that a non-United States answer tone is required for manual answer and autoanswer communications. This setting is modem dependent.

NOTONE specifies that a non-United States answer tone is not required.

SEP specifies that your autocall unit supports separator characters. When the system attempts to dial a phone number containing a separator character, the character is sent to the autocall unit to do the separation delay.

See the "DEFINEPN Procedure" on page 4-140 for a description of the use of separator characters.

NOSEP specifies that your autocall unit does not support separator characters. When the system attempts to dial a phone number containing a separator character, the character is not sent to the autocall unit. Instead, the system does a 3 second separation delay.

EON specifies that your autocall unit does not have answer tone detection capability. If your autocall unit does not have answer tone detection capability, you should use an end-of-number character in the phone numbers you specify with the DEFINEPN procedure. The end-of-number character will be sent from the system to the autocall unit.

See the "DEFINEPN Procedure" on page 4-140 for a description of the use of end-of-number characters.

NOEON specifies that your autocall unit has answer tone detection capability. If you use an end-of-number character in the phone numbers you specify with the DEFINEPN procedure, the character will not be sent from the system to the autocall unit.

primary sdlc time-out value specifies the primary SDLC time-out value in half-second increments. You can enter any number from 05 through 80, where the first digit indicates the seconds and the second digit represents tenths of seconds and must be 0 or 5. For example, 25 indicates two and one-half seconds and 30 indicates 3 seconds.

sdlc retry count specifies the number of primary SDLC error retries to be attempted. You can enter any value from 1 through 5. That value multiplied by 7 is the number of retries that are to be attempted.

IBMLPDA specifies an IBM modem with link problem determination aid (LPDA) functions. This includes IBM external modems such as the 3833, 3834, 3863, 3864, 3865, 3868, 5812, 5865, 5866, and 5868.

IBMWRAP specifies an IBM modem with wrap test capabilities. This includes IBM external modems such as the 3872, 3874, 3875, 5811, 5841, and 5842.

NONIBM specifies non-IBM modems being used with EIA/CCITT interface adapters.

Note: Most IBM modems run either the IBM LPDA tests or the IBM wrap tests. All other modems must be configured to the System/36 as non-IBM modems. Refer to the modem manuals for information on the tests that are supported.

X25 specifies that the line uses X.25 support.

NOX25 specifies that the line does not use X.25 support.

2400BPS specifies a DDSA (Digital Data Service Adapter/Attachment) line with a line speed of 2400 bps.

4800BPS specifies a DDSA line with a line speed of 4800 bps.

9600BPS specifies a DDSA line with a line speed of 9600 bps.

56KBPS specifies a DDSA line with a line speed of 56000 bps.

secondary sdlc inactivity time-out value specifies the time period that determines when the primary system is to be considered no longer active for a secondary SDLC nonswitched line. You can specify a number from 1-20 to indicate the number of 32-second multiples for the timer, or specify 0 to indicate that no secondary inactivity timer should be used.

token-ring network adapter address override specifies that you want to replace the universally-administered adapter address (preset address) with a locally-administered address (address controlled by you), or to reset the locally-administered address to the universally-administered address.

xxxxxxxxxxxx	The locally-administered adapter address replacing the universally-administered address. The specified hexadecimal address must be within the 400000000000 - 7FFFFFFFFFFF range.
R	Reset the locally-administered address to the universally-administered address.

Example 1

This example sets communications line 1 to be nonswitched point-to-point.

```
SETCOMM ,NONSWTCH
```

Example 2

This example replaces the universally-administered adapter address for communications line 10 with the locally-administered address 7FFFFFFFFFFF.

```
SETCOMM 10,,,,,,,,,,,,,7FFFFFFFFFFF
```

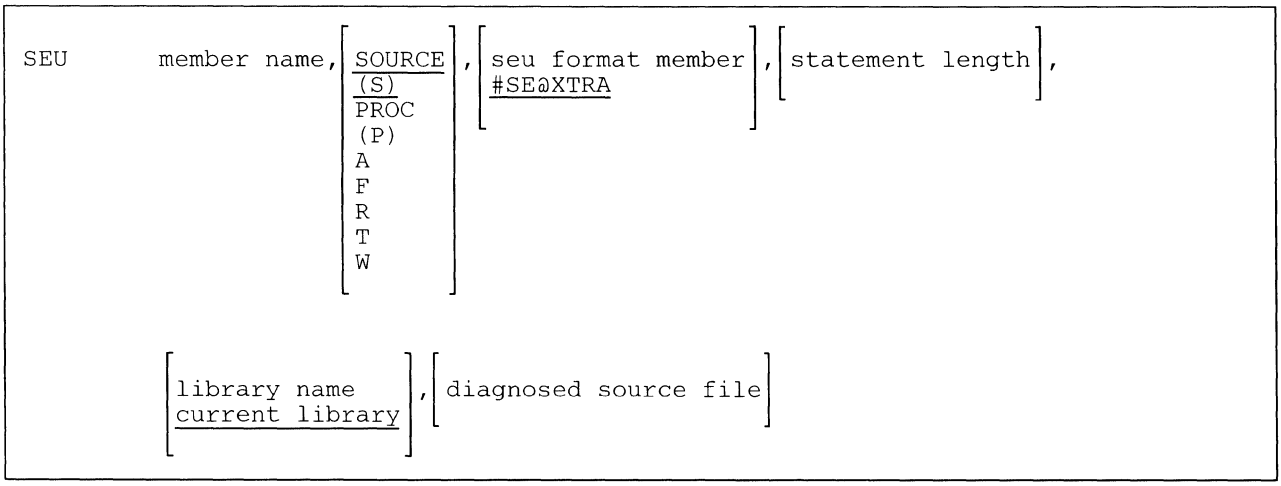
To reset the locally-administered address back to the universally-administered address:

```
SETCOMM 10,,,,,,,,,,,,,R
```

SEU Procedure

The SEU procedure starts the source entry utility (SEU) program. You can use SEU to create or change programs, procedures, message members, menus, or display formats.

SEU is part of the Utilities Program Product. For more information on how to use SEU, see the *SEU Guide*. For information about display formats and menus, see the manual *Creating Displays*. For more information about message members, see the "CREATE Procedure" on page 4-132.



S9020243-0

If no parameters are specified, SEU shows a display to allow you to enter the parameters.

member name specifies the source member or procedure that you want to create or change. A member name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special), except blanks. You should avoid using the following characters because these have special meanings in procedures: commas (,), hyphens (-), apostrophes ('), question marks (?), slashes (/), greater than signs (>), plus signs (+), and equal signs (=). Do not use DIR, LIBRARY, or ALL as a member name.

SOURCE or S specifies a library source member. If no parameter is specified, SOURCE or S is assumed.

PROC or P specifies a library procedure member.

A specifies an RPG II program that contains auto report specifications. The syntax of the RPG II program is checked by SEU. SEU displays the RPG specification for each statement in the source member.

F specifies a library source member that contains display format S- and D-specifications. SEU displays the format specification for each statement in the source member.

R specifies a library source member that contains an RPG II program. The syntax of the RPG II program is checked by SEU. SEU displays the RPG specification for each statement in the source member.

T specifies a message translation source member.

W specifies a source member containing work station utility (WSU) statements. SEU displays the WSU specification for each statement in the source member.

seu format member specifies the name of the SEU display format member to be used to edit source statements. If no format name is specified, #SE@XTRA is assumed.

statement length specifies the length for each source or procedure statement. This can be any decimal number from 40 to 120. If the member exists, the statement length of the member is assumed. If the member is being created, the values you can specify and the values that are assumed by SEU if no statement length is specified are as follows:

Member Type	Allowed Statement Length	Assumed Statement Length
SOURCE, S	40 to 120	96
PROC, P	40 to 120	120
A	80 to 96	96
F	40 to 120	96
R	80 to 96	96
T	40 to 80	80
W	40 to 120	96

library name specifies the library that contains or will contain the library member being changed or created. If no library name is specified, the current library is assumed.

diagnosed source file: This parameter is used by the RPGONL, COBOLONL, and FORTONL procedures to allow the diagnosed source members to be edited. When you start SEU, you need not specify this parameter.

This parameter specifies the name of a disk file that contains source statements and diagnostic messages. The file must have been created by a compiler, such as the RPGC procedure. If a diagnosed source file is specified, SEU uses that file as input and allows you to make changes. When SEU ends, the file, together with any changes you have made, replaces the source member specified in the first parameter.

See your programming manual for more information about diagnosed source members.

Example

This example shows how to start SEU to create or change a procedure named PAYROLL. The procedure is contained in the current library.

```
SEU PAYROLL,P
```

SEULOAD Procedure

The SEULOAD procedure creates a library named #SEULIB and copies the source entry utility (SEU) support from diskette into that library. SEULOAD copies additional support into the system library (#LIBRARY). The SEULOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the SEUSAVE procedure. See the "SEUSAVE Procedure" on page 4-468 for information about how to save the SEU support on diskette.

The SEULOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the SEULOAD procedure to restore support that has been saved by SEUSAVE.

If SEU is not currently on the system, you must use the TOLIBR procedure to copy the diskette file SEU into #LIBRARY before running SEULOAD.

If #LIBRARY was backed up with SEU on the system and then replaced before SEULOAD is run, you do not have to copy the diskette file SEU using the TOLIBR procedure.

```
SEULOAD  [ A1 ],[ S1    ],[ Y ]
         [ A2 ] [ S2    ] [ N ]
         [ A3 ] [ S3    ]
         [ A4 ] [ M1.nn ]
         [    ] [ M2.nn ]
```

S9020244-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #SEULIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #SEULIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified S1, is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Y or N specifies whether the display format source members for #SE@FORM and #SE@XTRA are to be loaded. If no parameter is specified, you will be prompted for the value. If you are prompted and you specify anything other than N, the source members will be loaded. Y specifies that the source members are to be loaded. N specifies that the source members are not to be loaded.

Example

Create #SEULIB on disk and copy the SEU support from diskette.

```
SEULOAD
```

SEUSAVE Procedure

The SEUSAVE procedure copies the source entry utility (SEU) support to diskette. The SEU support from the libraries #SEULIB and #LIBRARY is copied. You should use the "SEULOAD Procedure" on page 4-467 to load the SEU support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPUTIL and be located in diskette slot S1.

```
SEUSAVE
```

S9020245-0

The SEUSAVE procedure has no parameters.

Example

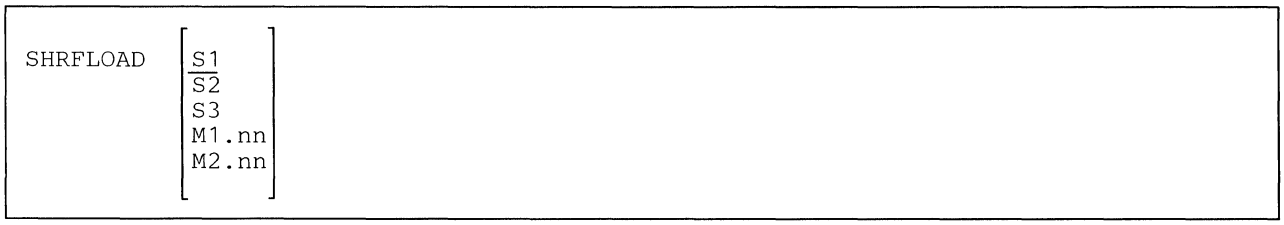
Copy the SEU support to diskette.

```
SEUSAVE
```

SHRFLOAD Procedure

The SHRFLOAD procedure reallocates #IWLIB to 435 blocks and copies the shared folders facility from diskette to the library. Shared folders facility copies additional support to the system library. The SHRFLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the SHRFSAVE procedure. See the "SHRFSAVE Procedure" on page 4-470 for more information about how to save the shared folders facility on diskette.

Note: Shared folders requires PC Support/36 to be on the system to load the shared folders facility. If PC Support/36 is not on the system, you will be prompted to load PC Support/36 before the shared folders facility can be loaded.



S9020602-0

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

SHRFSAVE Procedure

The SHRFSAVE procedure copies the shared folders facility from the libraries #IWLIB and #LIBRARY on diskette. The SHRFLOAD procedure should be used to load the shared folders facility from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPIWS. The diskette must be in slot 1.

```
SHRFSAVE
```

S9020603-0

The SHRFSAVE procedure has no parameters.

SLIB Procedure

The SLIB procedure specifies the name of the library for the display station session. The session library and the current library are the same when you are entering procedures or control commands from the keyboard. You can change the current library for a procedure by using the LIBRARY OCL statement. See the "LIBRARY OCL Statement" on page 5-67 for more information.

The current library is the library that the SSP searches first for procedures, programs, menus, display formats, and message members. If the SSP does not find the member in the current library, the SSP then searches the system library.

A library remains the session library until you sign off the system or:

- Another SLIB procedure is entered.
- The MENU OCL statement or command with a library parameter is processed or entered.
- A LIBRARY OCL statement is entered.
- A LIBRARY OCL statement with SESSION-YES is processed in a procedure.

Once a session library is specified, that library remains allocated to the display station until a different library or the system library is specified. If a different library is specified while the display station is still using one or more members from the previous session library (for example, using a menu from that library), the previous session library remains allocated to the display station until the display station is no longer using members from that library. Consequently, certain functions, such as the CONDENSE procedure, cannot be done on the library even though it is no longer the session library at the display station.

When a program within a procedure is interrupted, the session library is used during the inquiry request. The current library for the interrupted procedure will again be current when the procedure is resumed. For example, the session library is named MYLIB, and a procedure contains the following statement:

```
// LIBRARY NAME-JOBLIB
```

If a program within the procedure is interrupted, the session library (MYLIB, not JOBLIB) will be used during the inquiry request. For information about interrupting a program, see the manual *Operating Your System* for your system unit.

If the SLIB procedure is entered from a MENU display, one or more of the items on the displayed menu might not be correct because the procedures corresponding to those items do not exist in the new session library. Therefore, you should not change libraries without changing the menu. To remove a menu from the display, enter option 0 (zero).

SLIB

The SLIB procedure processes a LIBRARY OCL statement with SESSION-YES specified. See the "LIBRARY OCL Statement" on page 5-67 for more information.

```
SLIB     library name
```

S9020246-0

library name specifies the library that will be the session library. A library name must be specified.

Example

The following procedure specifies that a library called MYLIB is the session library.

```
SLIB MYLIB
```

SMF Procedure

The system measurement facility (SMF) allows you to record and print information about how your system is being used. The SMF procedure displays a menu that allows you to select the SMF task you want to do. The options on the menu are:

- Start the SMF data collection program (the SMFSTART procedure).
- Stop the SMF data collection program (the SMFSTOP procedure).
- Print the collected and analyzed data (the SMFPRINT procedure).
- Write output from an SMF report to a disk file (the SMFDATA procedure).

For more information about SMF, see the *SMF Guide.*

```
SMF
```

S9020247-0

The SMF procedure has no parameters.

Example

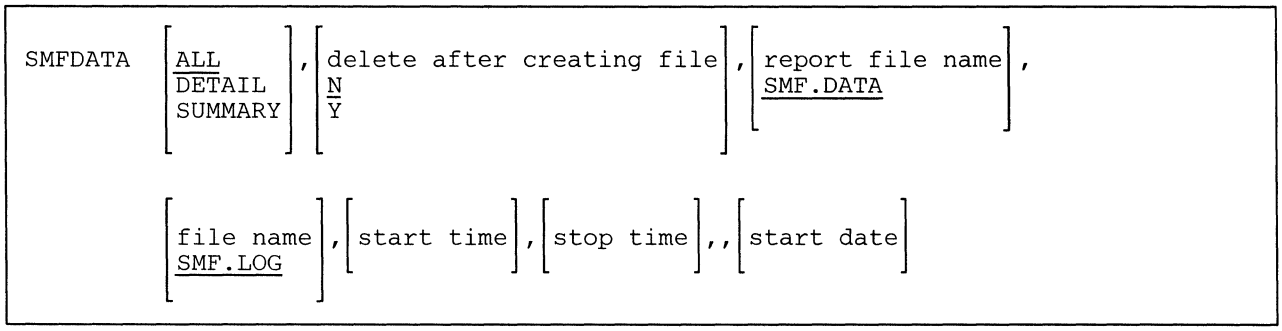
This example shows how to display the SMF menu.

```
SMF
```

SMFDATA Procedure

The SMFDATA procedure writes output from a system measurement facility (SMF) report to a disk file that can be used as input for analysis by application programs.

For more information about the SMFDATA procedure, see the *SMF Guide*.



S9020248-2

ALL, DETAIL, or SUMMARY specifies the type of file to be created by the SMFDATA procedure. If no parameter is specified, ALL is assumed. Figure 4-18 shows the types of information that are contained in each file.

ALL Report File	SUMMARY Report File	DETAIL Report File
IPL configuration information	IPL configuration information	IPL configuration information
Communications configuration data, if active and selected		Communications configuration data, if active and selected
The following information about each sample interval: • Device usage rates • Task work area usage • Disk cache utilization • Task status • I/O and SEC information by task, if selected • Terminated task data • User file access counters • System file access counters • Storage totals • System event counters (SEC) • I/O counters • Data storage attachment (DSA) usage • Communications line usage, if active and selected		The following information about each sample interval: • Device usage rates • Task work area usage • Disk cache utilization • Task status • Storage totals
Summary information	Summary information	Summary information

Figure 4-18. Options for SMF Report Files

delete after creating file specifies whether the SMF data collection file should be removed from the disk after the SMFDATA procedure ends.

N specifies that the file is not to be deleted. If no parameter is specified, N is assumed.

Y specifies that the file is to be deleted.

report file name specifies the name you want for the report file that will contain the data from the data collection file. If no name is specified, SMF.DATA is assumed.

file name specifies the name of the data collection file to be used as input for the SMFDATA procedure. If no name is specified, SMF.LOG is assumed.

start time specifies a beginning time for the SMF data to use. Only data collected after the specified start time is included in the report file. The time must be specified as 6 decimal digits in the form hhmmss, where:

hh specifies the hours, and can be any number from 00 to 23.

mm specifies the minutes, and can be any number from 00 to 59.

ss specifies the seconds, and can be any number from 00 to 59.

The time can be from 000001 to 235959. If no time is specified, the report file should start with the first data item collected by SMF.

stop time specifies the ending time for the SMF data. Only data that has a time before the specified stop time is included in the report file. The time must be specified as 6 decimal digits in the form hhmmss. The time can be from 000001 to 235959. If no time is specified, or 000000 is specified, the report file should stop with the last data item collected by SMF.

start date specifies the beginning date for the SMF data to use. Only data collected after the specified start date is included in the report file. The date must be specified as 6 decimal digits in the form yymmdd, where:

yy specifies the year, and can be any number from 00 through 99.

mm specifies the month, and can be any number from 00 through 12.

dd specifies the day, and can be any number from 00 through 31.

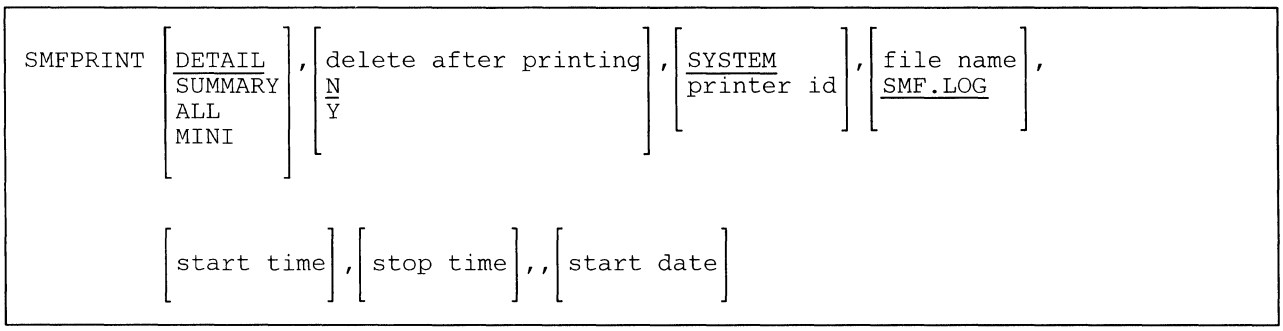
Example

This example shows how to create a report file named SMF.DATA from the data collection file named SMF.LOG. The report file should contain summary data.

```
SMFDATA SUMMARY
```

SMFPRINT Procedure

The SMFPRINT procedure lists a formatted report of the data collected by the system measurement facility (SMF) data collection program. For more information about SMF, see the *SMF Guide*.



S9020249-1

DETAIL, SUMMARY, ALL, or MINI specifies the type of listing to be created by the SMFPRINT procedure. If no parameter is specified, DETAIL is assumed. Figure 4-19 shows the information listed by each type.

DETAIL	SUMMARY	ALL	MINI
IPL configuration information	IPL configuration information	IPL configuration information	IPL configuration information
Communications configuration data, if active and selected		Communications configuration data, if active and selected	Communications configuration data, if active and selected
The following information about each sample interval: • Device usage rates • Task work area usage • Disk cache utilization • Task status • Storage totals		The following information about each sample interval: • Device usage rates • Task work area usage • Disk cache utilization • Task status • I/O and SEC information by task, if selected • Terminated task data • User file access counters • System file access counters • Storage totals • System event counters (SEC) • I/O counters • Data storage attachment (DSA) usage • Communications line usage, if active and selected	The following information about each sample interval: • Device usage rates • Task work area usage • Significant system event and I/O counters
Summary information	Summary information	Summary information	Summary information

Figure 4-19. Listing Options for SMFPRINT

delete after printing specifies whether the SMF data collection file is to be removed after the data is printed. Either Y or N can be specified.

N specifies not to delete the file. If no parameter is specified, N is assumed.

Y specifies to delete the file.

SYSTEM specifies that the system printer is to be used to print the output. If no parameter is specified, SYSTEM is assumed.

printer id specifies the work station ID of the printer on which the output is to be printed. You can use the STATUS WORKSTN command to determine the printer IDs.

file name specifies the SMF disk file that contains the data to be listed. If no name is specified, the name SMF.LOG is assumed.

start time specifies the beginning time for the SMF data to use. Only data collected after the specified start time is included in the listing. The time must be specified as 6 decimal digits in the form hhmmss, where:

hh specifies the hours, and can be any number from 00 to 23.

mm specifies the minutes, and can be any number from 00 to 59.

ss specifies the seconds, and can be any number from 00 to 59.

The time can be from 000001 to 235959. If no time is specified, or 000000 is specified, the listing should start with the first data item collected by SMF.

stop time specifies the ending time for the SMF data. Only data that has a time before the specified stop time is included in the listing. The time must be specified as 6 decimal digits in the form hhmmss. The time can be from 000001 to 235959. If no time is specified, or 000000 is specified, the listing should stop with the last data item collected by SMF.

start date specifies the beginning date for the SMF data to use. Only data collected after the specified start date is included in the listing. The date must be specified as 6 decimal digits in the form yymmdd, where:

yy specifies the year, and can be any number from 00 through 99.

mm specifies the month, and can be any number from 00 through 12.

dd specifies the day, and can be any number from 00 through 31.

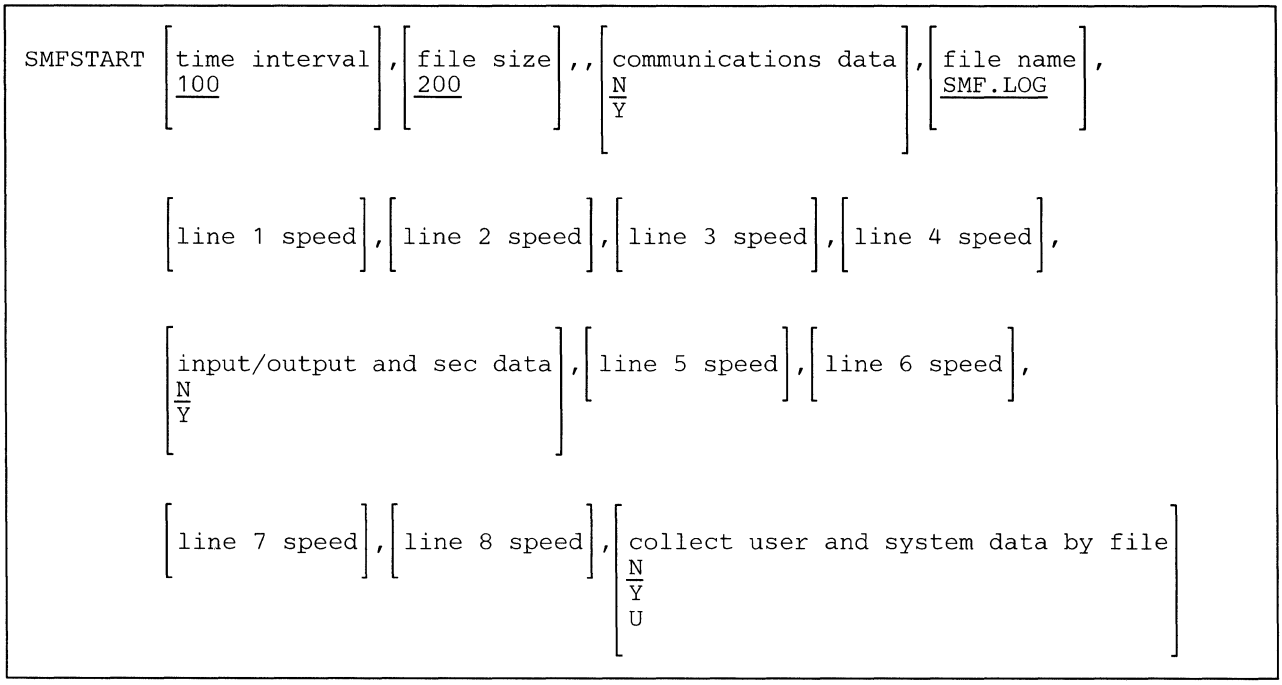
Example

This example shows how to print the SMF.LOG data collection file. A summary listing is to be printed.

```
SMFPRINT SUMMARY
```

SMFSTART Procedure

The SMFSTART procedure starts the system measurement facility (SMF) data collection program. For more information about SMF, see the *SMF Guide*.



S9020250-0

time interval specifies the time interval to use for the data collection. The time is specified in the format: MSS, where M is minutes (0 through 5) and SS is seconds (0 through 59). For example, 010 specifies a 10 second interval; 500 specifies a 5 minute interval. If no time is entered, a time interval of 1 minute is assumed.

file size specifies the size of the SMF data collection file in blocks. Any number from 1 through 312815 can be specified. If no file size is specified, a size of 200 is assumed.

The third parameter position is reserved for compatibility with the IBM System/34.

communications data specifies whether data related to the usage of communications is to be collected. Either N or Y can be specified.

N specifies that no communications is to be collected. If no parameter is specified, N is assumed.

Y specifies that the data is to be collected.

file name specifies the file that is to contain the collected SMF data. If no file name is specified, SMF.LOG is assumed.

line 1 speed specifies the speed of communications line 1 in bits per second. If no value is entered, no data is collected for that line.

line 2 speed specifies the speed of communications line 2 in bits per second. If no value is entered, no data is collected for that line.

line 3 speed specifies the speed of communications line 3 in bits per second. If no value is entered, no data is collected for that line.

line 4 speed specifies the speed of communications line 4 in bits per second. If no value is entered, no data is collected for that line.

input/output and sec data specifies whether additional information is to be collected, such as disk input/output operations, display station or printer input/output operations, system event counters, or the number of job steps started. Either N or Y can be specified.

N specifies that no additional data is to be collected. If no parameter is specified, N is assumed.

Y specifies that the additional data is to be collected.

line 5 speed specifies the speed of communications line 5 in bits per second. If no value is entered, no data is collected for that line.

line 6 speed specifies the speed of communications line 6 in bits per second. If no value is entered, no data is collected for that line.

line 7 speed specifies the speed of communications line 7 in bits per second. If no value is entered, no data is collected for that line.

line 8 speed specifies the speed of communications line 8 in bits per second. If no value is entered, no data is collected for that line.

collect user and system data by file specifies whether additional information is to be collected for user and system files. This information includes the file and job names, the block location of the file, and the number of reads and writes to, and scans of the file. N, Y, or U can be specified.

N specifies that no additional data is to be collected. If no parameter is specified, N is assumed.

Y specifies that additional data is to be collected for both system and user files.

U specifies that additional data is to be collected for user files only.

Example

This example shows how to start the SMF data collection program. The data is to be collected at 15 second intervals. Communications data for line 1 is to be collected, and line 1 has a speed of 1920 bits per second.

```
SMFSTART 015,,,Y,,1920
```

SMFSTOP Procedure

The SMFSTOP procedure stops the system measurement facility (SMF) data collection program. For more information about SMF, see the *SMF Guide*.

```
SMFSTOP
```

S9020251-0

The SMFSTOP procedure has no parameters.

Example

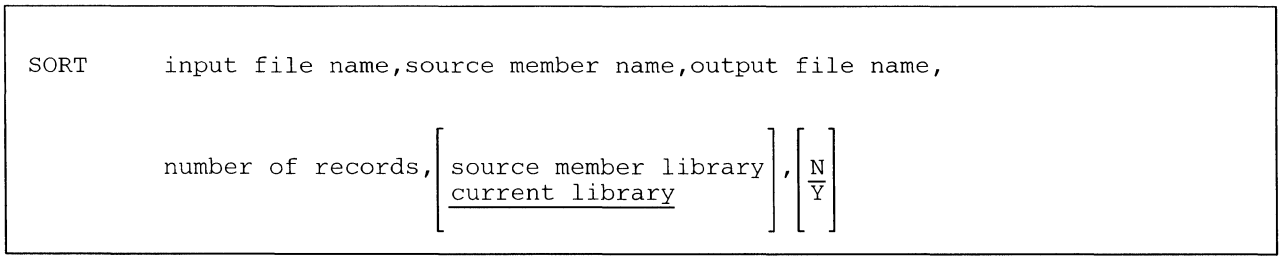
To stop the SMF data collection program, enter:

```
SMFSTOP
```

You can then run the SMFPRINT procedure to print the data collected. See "SMFPRINT Procedure" on page 4-477.

SORT Procedure

The SORT procedure allows you to sort data contained in disk files. For more information about the SORT procedure and the sort program, see the *Sort Guide*.



S9020252-0

input file name specifies the file to be sorted.

source member name specifies the source member that contains the sort specifications to be used to sort the file.

output file name specifies the file that is to contain the sorted records.

number of records specifies the number of records to allocate to the output file. The number can be from 1 through 8000000.

source member library specifies the name of the library that contains the source member. If no library name is specified, the current library is assumed.

N specifies that the job is not to be placed on the job queue, that is, the job is to be run from the display station that requested the procedure. If no parameter is specified, N is assumed.

Y specifies that the job is to be run from the job queue.

Example

This example shows how to sort a file named FILE1. The sorted file is to be named FILE2. The sort specifications are in a source member named SORTNAME that is in the library MYLIB. FILE2 is to contain up to 200 records.

```
SORT FILE1,SORTNAME,FILE2,200,MYLIB
```

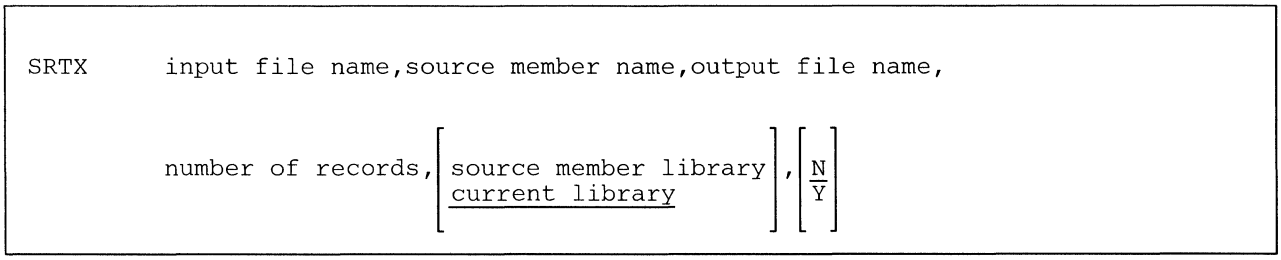
SPECIFY Procedure

This procedure is supported only for compatibility with the IBM System/34. Use the "ALTERCOM Procedure" on page 4-11 to change communications information.

SRTX Procedure

The SRTX procedure allows you to sort ideographic data contained in disk files. For more information about the SRTX procedure and the sort program, see the *Ideographic Sort Guide.*

Note: The SRTX procedure is supported only for Japanese extended character files and does not support extended character files for Taiwan, Korea, or People's Republic of China.



S9020253-0

input file name specifies the file to be sorted.

source member name specifies the library source member that contains the sort specifications to be used to sort the file.

output file name specifies the file that is to contain the sorted records.

number of records specifies the number of records to allocate to the output file. The number can be from 1 through 8000000.

source member library specifies the name of the library that contains the source member. If no library name is specified, the current library is assumed.

N specifies that the job is not to be placed on the job queue, that is, the job is to be run from the display station that requested the procedure. If no parameter is specified, N is assumed.

Y specifies that the job is to be run from the job queue.

Example

This example shows how to sort a file named FILE1. The sorted file is to be named FILE2. The sort specifications are in a source member named IDEOGRPH that is in the library MYLIB. FILE2 is to contain 200 records.

```
SRTX FILE1,IDEOGRPH,FILE2,200,MYLIB
```

SRTXBLD Procedure

The SRTXBLD procedure combines the #KACTIVE file and the #KAMAST file so that you can sort ideographic data contained in disk files. For more information about the SRTXBLD procedure and the sort program, see the *Ideographic Sort Guide*.

Note: The SRTXBLD procedure is supported only for Japanese extended character files and does not support extended character files for Taiwan, Korea, or People's Republic of China.

```
SRTXBLD
```

S9020254-0

The SRTXBLD procedure has no parameters.

SRTXLOAD Procedure

The SRTXLOAD procedure creates a library named #SRTXLIB and copies the ideographic sort program from diskette into that library. SRTXLOAD copies the #KACTIVE file and the #KAMAST file if specified. The SRTXLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the SRTXSAVE procedure. See the "SRTXSAVE Procedure" on page 4-488 for information about how to save the ideographic sort program and the #KACTIVE and #KAMAST files on diskette.

The SRTXLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the SRTXLOAD procedure to restore support that has been saved by SRTXSAVE.

If SRTX is not currently on the system, you must use the TOLIBR procedure to copy the diskette file SRTX into #LIBRARY before running SRTXLOAD.

If #LIBRARY was backed up with SRTX on the system and then replaced before SRTXLOAD is run, you do not have to copy the diskette file SRTX using the TOLIBR procedure.

```
SRTXLOAD  [ A1 ]  [ S1 ]  { restore #KACTIVE file }  { restore #KAMAST file }
          [ A2 ]  [ S2 ]  { Y                     }  { Y                    }
          [ A3 ]  [ S3 ]  { N                     }  { N                    }
          [ A4 ]  [ M1.nn ]
                  [ M2.nn ]
```

S9020255-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If no location is specified and the system has more than one disk unit, the file is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

restore #KACTIVE file specifies whether the #KACTIVE file is to be restored.

Y specifies that the #KACTIVE file is to be restored. If the file already exists on disk, a message asks if you want to replace the file on disk.

N specifies that the #KACTIVE file is not to be restored.

restore #KAMAST file specifies whether the #KAMAST file is to be restored. If the file already exists on disk, a message asks if you want to replace the file on disk.

Y specifies that the #KAMAST file is to be restored.

N specifies that the #KAMAST file is not to be restored.

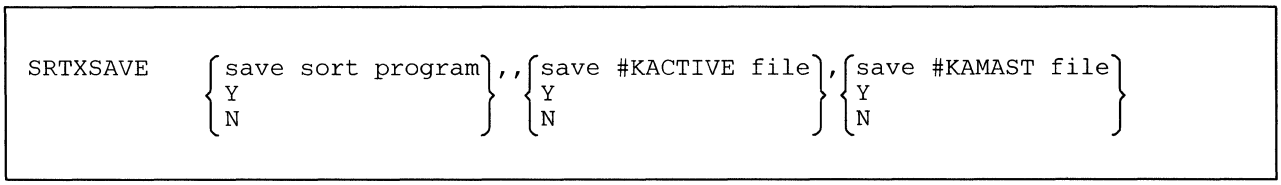
Example

This example shows how to load the #KACTIVE file from diskette.

```
SRTXLOAD Y,N
```

SRTXSAVE Procedure

The SRTXSAVE procedure copies the ideographic sort program and the #KACTIVE and #KAMAST files to diskette. The ideographic sort program from the library #SRTXLIB is copied. You should use the "SRTXLOAD Procedure" on page 4-486 to load the ideographic sort program from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPSSP and be located in diskette slot S1.



S9020256-1

save sort program specifies whether the ideographic sort program is to be copied to diskette.

Y specifies that the ideographic sort program is to be copied to diskette.

N specifies that the ideographic sort program is not to be copied to diskette.

The second parameter is ignored.

save #KACTIVE file specifies whether the #KACTIVE file is to be copied to diskette.

Y specifies that the #KACTIVE file is to be copied to diskette.

N specifies that the #KACTIVE file is not to be copied to diskette.

save #KAMAST file specifies whether the #KAMAST file is to be copied to diskette.

Y specifies that the #KAMAST file is to be copied to diskette.

N specifies that the #KAMAST file is not to be copied to diskette.

Example

This example shows how to save the #KACTIVE file to diskette.

```
SRTXSAVE N,,Y,N
```

STARTM Procedure

The STARTM procedure is used to start automatic monitoring of a BSC multipoint line. The System/36 responds to all host polls or selects on a BSC multipoint line with a negative acknowledgment. A nonswappable BSC interrupt handler is not required in main storage for the automatic monitoring function. Therefore, the System/36 can be placed in the host's polling list without any tasks in the System/36 being started until necessary. For more information on the STARTM procedure, see the manual *Using System/36 Communications*.

Automatic monitoring continues until stopped by the STOPM procedure, or until either a batch BSC job is started or an SSP-ICF BSC subsystem is enabled on that line.

The STARTM procedure runs the $MMST utility program.

```
STARTM   line number,{E},station address
                     {A}
```

S9020257-0

line number specifies the line number that is placed in automatic monitor mode. This line must be a multipoint line.

E specifies that the transmission code is EBCDIC.

A specifies that the transmission code is ASCII.

station address specifies the 2-character hexadecimal multipoint station address (either the poll or select sequence can be specified). See the "ALTERCOM Procedure" on page 4-11 for a list of valid hexadecimal multipoint station addresses.

Example

This example shows how to specify line 1 to be auto-monitored for an EBCDIC station address of C4.

```
STARTM 1,E,C4
```

STATEST Procedure

The STATEST procedure tests the communications line and the controller used for remote display stations and printers. It also tests SSP-ICF Finance controllers and SSP-ICF secondary Peer connections. (System/36 is the primary station for APPC or SSP-ICF Peer.)

You can run the test on either a switched or nonswitched line. On a nonswitched line, you can test up to seven secondary remote stations with a single run of the test. The autocall feature cannot be used for this test.

The system you are running from will send out a test command to the remote locations you specify. If the remote stations respond properly, the transmission is successful. If the remote stations do not respond properly, or do not respond at all, the transmission is not successful.

After you have begun the test transmissions, no remote work station can be placed online until the test has completed. If you wish to end the test, press command
key 7.

For information about running the STATEST procedure, see the manual *Using System/36 Communications.*

```
STATEST
```

S9020258-0

The STATEST procedure has no parameters.

Example

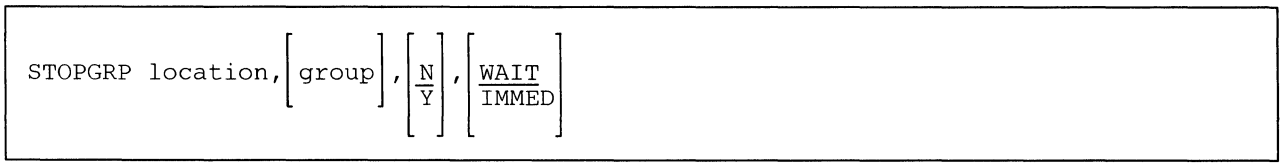
To start the STATEST procedure, enter:

```
STATEST
```

STOPGRP Procedure

The STOPGRP procedure stops a single session group or all session groups configured for a remote location using the advanced program-to-program communications (APPC) or advanced peer-to-peer networking (APPN) subsystems. The session group remains inactive until another STRTGRP procedure is run to start the session group. All sessions associated with the session group end normally or immediately when the STOPGRP procedure is run. For more information on the STOPGRP procedure, see the manuals *Using System/36 Communications* or *Advanced Peer-to-Peer Networking Guide*.

For more information about the APPC subsystem, see the manuals *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference*, *Interactive Communications Feature: Upline Subsystems Reference*, *Interactive Communications Feature: Finance Subsystem Reference*, and *Interactive Communications Feature: Base Subsystems Reference*.



S9020260-1

location specifies the name of the remote location where the session group(s) are to be stopped.

group specifies the name of a single session group to be stopped. If no parameter is specified, all session groups configured for the specified remote location are stopped.

N specifies that the APPC subsystem does not allow the remote location to complete any requested activity before ending the sessions in the group. If no parameter is specified, N is assumed.

Y specifies that the APPC subsystem allows the remote location to complete any requested activity before ending the sessions in the group.

WAIT specifies that the session groups are stopped when the current activity stops.

IMMED specifies that all sessions are terminated even if some sessions are active.

STOPM Procedure

This procedure stops the automatic monitoring function of a BSC multipoint line. The line being monitored could have been previously placed in automatic monitor mode in one of three ways:

- By the STARTM procedure
- By disabling an SSP-ICF BSC subsystem
- By the ending of a batch BSC job

The STOPM procedure runs the $MMSP utility program.

```
STOPM    line number
```

S9020259-0

line number specifies that automatic monitor mode is to be stopped for the specified line. This line must be a multipoint line.

Example

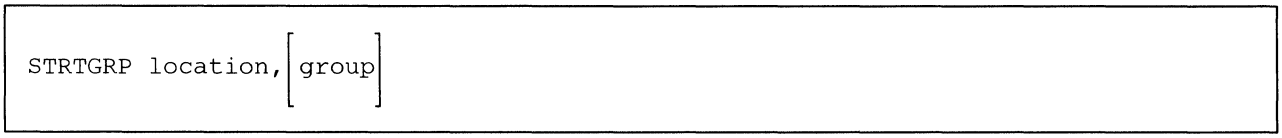
To stop auto-monitoring line 1, enter:

```
STOPM 1
```

STRTGRP Procedure

The STRTGRP procedure starts a single session group or all session groups configured for a remote location using the advanced program-to-program communications (APPC) or advanced peer-to-peer networking (APPN) subsystems. For more information on the STRTGRP procedure, see the manuals *Using System/36 Communications* or *Advanced Peer-to-Peer Networking Guide.*

For more information about the APPC subsystem, see the manuals *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference*, *Interactive Communications Feature: Upline Subsystems Reference*, *Interactive Communications Feature: Finance Subsystem Reference*, and *Interactive Communications Feature: Base Subsystems Reference.*



S9020261-0

location specifies the name of the remote location where the session group(s) are to be started.

group specifies the name of a single session group to be started. If no parameter is specified, all session groups configured for the specified remote location are started.

SWDLOAD Procedure

The SWDLOAD procedure creates a library named #SWLIB and copies software distribution support from diskette into that library and the system library, #LIBRARY. Two files, SWPROD and SWBASE, are also copied from diskette. File SWBASE contains all IBM product definitions, and file SWPROD contains user product definitions.

The SWDLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the SWDSAVE procedure. See the "SWDSAVE Procedure" on page 4-495 for information about how to save software distribution support on diskette.

The SWDLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the SWDLOAD procedure to restore support that has been saved by SWDSAVE.

If software distribution is not currently on the system, you must use the TOLIBR procedure to copy the diskette file IWS into #LIBRARY before running SWDLOAD.

If #LIBRARY is backed up with software distribution on the system and then replaced before SWDLOAD is run, you do not have to copy the diskette file IWS using the TOLIBR procedure.

```
SWDLOAD  [A1 ]  [S1   ]
         [A2 ]  [S2   ]
         [A3 ]  [S3   ]
         [A4 ]  [M1.nn]
                [M2.nn]
```

S9020630-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough disk space is available on that unit, the other units (if they exist) are checked, and the files SWPROD and SWBASE and library #SWLIB are placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, the files are placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

Example

Create the files SWPROD and SWBASE and library #SWLIB on disk and copy software distribution support from diskette.

```
SWDLOAD
```

SWDSAVE Procedure

The SWDSAVE procedure copies software distribution support onto diskette. Software distribution support from the libraries #SWLIB and #LIBRARY is copied. You should use the "SWDLOAD Procedure" on page 4-494 to load software distribution support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPSWD and be located in diskette slot 1.

```
SWDSAVE
```

S9020631-0

The SWDSAVE procedure has no parameters.

Example

Copy software distribution support to diskette.

```
SWDSAVE
```

SWITCH Procedure

The SWITCH procedure sets one or more of the user program status indicator (UPSI) switches for the display station to on (1) or off (0). The switch setting remains in effect until the display station session ends or one of the following occurs:

- Another SWITCH procedure is processed
- A SWITCH OCL statement is processed
- A program changes the setting of any of the indicators

All switches are set to off when a display station session begins; that is, when an operator signs on.

A job placed on the job queue uses a copy of the switches for the display station as they existed when the job was placed on the queue.

You can use the STATUS SESSION command to determine the current switch settings. To determine the switch settings in a procedure, see the "SWITCH (Switches) Condition" on page 3-48.

Notes:

1. *If an SSP procedure changes the setting of a switch, the switch is returned to its original setting when the SSP procedure ends.*
2. *A set of switches exists for each running multiple requester terminal (MRT) procedure.*

The SWITCH procedure processes a SWITCH OCL statement.

```
SWITCH   switch settings
```

S9020262-0

switch settings specifies how the switches are to be set, and consists of 8 characters, one for each of the eight UPSI switches. The first, or leftmost, character gives the setting of switch 1; the second character gives the setting of switch 2; and so on.

The parameter must always contain 8 characters. For each switch, one of the following characters must be used:

Character	Meaning
0 (zero)	Set the switch off
1 (one)	Set the switch on
X	Leave the switch as it is

Example

This example shows how to set the switches to the following:

Switch	Switch Setting	Result
1	1	Set on
2	X	Unaffected
3	0	Set off
4	1	Set on
5	1	Set on
6	0	Set off
7	X	Unaffected
8	X	Unaffected

```
SWITCH 1X0110XX
```

SYSLIST Procedure

The SYSLIST procedure changes the method of listing system list output. System list output is generated by all SSP utility programs except for the following:

- Data communications utility programs
- The service aid procedures

The SYSLIST procedure causes the system list output to be:

- Listed on the printer that was assigned to the display station during system configuration
- Listed on one of the other printers
- Displayed at the display station
- Not listed at all

You can use the PRINT procedure to have all your printed output go to a specified printer. See the "PRINT Procedure" on page 4-350 for more information.

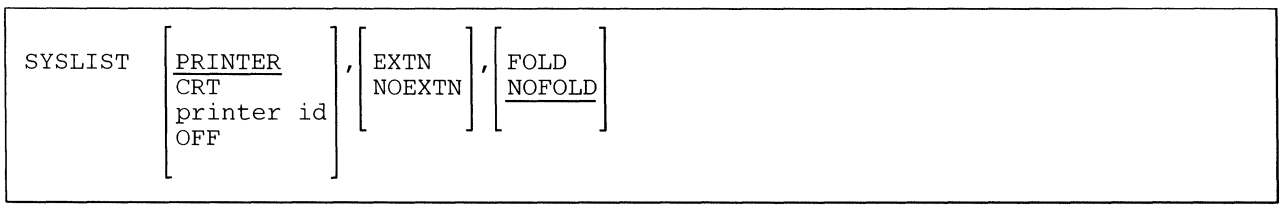
The SYSLIST assignment remains in effect until you sign off the system or the assignment is changed by one of the following:

- Another SYSLIST procedure
- The PRINT procedure
- A SYSLIST OCL statement

Notes:

1. *If the SYSLIST procedure is run during inquiry mode (the operator pressed the Attn key), the change made by the SYSLIST procedure is in effect only during inquiry mode.*
2. *The SYSLIST procedure does not change the system list device if debugging is turned on. For more information refer to the "DEBUG OCL Statement" on page 5-28.*

The SYSLIST procedure processes a SYSLIST OCL statement.



PRINTER specifies that system list output is to be listed on the printer that was assigned to the display station during system configuration, by the SET procedure, or by the $SETCF utility program. If no parameter is specified, PRINTER is assumed.

CRT specifies that system list output is to be displayed at the display station. At least the first 80 columns of output are displayed. Depending on the model of the display station, up to 21 lines are displayed at a time.

After each system list display, the operator can request that the next set of lines be displayed or that the display be ended. If the operator enters a blank, the number of lines shown on the next display will be the same as the number of lines shown on the previous display. If the operator enters 0, the display will be ended. If the operator enters any characters other than a blank or 0 through 18 (21 for the 3180 Model 2 display station), the next set of lines is displayed.

printer id specifies that system list output is to be printed on the printer with the specified printer ID. Printer IDs are assigned during system configuration and can be modified by the ASSIGN control command. To print output on the system printer, specify the printer ID of the system printer.

OFF specifies that any system list output is not to be printed or displayed.

EXTN specifies that the extended characters in the system list output are to be printed or displayed.

NOEXTN specifies that the extended characters in the system list output are not to be printed or displayed. The system-defined default ideographic character will be listed for any extended character.

Note: The EXTN and NOEXTN parameters are for the ideographic version of the SSP and are ignored for nonideographic systems. When a session begins, extended character processing for system list output is assumed. During the session, the EXTN and NOEXTN parameters can be used to turn on and turn off extended character processing for system list output.

FOLD specifies that system list output displayed at the display station is not to be truncated if it exceeds the column width of the display station. Instead, data after column position 75 is continued on the next line of the display (for the 3180 Model 2 display station, data after column position 130 is continued on the next line).

NOFOLD specifies that system list output displayed at the display station is truncated if it exceeds the column width of the display station. If no parameter is specified, NOFOLD is assumed.

Example

Print output on the printer assigned to the display station.

```
SYSLIST PRINTER
```

TAPECOPY Procedure

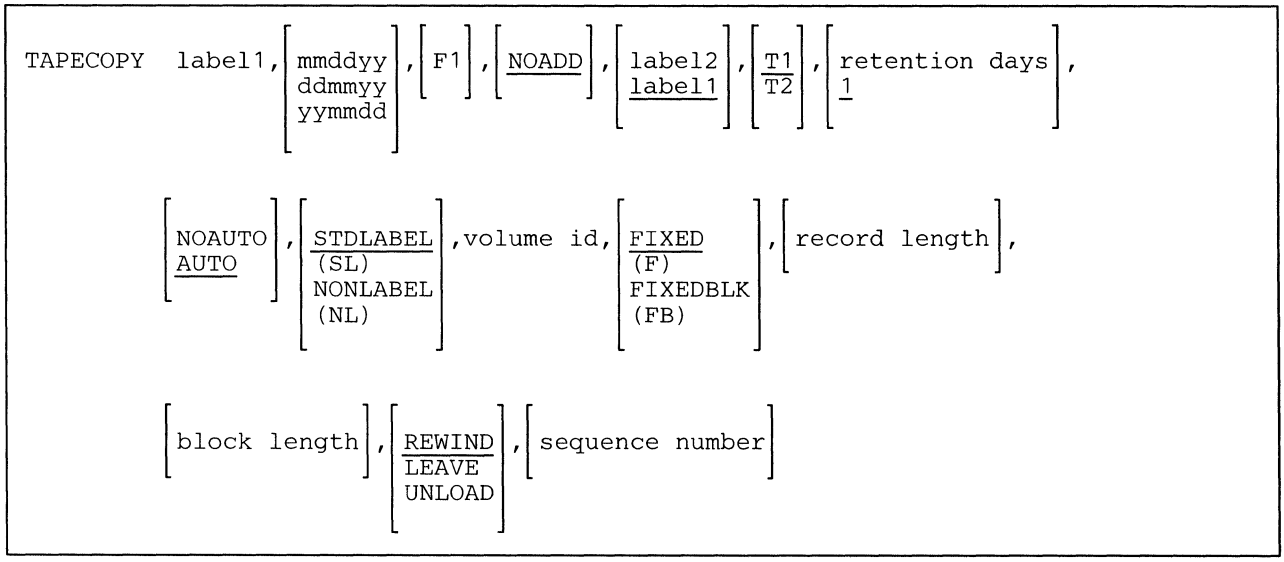
The TAPECOPY procedure can be used to do the following:

- Copy a disk file to an exchange tape file
- Add a file from disk to an existing exchange tape file
- Copy an exchange file from tape to a disk file
- Add an exchange file from tape to an existing disk file

Note: TAPECOPY cannot be used with a tape cartridge (TC) drive.

The TAPECOPY procedure runs the $TCOPY utility program.

To copy a disk file to an exchange tape file:



S9020264-0

To add a file from disk to an existing exchange tape file:

```
TAPECOPY  label1,[mmddyy|ddmmyy|yymmdd],[F1],{ADD|ADDNOCHK},[label2|label1],[T1|T2],[mmddyy|ddmmyy|yymmdd],

          [NOAUTO|AUTO],[STDLABEL|(SL)|NONLABEL|(NL)],volume id,[FIXED|(F)|FIXEDBLK|(FB)],[record length],

          [block length],[REWIND|LEAVE|UNLOAD],[sequence number]
```

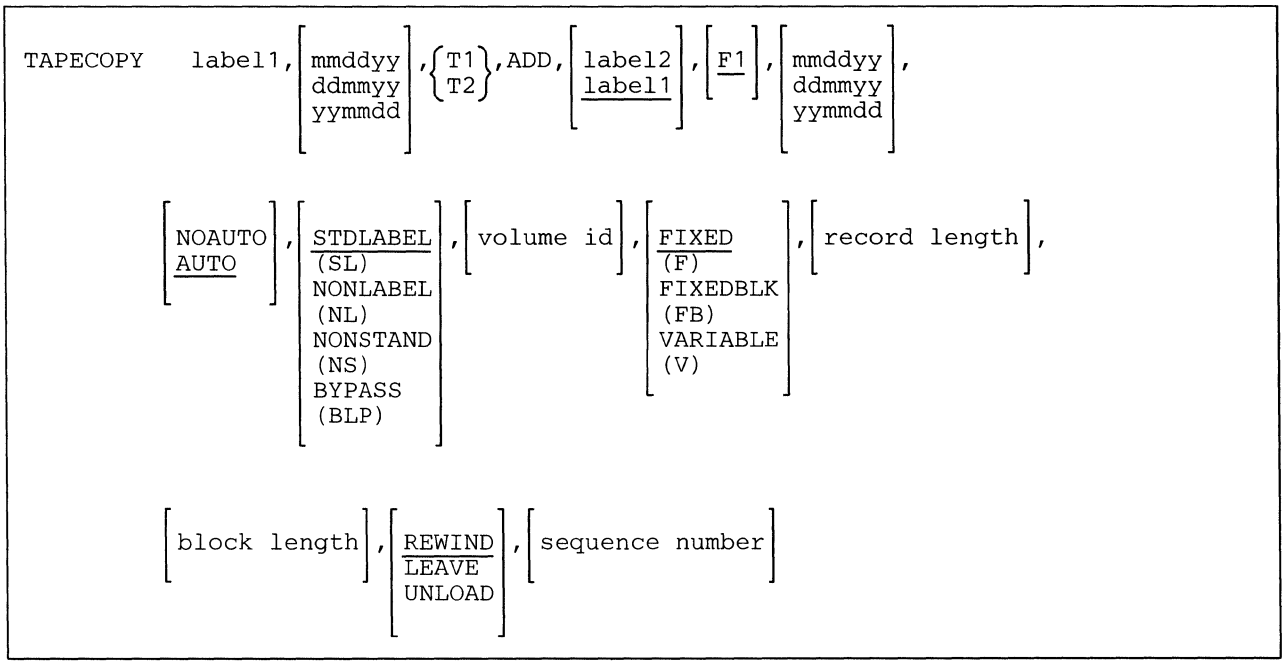
S9020265-0

To copy an exchange file from tape to a disk file:

```
TAPECOPY  label1,[mmddyy|ddmmyy|yymmdd],{T1|T2},[NOADD],[label2|label1],[F1],,

          [NOAUTO|AUTO],[STDLABEL|(SL)|NONLABEL|(NL)|NONSTAND|(NS)|BYPASS|(BLP)],[volume id],[FIXED|(F)|FIXEDBLK|(FB)|VARIABLE|(V)],[record length],

          [block length],[REWIND|LEAVE|UNLOAD],[sequence number],{RECORDS,records|BLOCKS,blocks},

          [key length],[key location],[NODUPKEY|DUPKEY]
```

S9020266-1

To add an exchange file from tape to an existing disk file:



S9020267-0

label1 specifies the label of the file to be copied. If the file being copied is a disk file, the label cannot exceed 8 characters. If the file being copied is a tape file, the number of characters cannot exceed 17. This parameter is required for a standard label tape, but an error message will be issued if you specify label1 for a tape that is not a standard label tape.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file. If specified for the second parameter, it refers to the creation date of the file being copied. If specified for the seventh parameter, it refers to the creation date of an existing file and is valid only if ADD or ADDNOCHK is specified. The date must be specified in the same format as the session date. If more than one file exists with the specified label and a creation date is not specified, the most recent date is used if the file is a disk file or the first file found is used if the file is a tape file.

F1 specifies a disk file. If specified for the third parameter, F1 indicates that a disk file is being copied or added to a tape file. If specified for the sixth parameter, F1 indicates that a tape file is being copied or added to a disk file. If T1 or T2 is specified for the third parameter, F1 is assumed for the sixth parameter.

T1 or T2 specifies a tape file. If specified for the third parameter, T1 or T2 indicates that a tape file is being copied or added to a disk file. If specified for the sixth parameter, T1 or T2 indicates that a disk file is being copied or added to a tape file. If F1 is specified for the third parameter, T1 is assumed for the sixth parameter. T1 indicates that processing should begin with the tape mounted on tape drive 1. T2 indicates that processing should begin with the tape mounted on tape drive 2. TC is not allowed with the TAPECOPY procedure.

ADD specifies that the file specified in the first parameter is to be added to the file specified in the fifth parameter. If a tape file is being added to, a check is made to ensure that the tape file is the last file on the tape. If the tape file is not the last file, an error message is issued. This error allows you to continue, which allows you to add to the tape file; however, any files after the file being added to will be lost.

ADDNOCHK specifies that the file specified in the first parameter is to be added to an existing file specified in the fifth parameter. ADDNOCHK is valid only when adding to a tape file and no check is made to see that it is the last file on the tape. Any files after the file being added to will be lost.

NOADD specifies that the file specified in the first parameter will be used to create a new output file. If this parameter is omitted, NOADD is assumed.

label2 specifies the label of the file. If ADD or ADDNOCHK is specified, this parameter refers to the label of an existing disk or tape file to which records will be added. IF NOADD is specified, this parameter refers to the label of the new disk or tape file being created. If the file is a disk file, the label cannot exceed eight characters. If the file is a tape file, the label cannot exceed 17 characters. If the label specified does exceed these limits, it will be truncated. If this parameter is omitted, the label specified in parameter one (label1) is assumed.

retention days specifies the retention period for the newly created tape file. If this parameter is not specified, one day is assumed. If a retention period of 999 is specified, the file becomes a permanent file. This parameter is valid only for tape files.

NOAUTO If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

AUTO If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 or T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

If no parameter is specified, AUTO is assumed.

STDLABEL or SL specifies that the tape to be processed is a standard label tape. If a tape file is to be copied or added to a disk file, or a disk file is to be added to an existing tape file, most of the information needed to process the tape can be taken from the tape file label. If a disk file is to be copied to a tape file, all information needed to process the file must be supplied or taken from the disk file. If this parameter is omitted, STDLABEL is assumed.

NONLABEL or NL specifies that the tape to be processed is a nonlabeled tape and all the information needed to process the file must be provided.

NONSTAND or NS specifies that the tape to be processed has nonstandard labels and all the information needed to process the file must be provided. NONSTAND is valid only when a tape file is being copied or added to a disk file.

BYPASS or BLP specifies that the tape has standard tape labels, but label processing is to be bypassed. All the information needed to process the file must be supplied. BYPASS is only valid when a tape file is being copied or added to a disk file.

volume id specifies the volume identification of the tape to be processed. The volume ID can be from 1 through 6 alphameric characters. It is used to check that the correct tape volume is mounted. If the correct tape volume is not mounted, an error message is issued. The operator can continue and process the file, retry after mounting the correct tape, or cancel the procedure.

The volume ID parameter is required if a disk file is being copied to or added to a file on a standard label tape. This parameter is not allowed if NONLABEL, NONSTAND, or BYPASS is specified.

If a disk file is being copied or added to a tape file and the file requires more than one tape volume, the volume ID of the first volume is checked.

FIXED or F specifies that the record format of the file to be processed is fixed length, unblocked records. If the record format is not specified, FIXED is assumed, except when reading or adding to a standard label file (REEL-SL) that has a HDR2 label. If the standard label file has a HDR2 label, the record format specified in the HDR2 label is used.

FIXEDBLK or FB specifies that the record format of the file to be processed is fixed length, blocked records.

VARIABLE or V specifies that the record format of the file to be processed is variable length, unblocked records. VARIABLE is valid only when a tape file is being copied or added to a disk file.

Note: If the type of processing is STDLABEL and the record format specified (or defaulted to) does not match the record format in the tape file label, an error message is issued.

record length specifies the number of bytes in a logical tape record. For variable length records this is the maximum length. The value specified can be from 18 to 4096 bytes.

The record length parameter is required if NONLABEL, NONSTAND, BYPASS, or STDLABEL and the file being processed does not have an HDR2 label, is specified and a tape file is being copied or added to a disk file. Also, the record length parameter is required if NONLABEL (NL) is specified, and a disk file is being added to (not copied to) a tape file.

If the record length is not specified when a new tape file is created, the record length will be the same as the disk file. If the record length is specified when a new tape file is created, and it is not the same as the disk file record length, an error message is issued. The job can be canceled or the job can be continued and the disk record will be padded or truncated to the record length specified. If the record length is not specified when a tape file is being copied or added to a disk file or a disk file is being added to a tape file, the record length is taken from the tape file label. If the record length is specified, it must be the same as the record length in the tape file label.

block length specifies the number of bytes in a physical block of data of the tape file. The value specified can be from 18 to 32767 bytes. This parameter is required if FIXEDBLK (FB) was specified and NONLABEL (NL), NONSTAND (NS), or BYPASS (BLP) was specified. It is also required when a new tape file is created and the record format is FIXEDBLK (FB).

The block length is required if the tape is a STDLABEL (SL) tape that does not have an HDR2 label. The block length is optional when a tape file is being copied or added to a disk file or a disk file is being added to a tape file and STDLABEL (SL) is specified. If the block length is not specified, it is taken from the tape label. If the block length is specified, it must be the same as the block length in the tape file label.

The block length is not allowed if FIXED (F) or VARIABLE (V) is specified. If you specify a block length greater than the region size, an error message is issued.

REWIND specifies that the tape should be rewound after the TAPECOPY procedure has run. REWIND is assumed if this parameter is not specified.

LEAVE specifies that the tape should be left where it is after the TAPECOPY procedure has run. The next step within a procedure accessing the same tape unit starts at this position. LEAVE information is maintained by the system from job step to job step, but is **not** passed from job to job.

UNLOAD specifies that the tape should be rewound and unloaded after the TAPECOPY procedure has run.

sequence number specifies which file on the tape to process by its position on the tape. If this parameter is omitted, and NONLABEL, NONSTAND, or BYPASS is specified, a sequence number of 1 is assumed. If this parameter is omitted, and STDLABEL is specified, the file will be found by its file label. If a sequence number is specified, and STDLABEL is specified, the file is located by sequence number first, and then checked by the file label. If it is not the correct file, the file is found by its file label.

RECORDS,value1 specifies that the disk file being created must be large enough to contain the number of records specified by value1. You can specify a value from 1 through 8000000. If a tape file is being copied to a new disk file, either a RECORDS,value1 or a BLOCKS,value2 is required.

BLOCKS,value2 specifies that the disk file being created must be large enough to contain the number of blocks specified by value2. You can specify a value from 1 through the maximum number of blocks of disk storage configured on the system. If a tape file is being copied to a new disk file, either a RECORDS,value1 or a BLOCKS,value2 is required.

key length specifies the key length of the indexed file that is being created. The value specified can be any decimal number from one through 120. The key length must be specified with the key location, and the sum of key length and key location cannot exceed the record length plus one. This parameter is not allowed when you create or add to a tape file or add to a disk file.

key location specifies the relative displacement of the start position of the record key for an indexed file that is being created. The key location must be specified with the key length and the sum of the key location and key length cannot exceed the record length plus one. This parameter is not allowed when you create or add to a tape file or add to a disk file.

DUPKEY specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified when you create an indexed disk file, NODUPKEY is assumed.

NODUPKEY specifies that duplicate keys are not to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified when you create an indexed disk file, NODUPKEY is assumed.

Example 1

This example takes a disk file called FILE1 and copies it to tape. The name of the tape file is to be FILE2. The tape is processed as a standard label tape with a volume ID of VOL001 and is mounted on tape drive 1. The tape is rewound after copying the file.

```
TAPECOPY FILE1,,F1,NOADD,FILE2,T1,,,STDLABEL,VOL001
```

Example 2

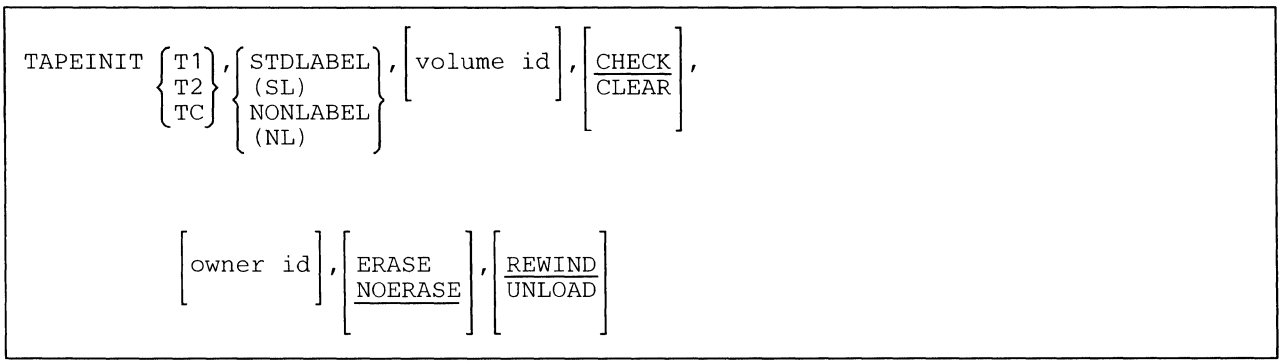
This example takes a file from a tape that contains 500 records and copies it to a disk file. The name of the tape file is FILE2, and the file to be created on disk is called FILE1. The tape is a standard label tape with a volume ID of VOL001 mounted on tape drive 1. The tape is rewound after copying the file.

```
TAPECOPY FILE2,,T1,,FILE1,F1,,,,VOL001,,,,,,RECORDS,500
```

TAPEINIT Procedure

The TAPEINIT procedure prepares a tape so that it can be used to save files and libraries. This preparation is called **initialization**. You can also use this procedure to erase a tape.

The TAPEINIT procedure runs the $TINIT utility program.



S9020268-1

T1 specifies that the tape to be prepared is on tape drive 1.

T2 specifies that the tape to be prepared is on tape drive 2.

TC specifies that the tape to be prepared is a tape cartridge. STDLABEL (SL) is the only tape format allowed if TC is specified.

STDLABEL or SL specifies that a labeled tape is to be prepared.

NONLABEL or NL specifies that an nonlabeled tape is to be prepared.

volume id specifies the volume identification that is written on the tape. Up to 6 alphameric characters can be specified. The parameter is required if STDLABEL (SL) is specified and not allowed if NONLABEL (NL) is specified.

CLEAR specifies that a new volume label is to be written without checking for an expired file.

CHECK specifies that the first data file is checked to see if it is an expired file. If it is, a new volume label is written. If it is not an expired file, an error message will be issued. CHECK should not be specified for blank tapes. If this parameter is not specified, CHECK is assumed.

owner id specifies an additional identification field. Up to 14 characters can be specified. If an owner ID is not specified, the field is left blank. The owner ID is not allowed if NONLABEL is specified.

ERASE specifies that after the new volume label has been written, blanks are to be written to the end of the tape.

NOERASE specifies that blanks are not to be written to the end of the tape after the new volume label has been written. If this parameter is not specified, NOERASE is assumed. This parameter is not valid if TC is specified.

REWIND specifies that after the tape is prepared, it should be rewound. REWIND is assumed if this parameter is not specified.

UNLOAD specifies that after the tape is prepared, it should be rewound and unloaded. If UNLOAD and TC are specified, the tape will be rewound after processing.

Example

This example shows how to initialize a tape on tape drive 1. The tape should be initialized as a standard label tape with a volume id of VOL001. The tape should be checked to ensure that the first data file is an expired file. The tape should be erased and then rewound and unloaded.

```
TAPEINIT T1,SL,VOL001,,,ERASE,UNLOAD
```

TAPESTAT Procedure

The TAPESTAT procedure can be used to display or print information about the tape volumes that was recorded in the volume statistical logs. If you are having trouble reading from or writing to a specific tape, the TAPESTAT procedure can be used to help determine what type of errors the tape is producing.

If the tape to be checked is a reel on tape drive 1 or tape drive 2, the following information can be displayed or printed:

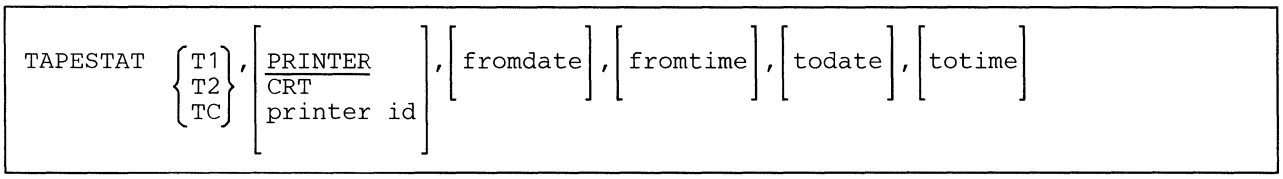
- Tape unit used
- The time span for the record encountered
- The volume serial number
- Number of K bytes read
- Number of K bytes written
- Number of permanent read errors
- Number of permanent write errors
- Number of temporary read errors
- Number of temporary write errors
- Number of read error retries
- Number of write error retries
- Number of overruns
- Number of velocity checks
- Number of multitrack errors
- Number of end data checks
- Number of start read checks
- Number of readback failures
- Number of envelope checks
- Number of no pointer errors
- Number of crease errors
- Number of skew errors
- Number of track 4 errors
- Number of track 5 errors
- Number of parity check errors

If the tape to be checked is a tape cartridge, the following information can be displayed or printed:

- Tape unit used
- The time span for the record encountered
- The volume serial number
- The date
- The time
- Number of K bytes read
- Number of K bytes written
- Number of permanent read errors
- Number of permanent write errors
- Number of data errors
- Number of underruns
- Number of read parity errors
- Number of write parity errors
- Number of read parity retries
- Number of write parity retries

TAPESTAT

The TAPESTAT procedure runs the $FEAIDS utility program.



S9020269-1

T1 specifies that the tape to be checked is on tape drive 1.

T2 specifies that the tape to be checked is on tape drive 2.

TC specifies that the tape to be checked is a tape cartridge.

PRINTER specifies that the system printer is to be used to print the report. If this parameter is not specified, PRINTER is assumed.

CRT specifies that the display station from which the procedure was entered will be used to display the report. Only the first 80 characters of the report are displayed.

printer id specifies that the report is to printed on the printer with the specified printer ID. Printer IDs are assigned during system configuration and can be modified by the ASSIGN control command.

fromdate specifies the date of the earliest entry on the tape to be included in the report. The date must be specified in the session date format. If this parameter is not specified, the date of the entries is not checked.

fromtime specifies the time of the earliest entry on the tape to be included in the report. The time must be specified in hhmmss (hours-minutes-seconds) format. If this parameter is not specified, the time of the entries is not checked.

todate specifies the date of the latest entry on the tape to be included in the report. The date must be specified in the session date format. If this parameter is not specified, the date of the entries is not checked.

totime specifies the time of the latest entry on the tape to be included in the report. The time must be specified in hhmmss (hours-minutes-seconds) format. If this parameter is not specified, the time of the entries is not checked.

Example

This example shows how to check the tape on tape drive 1. All entries on the tape should be checked. The report should be printed on printer P2.

```
TAPESTAT T1,P2
```

TEXTCONV Procedure

The TEXTCONV procedure starts the Text Conversion Aid to convert documents created by the Text Management System (TMS) to a form that can be used by DW/36.

```
TEXTCONV
```

S9020487-0

The TEXTCONV procedure has no parameters.

TEXTDCT Procedure

The TEXTDCT procedure allows you to maintain a supplemental dictionary using DW/36.

For more information about DW/36, see the online information. To read the online information:

- Run the READINFO procedure. For DW/36, specify:

  ```
  READINFO #WPDOC,#WPFLDR
  ```

```
TEXTDCT
```

S9020270-0

The TEXTDCT procedure has no parameters.

TEXTDOC Procedure

The TEXTDOC procedure allows you to create or maintain a document using DW/36.

For more information about DW/36, see the online information. To read the online information:

- Run the READINFO procedure. For DW/36, specify:

  ```
  READINFO #WPDOC,#WPFLDR
  ```

If you enter TEXTDOC with no parameters, the *Work with documents* display is shown.

To create or maintain a document:

```
TEXTDOC [CREATE  ],[document name],[folder name],[subdirectory]
        [DELETE  ]
        [PAGINATE]
        [REVISE  ]
        [VIEW    ]
```

S9020271-1

To copy a document to another document:

```
TEXTDOC COPY,[document name],[folder name],[new document name  ],
                                           [document name      ]

      [new folder name  ],[NOREPLACE],[subdirectory],[new subdirectory]
      [folder name      ],[REPLACE   ]
```

S9020272-2

To merge data from other documents or files into your document:

```
TEXTDOC MERGE,[document name],[folder name],[new document name  ],
                                           [document name      ]

      [new folder name  ],[NOREPLACE],[NOOPTIONS],[subdirectory],[new subdirectory]
      [folder name      ],[REPLACE   ],[OPTIONS  ]
```

S9020527-2

To print a document:

```
TEXTDOC PRINT,[document name],[folder name],[NOOPTIONS / OPTIONS],[subdirectory]
```

S9020273-1

To rename a document:

```
TEXTDOC RENAME,[document name],[folder name],[new document name],[subdirectory]
```

S9020274-1

To print documents to a disk file:

```
TEXTDOC PRTFILE,[document name / ALL],[folder name],[CHECK / NOCHECK],[select status],
    [new status],[TEXT / NOTEXT],[filename],[ERRLOG / NOERRLOG],[DELETE / NODELETE],[subdirectory]
```

S9020528-2

To check the spelling in a document:

```
TEXTDOC  SPELL,[document name],[folder name],[beginning page],[ending page],
    [subdirectory]
```

S9020558-1

CREATE specifies that you want to create a new document.

DELETE specifies that you want to remove a document from a folder.

PAGINATE specifies that you want to adjust the page endings for a document.

REVISE specifies that you want to change a document.

VIEW specifies that you want to look at a document but not change it.

COPY specifies that you want to copy a document to another document. You can copy to the same folder if a different document name is specified.

MERGE specifies that you want to merge data on files into another document.

PRINT specifies that you want to print a document.

RENAME specifies that you want to give an existing document a new name.

PRTFILE specifies that you want to print one or more documents to a disk file.

SPELL specifies that you want to check the spelling in your document. This will be done as a batch job.

document name specifies the document you want to work with. If PRTFILE is specified, document name specifies the name of the document you want to have output to a file.

ALL specifies that you want to process all text documents within a folder or all text documents with a given status value within a folder. ALL is only valid if PRTFILE is specified.

folder name specifies the folder in which the document is stored or is to be stored. If PRTFILE is specified, folder name specifies the name of the folder which contains the documents you want to process. If this parameter is not specified, the user profile folder will be used.

new document name specifies the name of the new document. A document name can be up to 12 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because they have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), and hyphens (-). Do not use DIR, LIBRARY, or ALL as a document name.

subdirectory specifies the name of the subdirectory to be used. Subdirectory is stored in your profile and defaults if you do not specify it. If you have previously used a subdirectory but now want to work out of the root directory, you are required to use a forward slash (/) character for the subdirectory prompt to override the profile value.

beginning page specifies the page that you want the spelling check to begin on. This parameter is valid only if SPELL is specified.

ending page specifies the page that you want the spelling check to end on. This parameter is valid only if SPELL is specified.

NOOPTIONS specifies that the default print options are to be used for the document to be printed. If no parameter is specified, NOOPTIONS is assumed. If the first parameter is MERGE, NOOPTIONS specifies the merge will take place with the default MERGE options.

OPTIONS specifies that the print options are to be displayed before the document is printed. If you change the print options when they are displayed, the changes will only affect the printing of this document: the changes are not saved. If the first parameter is MERGE and OPTIONS is specified, the *Merge Options* display will be shown and you can specify options that affect how the information is merged into the new document.

new folder name specifies the name of the folder where the copied document is to be stored. If you are giving the new document the same name as the previous document, you must specify a name different from the previous folder. If this parameter is not specified, *folder name* is used.

A folder name has the same restrictions as a document name but can only be 8 characters long.

NOREPLACE specifies that if a document with the same name already exists, an error message will be issued. If no parameter is specified, NOREPLACE is assumed.

REPLACE specifies that if a document with the same name already exists, it will be replaced.

CHECK specifies that the document status is to be checked. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, the prompt will default from the user profile.

NOCHECK specifies that you do not want to make selections based on the document status. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, the prompt will default from the user profile.

select status specifies the status field of the text documents you want to be selected. This parameter is only valid if CHECK is specified. If this parameter is not specified, the prompt will default from the user profile.

new status specifies the new value for the status field of the documents selected. If you do not want the status to be changed, specify the same value for this parameter as for the select status parameter. This parameter is only valid if CHECK is specified. If this parameter is not specified, the prompt will default from the user profile.

new subdirectory specifies the name of the new subdirectory to be used.

TEXT specifies that you want the document text to be included with the document information in the file. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, the prompt will default from the user profile.

NOTEXT specifies that you want only the document description records. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, it will default from the user profile.

file name specifies the name of the disk file you want to print your document to. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, it will default from the user profile.

ERRLOG specifies that you want the document error log to be included with the document. This will let you see if any problems were encountered while the output disk file was created. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, the prompt will default from the user profile.

NOERRLOG specifies that you do not want the document error log to be included in the document. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, it will default from the user profile.

DELETE specifies that you want the document to be deleted after the disk file output is created. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, the prompt will default from the user profile.

NODELETE specifies that you want the document to remain in the folder. This parameter is only valid if PRTFILE is specified. If this parameter is not specified, NODELETE is assumed.

File Format for PRTFILE Files

For each DW/36 document copied to a disk file, one header record is created as the first record in that file. All records in PRTFILE files have a record length of 256 bytes.

Header Record

The header record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	1	B	'20'X
2	2	C	PF
4	253	C	Unused (unused positions are set to blanks)

Following the header record are the document records (record identifiers 10 through 99). There will be a start document record as the first record for each DW/36 document printed to the file.

Start Document Record

The start document record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (10)
3	4	C	Form (PF = print file)
7	12	C	Document name
19	8	C	Folder name
27	62	C	Subdirectory
89	168	C	Unused (unused positions are set to blanks)

Document Description Records

Following the document record are document description records. The first document description record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (11)
3	60	C	Subject
63	35	C	Author
98	35	C	Receiver
133	1	C	Online (1 = yes, 2 = no)
134	16	C	Class
150	8	C	Date complete
158	8	C	Last revised date
166	8	C	Creation date
174	8	C	Action due date
182	8	C	Retention date
190	10	C	Project
200	10	C	Reference
210	16	C	Keyword
226	1	C	Status
227	2	B	Internal status ('8000'X = print as labels, '4000'X = object (graphics), '0000'X = other)
229	6	C	Last revised time
235	22	C	Unused (unused positions are set to blanks)

The second document description record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (12)
3	35	C	Unused (unused positions are set to blanks)
38	60	C	Capitalized subject
98	13	C	Original system
111	44	C	Original name
155	8	C	Unused (unused positions are set to blanks)
163	1	C	Reassign text names (1 = yes, 2 = no)
164	1	C	Delete text/controls (1 = yes, 2 = no)
165	2	B	DIA type
167	17	C	Unused (unused positions are set to blanks)
184	8	C	Archive file name
192	8	C	Archive date
200	6	C	Archive volume ID
206	1	C	Archive device (1=diskette, 2=tape)
207	2	B	Type available Bit 16 = DW/36 unresolved Bit 15 = DW/36 resolved Bit 14 = L2-DCA (FFT) Bit 13 = L3-DCA (RFT) Bit 12 = PC data Bit 11 = DW/4 internal
209	1	C	PC mark for archive (1 = yes, 2 = no)
210	1	C	PC read only (1 = yes, 2 = no)
211	1	C	PC system flag (1 = yes, 2 = no)
212	1	C	PC hidden document (1 = yes, 2 = no)
213	44	C	Unused (unused positions are set to blanks)

General Format Record

The general format record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (20)
3, 5, 7	2	B	Dictionaries 01 = American English 02 = UK English 03 = German 04 = Dutch 05 = National French 06 = Canadian French 07 = Italian 08 = Spanish 09 = Swedish 11 = Danish 12 = Norwegian 14 = Icelandic 51 = American English Legal 76 = American English Medical
9	4	B	System text unit name
13	2	B	Text unit name processing
15	2	B	Right-to-left data exists (0 = no, 1 = yes)
17	168	C	Month names (array of 12 names, each 14 characters long)
185	1	C	Change symbol character
186	4	C	Print forms number
190	8	C	File/query name
198	8	C	Query library
206	2	B	Graphic Character Set ID
208	2	B	Code Page Global ID
210	47	C	Unused (unused positions are set to blanks)

Page Format Record

The page format record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (30)
3	2	B	Format ID number
5	8	C	Format name
13	40	C	Description
53	4	B	Page width
57	4	B	Page depth
61	2	B	First typing line for first page
63	2	B	First typing line for subsequent pages
65	2	B	Last typing line
67	2	B	First page forms type (0 = current, 1 = paper, 2 = envelope)
69	2	B	Other pages forms type (0 = current, 1 = paper, 2 = envelope)
71	2	B	First page paper source (0 = current, # = drawer number)
73	2	B	Other pages paper source (0 = current, # = drawer number)
75	2	B	Paper feed
77	2	B	Offset stack
79	2	B	Destination drawer
81	2	B	Quality of print
83	2	B	Duplex print
85	1	C	Header all pages (1 = yes, 2 = no)
86	1	C	Header even pages (1 = yes, 2 = no)
87	1	C	Header odd pages (1 = yes, 2 = no)
88	1	C	Header on first page (1 = yes, 2 = no)
89	1	C	Footer on all pages (1 = yes, 2 = no)
90	1	C	Footer on even pages (1 = yes, 2 = no)
91	1	C	Footer on odd pages (1 = yes, 2 = no)
92	1	C	Footer on first page (1 = yes, 2 = no)
93	2	B	First line for header

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
95	2	B	First line for footer
97	4	B	Unused
101	3	C	Degrees to rotate, blanks imply auto
104	153	C	Unused (unused positions are set to blanks)

Line Format Record

The line format record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (40)
3	2	B	Format ID number
5	8	C	Format name
13	40	C	Description
53	1	C	Adjust (1 = yes, 2 = no)
54	1	C	Print half index up (superscript)/half index down (subscript)/word underscore as spaces (1 = yes, 2 = no)
55	1	C	Print underline/bold as spaces (1 = yes, 2 = no)
56	1	C	Auto hyphen (1 = yes, 2 = no)
57	1	C	Compress wide paragraphs (1 = yes, 2 = no)
58	2	B	Typestyle number (font ID)
60	2	B	Lines per inch
62	2	B	Document left-most left margin
64	2	B	Left margin
66	2	B	Right margin
68	2	B	Spacing 1 = one half 2 = single 3 = one and one half 4 = double 5 = two and one half 6 = triple
70	2	B	Zone width
72	2	B	Line alignment (justification percent)
74	2	B	Number of tabs (0 through 48)
76	96	B	Array of tab positions (2 bytes per tab)
172	48	C	Array of tab types (1 character per tab) 3 = left 4 = right 5 = center 6 = decimal 7 = comma 8 = colon

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
220	2	B	Line numbering state 1 = on, resume on next line 2 = on, reset on next line 3 = off, continue numbering 4 = off, suspend numbering
222	2	B	Line orientation 0 = left-to-right 1 = right-to-left
224	2	B	Color 0 = base 1 = blue 2 = red 3 = pink 4 = green 5 = turquoise 6 = yellow 8 = black 16 = brown
226	2	B	Font width (in 1440ths of an inch)
228	2	B	Font spacing 0 = based on font ID 1 = mono 2 = proportional
230	2	B	Graphic character set ID
232	2	B	Code page global ID
234	23	C	Unused (unused positions are set to blanks)

Text Lines

The following records define text lines (record identifiers 50 through 79). If there are more than 235 bytes of text in a line, a second record will be created with the start position of 236. The second record will have the required carrier return indicator set to 1 for yes or 2 for no, but the required carrier return indicator for the first record will be blank (indicating that the line did not end on this record).

Text Half Index Up (Superscript) Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (50)
3	2	B	Page number
5	2	B	Line number
7	2	B	Print position of first character in line (1, 236, etc.)
9	1	C	Margin text (1 = yes, 2 = no)
10	2	B	Position of last non-blank character in line
12	1	C	Required carrier return (1 = yes, 2 = no)
13	9	C	Unused (unused positions are set to blanks)
22	235	C	Line

Text Bold Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (51)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Up (Superscript) Underline Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (52)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Up (Superscript) Overstrike Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (53)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Up (Superscript) Backspace Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (54)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Base Line Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (60)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Base Bold Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (61)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Base Underline Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (62)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Base Overstrike Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (63)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Base Backspace Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (64)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Down (Subscript) Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (70)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Down (Subscript) Bold Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (71)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Down (Subscript) Underline Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (72)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Down (Subscript) Overstrike Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (73)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Text Half Index Down (Subscript) Backspace Record

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (74)[1]
[1] The remaining columns and field lengths are the same as described for the "Text Half Index Up (Superscript) Record."			

Instruction Record

The instruction record is used for the Change Font control, the Set Color control, and any unrecognized or pass through controls. The instruction record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (80)
3	2	B	Page number
5	2	B	Line number
7	1	C	Continuation record follows (1 = yes, 2 = no)
8	2	B	Position of instruction
10	2	B	Length of instruction
12	10	C	Unused (unused positions are set to blanks)
22	235	C	Instruction (may contain values less than '40'X)

Start Error Log Record

The start error log record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (90)
3	4	C	Form
7	12	C	Document
19	8	C	Folder
27	62	C	Subdirectory
89	168	C	Unused (unused positions are set to blanks)

End Document Record

The end document record contains the following fields:

Beginning Column	Field Length	Character (C) or Binary (B)	Contents or Description
1	2	C	Record identifier (99)
3	4	C	Form
7	12	C	Document
19	8	C	Folder
27	62	C	Subdirectory
89	168	C	Unused (unused positions are set to blanks)

TEXTFLDR Procedure

The TEXTFLDR procedure allows you to create or maintain a folder using DW/36. If you enter TEXTFLDR with no parameters the *Work with Folders* display is shown.

For more information about DW/36, see the online information. To read the online information:

- Run the READINFO procedure. For DW/36, specify:

  ```
  READINFO #WPDOC,#WPFLDR
  ```

S9020275-0

folder name specifies the folder you want to work with. If you want to create a folder, specify a name that does not already exist on your system. A folder name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because they have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), equal signs (=), and hyphens (-). Do not use DIR, LIBRARY, or ALL as a folder name.

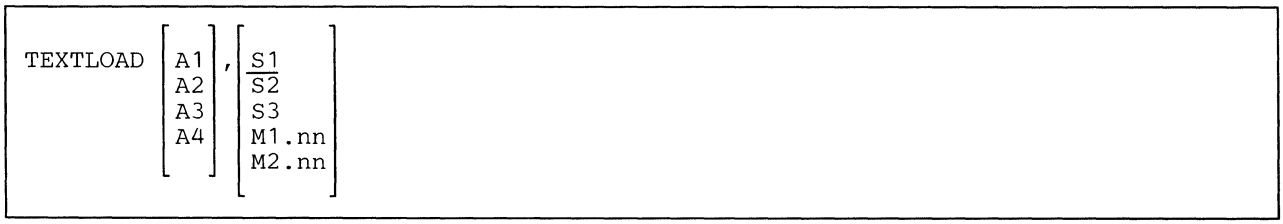
TEXTLOAD Procedure

The TEXTLOAD procedure creates a library named #TULIB and copies the DW/36 support from diskette into that library. TEXTLOAD copies additional support into the system library (#LIBRARY). The TEXTLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the TEXTSAVE procedure. See the "TEXTSAVE Procedure" on page 4-536 for information about how to save the DW/36 support on diskette.

The TEXTLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the TEXTLOAD procedure to restore support that has been saved by TEXTSAVE.

If DW/36 is not currently on the system, you must use the TOLIBR procedure to copy the diskette file TXT into #LIBRARY before running TEXTLOAD.

If #LIBRARY was backed up with DW/36 on the system and then replaced before TEXTLOAD is run, you do not have to copy the diskette file TXT using the TOLIBR procedure.



S9020276-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #TULIB is placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, #TULIB is placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #TULIB on disk and copy the DW/36 support from diskette. #TULIB is to be placed on the second disk if space is available.

```
TEXTLOAD A2
```

TEXTOBJ Procedure

The TEXTOBJ procedure allows you to maintain document objects using DW/36.

S9020593-0

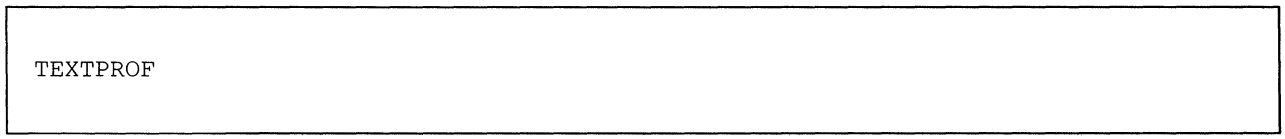
folder name specifies the folder that contains the document objects you wish to work with. If this parameter is not specified, the procedure will use the last used folder name stored in the user profile.

TEXTPROF Procedure

The TEXTPROF procedure allows you to create or maintain DW/36 user profiles.

For more information about DW/36, see the online information. To read the online information, run the READINFO procedure. For DW/36, specify:

```
READINFO #WPDOC,#WPFLDR
```



S9020619-0

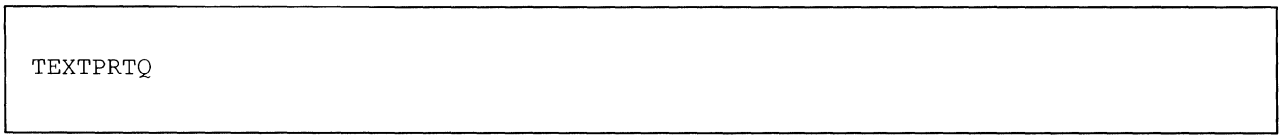
The TEXTPROF procedure has no parameters.

TEXTPRTQ Procedure

The TEXTPRTQ procedure allows you to perform various print tasks. You can view, hold, release, cancel, change, or move jobs that are waiting on the print queue to be printed. The *Work with documents to be printed* screen is displayed when TEXTPRTQ is entered on a command line.

For more information about DW/36, see the online information. To read the online information, run the READINFO procedure. For DW/36, specify:

```
READINFO #WPDOC,#WPFLDR
```



S9020559-0

The TEXTPRTQ procedure has no parameters.

TEXTREL Procedure

The TEXTREL procedure releases documents that have been held for later printing.

For more information about DW/36, see the online information. To read the online information, do any of the following:

- Run the READINFO procedure. For DW/36, specify:

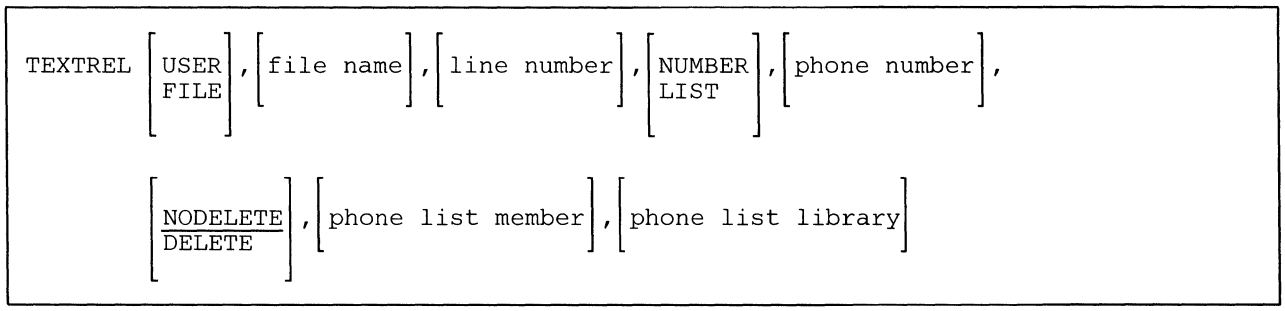
  ```
  READINFO #WPDOC,#WPFLDR
  ```

- Select option 11 from the Use DW/36 menu
- Press the Help key from any DW/36 display

To release all documents for a user identification:

S9020548-0

To release a file and send it to a printer:



S9020277-2

USER specifies that you want to release all the documents for your user ID.

FILE specifies that you want to release a specific file that contains a document and send it to a 6670 or 6580 printer.

user id specifies the user identification for the documents to be released. The TEXTREL procedure can be run only from the system console to release documents for other user identifications. Users can release documents for their own user identification from any display station. If user id is not specified, the documents for your user ID are released. This parameter is valid only if USER is specified as the first parameter.

file name specifies the name of the file to be released. This parameter is valid only if FILE is specified as the first parameter.

line number specifies the number of the communications line over which a file containing a document is to be sent to be printed. If no parameter is specified, 1 is assumed.

NUMBER specifies that the system console operator is to call to establish communications with a printer.

LIST specifies that the autocall feature is to be used, and that the printer is to be called from a phone list.

phone number specifies the telephone number of the printer to which a file containing a document is to be sent.

NODELETE specifies that you do not want the file containing a document to be removed from the system after the document has been printed. If no parameter is specified, NODELETE is assumed.

DELETE specifies that you want the file containing a document to be removed from the system after the document has been printed.

phone list member specifies the name of the phone list member that contains the telephone number of the printer to be called.

phone list library specifies the name of the library that contains the phone list member.

TEXTSAVE Procedure

The TEXTSAVE procedure copies only the DW/36 support to diskette. The DW/36 support from the libraries #TULIB and #LIBRARY is copied. You should use the "TEXTLOAD Procedure" on page 4-531 to load the DW/36 support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPTXT.

Note: TEXTSAVE only copies DW/36 support from the system. TEXTSAVE does not save optional SSP support that DW/36 uses, which may be on the system (for example, print online information). To save optional SSP support that may be needed by DW/36, #TULIB and #LIBRARY should also be saved.

```
TEXTSAVE
```

S9020278-2

The TEXTSAVE procedure has no parameters.

TOLIBR Procedure

The TOLIBR procedure copies a disk, diskette, tape, or tape cartridge file that contains one or more library members into a library. You can also copy selected members from the file to the library.

All sector mode files to be copied by TOLIBR must have been created either by the FROMLIBR procedure or by the $MAINT utility program. If the sector-mode file was created by an IBM System/34, any System/34 load members are not copied. If the sector-mode file was created by an IBM System/32, any System/32 load members are not copied.

Each library member in a record mode file that is to be copied by TOLIBR must begin with a COPY statement and end with a CEND statement. Refer to the "COPY and CEND Statements" on page A-69 for the format of these statements. COPY and CEND statements are automatically inserted in members created by $MAINT. **You** must insert them at the beginning and end of members that were not created by the $MAINT utility program or by the FROMLIBR procedure. Do not insert more than the one required CEND statement, however. If a CEND statement exists within the member, an error message will be issued.

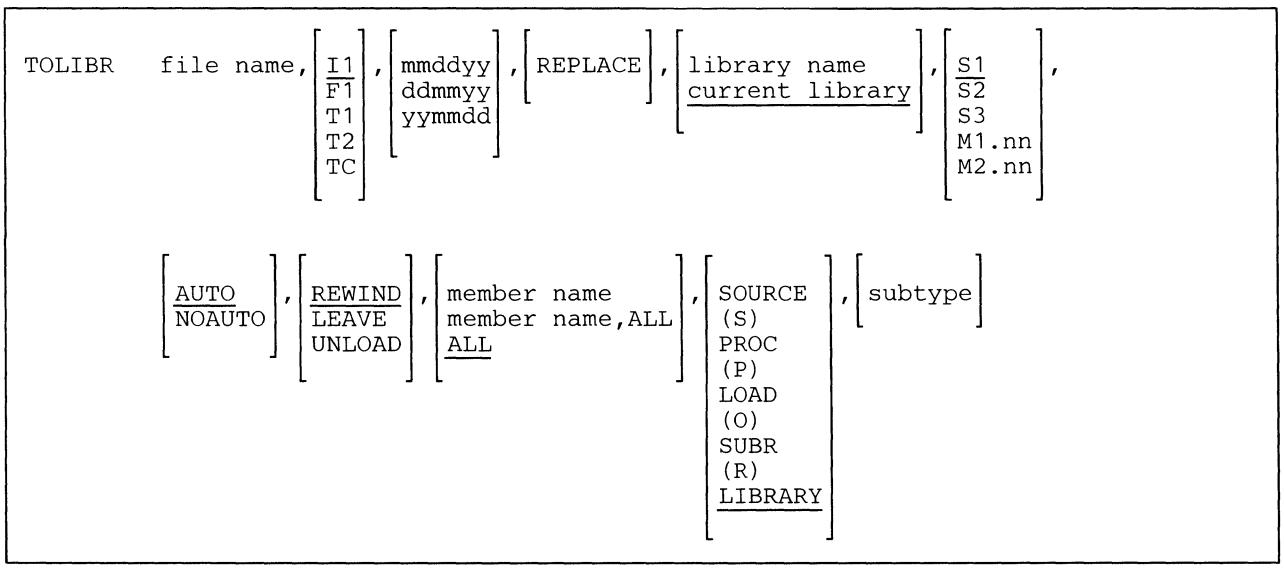
If the record mode file is organized as a direct disk file, you must insert an END statement following the last CEND statement in the file. The format of the END statement is:

```
// END
```

where only one blank must separate the // and the END.

Members in a record mode file with a subtype of TXT (text) will not be copied. If you attempt to copy a TXT member, a message will be issued allowing you to change the subtype to UNS (unspecified) in order to copy the member.

The TOLIBR procedure runs the $MAINT utility program.



S9020279-1

file name specifies the file containing the one or more library members to be copied into the library.

I1 specifies that the file is on diskette. If no parameter is specified, I1 is assumed.

F1 specifies that the file is on disk.

T1, T2, or TC specifies that the file is on tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. TC indicates that the tape is a tape cartridge.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file containing the members to be copied. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If more than one file exists with the specified file name, and if the date is not specified, the following applies:

- If I1 was specified or assumed, the first file with that name on the diskette is copied.
- If F1 was specified, the most recently created file with that name is copied.
- If T1, T2, or TC was specified, the first file with that name on tape is copied.

REPLACE specifies that if library members already exist with the specified library name, they are to be replaced. If REPLACE is specified, new members replace existing members with duplicate names, and no messages regarding the replacements are displayed.

If REPLACE is not specified, members are placed in the library until a duplicate is found, at which time the system displays a message telling the operator that a duplicate exists. In response to the message, the operator can either cancel the job or continue processing. If the job is continued, the new member replaces the existing member in the library. If other duplicates are found during the job, then existing members are automatically replaced and no messages are displayed regarding the duplicate members.

library name specifies the library that will contain the copied members. If a library name is not specified, the current library is assumed.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. This parameter is valid only if the I1 is specified. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01. This parameter is valid only if the I1 is specified.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only the specified tape drive will be used for all tape volumes.

- If TC is specified, the AUTO/NOAUTO parameter is ignored because there is only one tape cartridge drive.

REWIND specifies that the tape should be rewound after the TOLIBR procedure has run. REWIND is assumed if this parameter is not specified. This parameter is not valid if I1 or F1 is specified.

LEAVE specifies that the tape should be left where it is after the TOLIBR procedure has run. The next step within a procedure accessing a tape starts at this position if the same tape unit is specified. This parameter is not valid if I1 or F1 is specified.

UNLOAD specifies that the tape should be rewound and unloaded after the TOLIBR procedure has run. This parameter is not valid if I1 or F1 is specified. If UNLOAD and TC are specified, the tape will be rewound after processing.

member name specifies the library member to be copied.

member name,ALL specifies that the library members whose names begin with the specified characters (member name) are to be copied. Up to 7 characters can be specified for member name.

ALL specifies that all members of the library or within a certain subtype are to be copied. If no parameter is specified, ALL is assumed.

SOURCE or S specifies that only library source members are to be copied.

PROC or P specifies that only the library procedure members are to be copied.

LOAD or O specifies that only the library load members are to be copied.

SUBR or R specifies that only the library subroutine members are to be copied.

LIBRARY specifies that all library member types (SOURCE, PROC, LOAD, and SUBR) for the specified member name and subtype are to be copied. If no parameter is specified, LIBRARY is assumed.

subtype specifies that the members being copied are to be of the subtype specified. Valid subtypes are:

ARP RPG auto report member
ARS Automatic response member
ASM Assembler member
BAP BASIC procedure (source member)
BAS BASIC member
COB COBOL member
DFU Data file utility member
DTA Data member
FMT Display format member
FOR FORTRAN member
KEY KEYS procedure
MNU Menu member
MSG Message member
PHL Phone list member
QDE Query data entry member
QRY Query member
RPG RPG member
SRT Sort member
UNS Unspecified
WSU Work station utility member
X25 X.25 packet switching link control

Example 1

Copy the members from a diskette file called PAY into a library named MYLIB and replace any duplicate members without a warning message being displayed.

```
TOLIBR PAY,,,REPLACE,MYLIB
```

Example 2

Copy only source members from a diskette file named PAY into a library named MYLIB and replace any duplicate members without a warning message being displayed.

```
TOLIBR PAY,,,REPLACE,MYLIB,,,,,SOURCE
```

Example 3

Copy only the procedure member named XYZ from a diskette file named PAY into a library named MYLIB. Issue a message if a member already exists with that name.

```
TOLIBR PAY,,,,MYLIB,,,,XYZ,PROC
```

Example 4

Copy only COBOL members from a diskette file named PAY into a library named MYLIB. Issue a message if a member already exists with that name.

```
TOLIBR PAY,,,,MYLIB,,,,,,COB
```

TRANSFER Procedure

The TRANSFER procedure copies basic data exchange or I-exchange diskette files to a sequential or indexed disk file. It also creates basic data exchange or I-exchange diskette files from a sequential, direct, or indexed disk file. See "Basic Data Exchange Files" on page 4-548 and "I-Exchange Files" on page 4-548 for more information about these types of files.

To display or print a basic data exchange or I-exchange diskette file, see the "LISTFILE Procedure" on page 4-267.

The TRANSFER procedure allows you to do the following:

- Convert a sequential, indexed, or direct disk file to a basic data exchange or I-exchange diskette file.
- Convert a basic data exchange or I-exchange diskette file to a sequential or indexed disk file.
- Add a diskette file that is in the basic data exchange or I-exchange to an existing sequential disk file.
- Add a disk file to an existing basic data exchange or I-exchange diskette file.

Deleted records in a delete-capable file are not transferred to diskette. Also, if a delete-capable file is copied to diskette, it will no longer be delete-capable when it is transferred back to disk.

Disk files will not be extended when they are copied from diskette.

When a basic data exchange or I-exchange diskette file is transferred to a sequential or indexed disk file, records are placed in the disk file sequentially, using the record length of the diskette file.

When a basic exchange diskette file is created from a disk file, the record length of the diskette file is set to that of the disk file, to 128 (for diskette 1), or to 256 (for diskette 2D), whichever is smaller. When an I-exchange diskette file is created from a disk file, the record length of the diskette file is set to that of the disk file.

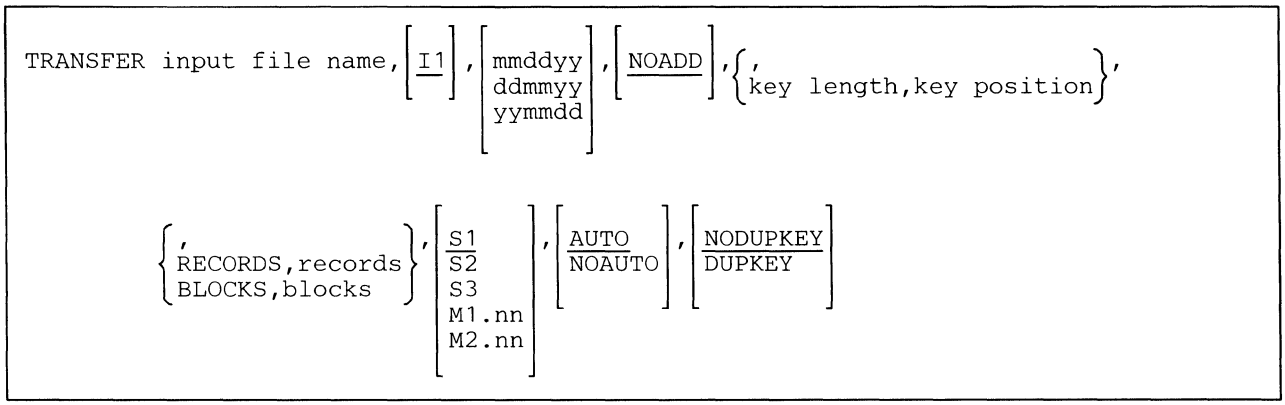
When a basic data exchange diskette file is added to an existing sequential disk file, the record length of the disk file is used for all records added to the file. This causes the records in the diskette file to be either truncated or padded with zeros (X'00') if their record length is different from the existing disk file.

When an I-exchange diskette file is added to an existing sequential disk file, the record length of the diskette file must equal the record length of the disk file.

When a disk file is added to an existing basic exchange or I-exchange diskette file, the record length of the diskette file must be the same as the record length of the disk file. However, when you are adding a disk file (whose record length is greater than the diskette sector size) to a basic data exchange file (whose record length is equal to the diskette sector size), the disk file records are truncated to the sector size. For the diskette 1, the sector size is 128 bytes; for the diskette 2D, the sector size is 256 bytes.

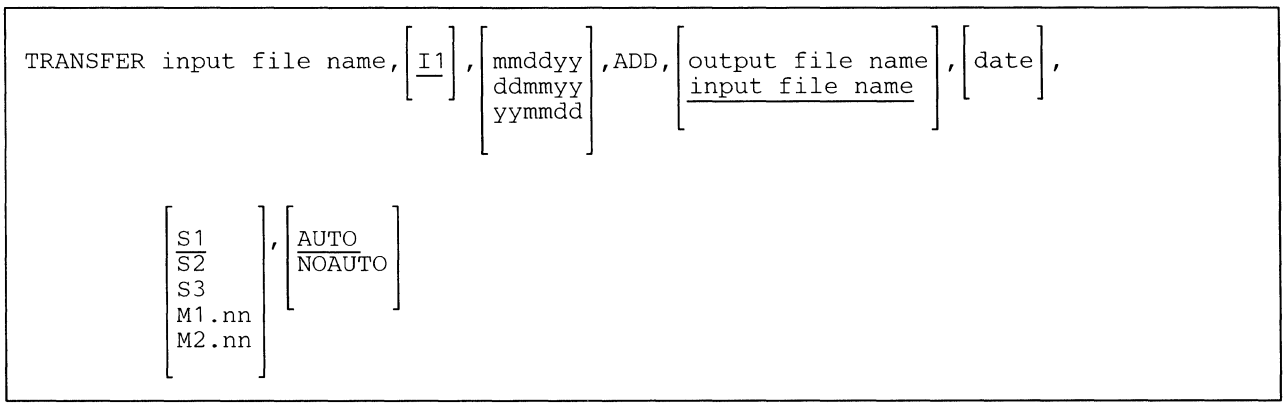
The TRANSFER procedure runs the $BICR utility program. The $BICR utility program processes records sequentially. If the input file is an indexed file and the output file is to be a basic data exchange diskette file, the records are read sequentially by key. If the output file is to be an I-exchange diskette file, the records are read sequentially by the physical order they exist in the file.

To copy a basic data exchange or I-exchange diskette file to a new disk file:



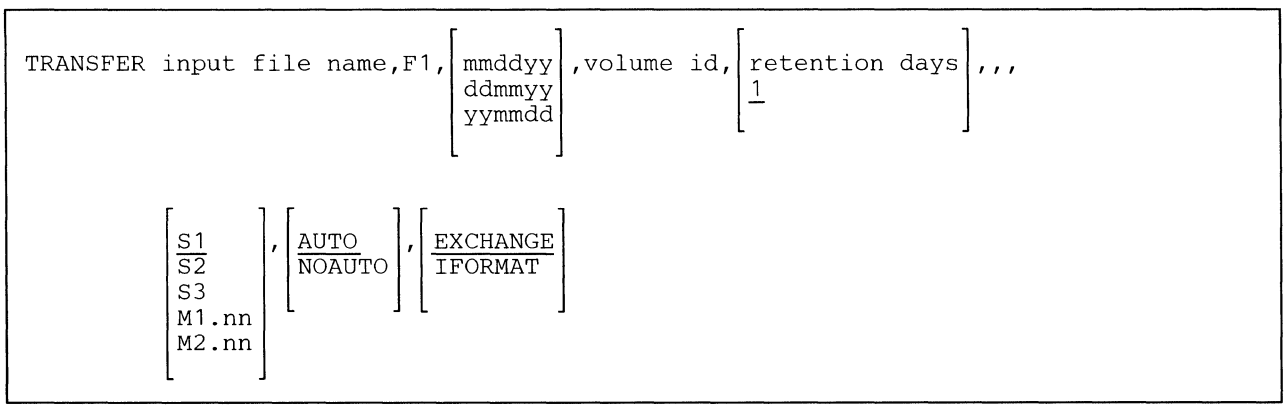
S9020280-0

To add a basic data exchange or I-exchange diskette file to an existing disk file:



S9020281-0

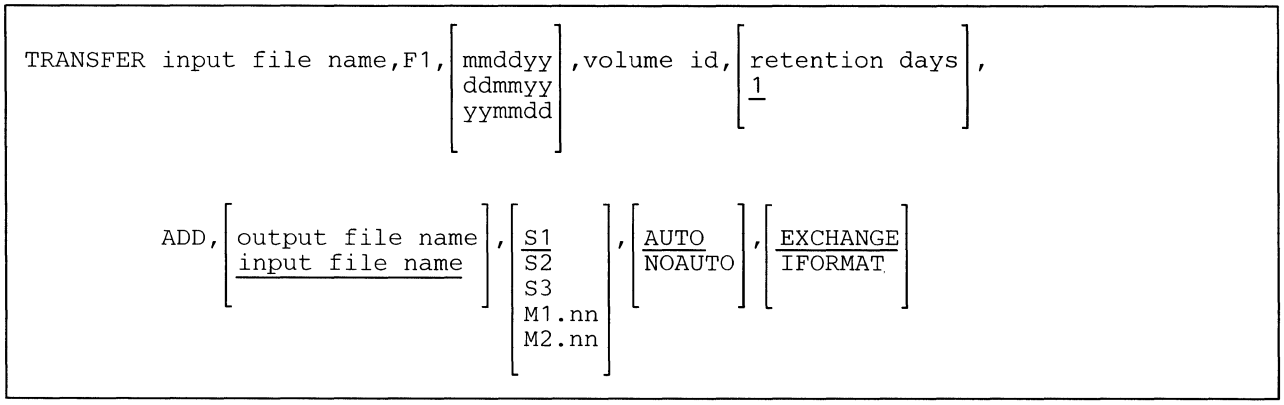
To copy a disk file to a basic data exchange or I-exchange diskette file:



S9020282-0

TRANSFER

To add a disk file to an existing basic data exchange or I-exchange diskette file:



S9020283-0

input file name specifies the file being transferred. If a new file is being created, it is given the specified file name.

I1 specifies that a basic data exchange or I-exchange diskette file is being transferred to a sequential or indexed disk file. If no parameter is specified, I1 is assumed.

F1 specifies that a disk file is being transferred to a basic data exchange or I-exchange diskette file.

mmddyy, ddmmyy, or yymmdd specifies the creation date of the file being transferred. The date, if specified, must be in the session date format; use the STATUS SESSION command to determine the session date format. If more than one file exists with the specified file name, and if the date is not specified, the following applies:

- If I1 is specified, the first file with that name is transferred.
- If F1 is specified, the most recently created file with that name is transferred.

NOADD specifies that the basic data exchange or I-exchange diskette file being transferred will become a new disk file. NOADD is assumed whenever a file is transferred from diskette to disk.

ADD specifies that the input file is to be added to the output file. (The first record in the input file is to immediately follow the last record in the output file.)

key length specifies the key length for the indexed disk file that is being created. The key length can be any number from 1 through 120. It must be specified with the key position, and the sum of the key length and the key position must not exceed the length of the record.

key position specifies the starting position of the key for the indexed disk file that is being created. For basic data exchange files, the key position can be any number from 1 through 128 for type 1 diskettes or from 1 through 256 for type 2D diskettes. For I-exchange files, the key location can be any number from 1 through 4096. The key location must be specified with the key length, and the sum of the key length and the key position must not exceed the length of the record.

DUPKEY specifies that duplicate keys are to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file will be the attribute of the output file.

NODUPKEY specifies that duplicate keys are not to be allowed in the indexed file being created. If the file being created is not an indexed file, this parameter is ignored. If this parameter is not specified, the attribute of the input file will be the attribute of the output file.

RECORDS specifies that the disk file being created must be large enough to contain the number of records specified.

records specifies the number of records for the file. Any number from 1 through 8000000 can be specified.

Either RECORDS or BLOCKS is required if:

- The file that is being transferred is on more than one diskette.
- The created disk file is to be larger than the file being transferred.

BLOCKS specifies that the disk file being created must be large enough to contain the number of blocks specified.

blocks specifies the number of blocks for the file. Any number from 1 through 312815 can be specified.

output file name specifies the existing disk file to which a basic data exchange or I-exchange diskette file is to be added, or the name of a basic data exchange or I-exchange diskette file to which records from a disk file are to be added. The output file name is allowed only if ADD is specified. If no output file name is specified, the input file name is assumed.

date specifies the creation date of the existing disk file. The date is allowed only if ADD is specified. The date must be specified in the same format as the session date. You can use the STATUS SESSION command to determine the format of the date.

volume id specifies the volume ID of the diskette. From 1 through 6 characters can be specified.

retention days specifies the length of the retention period in days for the created basic data exchange or I-exchange diskette file. Any number from 1 through 999 can be specified. If no retention period is specified, 1 day is assumed. If a retention period of 999 days is specified, the diskette file is a permanent file. For more information on diskette file retention, see the "FILE OCL Statement (for Diskette Files)" on page 5-43.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

EXCHANGE specifies that a disk file is being transferred to a basic data exchange diskette file. If a parameter is not specified, EXCHANGE is assumed.

IFORMAT specifies that a disk file is being transferred to an I-exchange diskette file.

Example 1

Create a disk sequential file named FILE2 from a diskette basic data exchange or I-exchange file named FILE2.

```
TRANSFER FILE2
```

Example 2

Create an indexed disk file named FILE2 from a basic data exchange file named FILE2. The key is to be in positions 1 through 5.

```
TRANSFER FILE2,,,,5,1
```

Example 3

Add a basic data exchange or I-exchange diskette file named FILE1 to an existing disk file named FILE1.

```
TRANSFER FILE1,,,ADD
```

Example 4

Create a basic data exchange diskette file named FILE3 on a diskette from a disk file named FILE3. The file is to be saved for 30 days. The volume ID of the diskette is VOL001.

```
TRANSFER FILE3,F1,,VOL001,30
```

Example 5

Create an I-exchange diskette file named FILE3 on a diskette from a disk file named FILE3. The file is to be saved for 30 days. The volume ID of the diskette is VOL002.

```
TRANSFER FILE3,F1,,VOL002,30,,,,,IFORMAT
```

Basic Data Exchange Files

When basic data exchange diskette files are copied or created, the diskettes being used must be initialized in one of the following formats using the INIT procedure's FORMAT parameter:

- Diskette 1 (one-sided) diskettes must be initialized in the 128-bytes per sector format.
- Diskette 2D (two-sided, double-density) diskettes must be initialized in the 256-bytes per sector format.

If the diskette format is not known, you can use the CATALOG procedure to list the diskette format.

These diskette files can be used to exchange data between systems and devices. See the *IBM Diskette General Information Manual*, GA21-9182, for information about the data set label fields.

The records in a basic data exchange file are not blocked and cannot span diskette sectors. That is, only one diskette sector (128 bytes for the diskette 1, 256 bytes for the diskette 2D) is used for each record. The data is truncated if the record length is greater than the diskette sector size.

I-Exchange Files

These diskette files can be used to exchange data between systems and devices that support the diskette exchange type I.

Note: Feature 6101 is needed to do a transfer with I-exchange files.

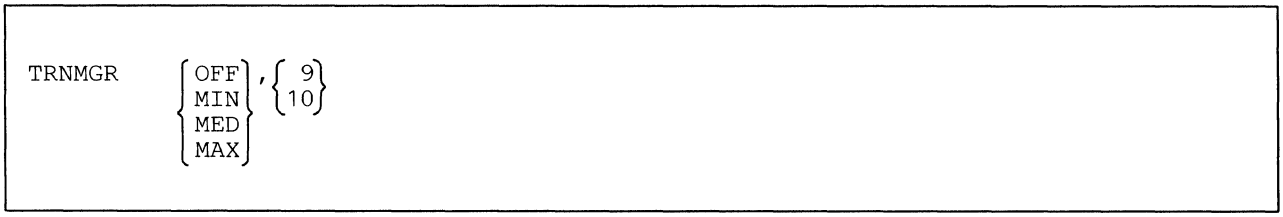
When I-exchange files are copied, the diskettes must be initialized in one of the following formats using the INIT procedure's FORMAT or FORMAT2 parameter:

- Diskette 1 (one-sided) diskettes can be initialized in either the 128- or 512-bytes per sector format.
- Diskette 2D (two-sided, double density) diskettes can be initialized in either the 256- or 1024-bytes per sector format.

The records in an I-exchange file are blocked and can span diskette sectors. That is, several records and parts of records can be placed in a diskette sector, or a record can extend from one sector to another. However, the records cannot span diskette volumes.

TRNMGR Procedure

The TRNMGR procedure allows you to start, stop, or change the error reporting level for a specified line in an IBM Token-Ring Network. It provides several reporting levels so that you can control the amount of information logged in the system history file.



OFF specifies that all error reporting for the specified line is to stop.

MIN specifies that minimum error reporting for the specified line is to start. At this level, only the conditions that indicate degraded performance are displayed.

MED specifies that medium error reporting for the specified line is to start. At this level, conditions that indicate degraded performance and potentially degraded performance are displayed.

MAX specifies that maximum error reporting for the specified line is to start. At this level, all error conditions are displayed.

9 specifies that error reporting is to occur for communications line 9.

10 specifies that error reporting is to occur for communications line 10.

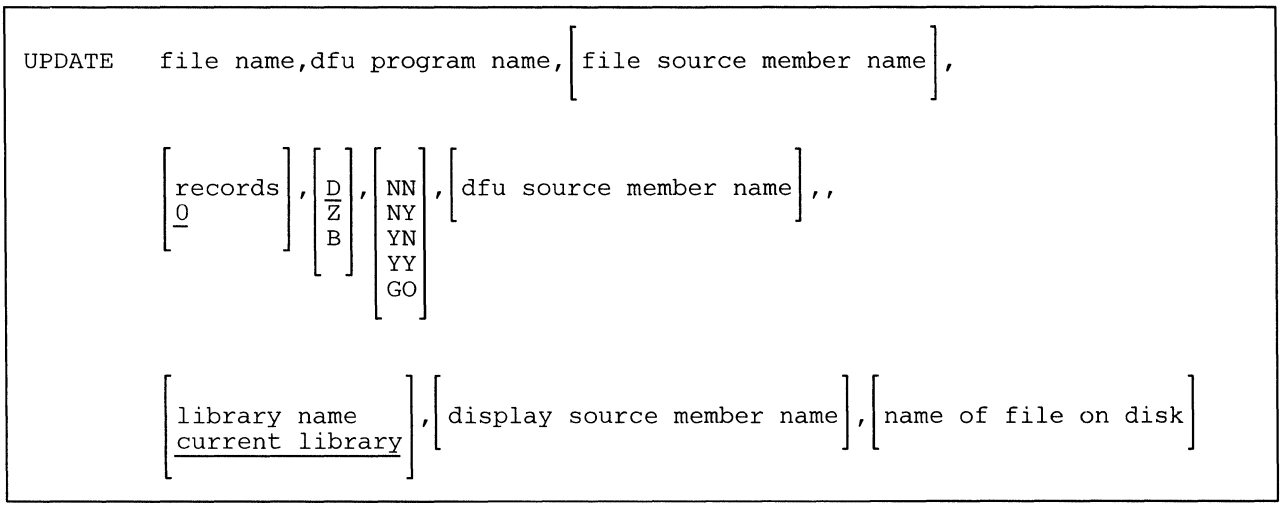
Example

This example shows how to start maximum error reporting for communications line 10.

```
TRNMGR MAX,10
```

UPDATE Procedure

The UPDATE procedure allows you to change a disk file using the data file utility (DFU). File specifications are used to define the format of the records in the file. For more information about DFU, see the *DFU Guide*.



S9020284-0

file name specifies the file to be changed. The file name can be from 1 through 8 characters.

dfu program name specifies the DFU program to be created to process the file. If the program does not exist in the library, DFU starts the setup procedures. If the program does exist in the library, DFU goes directly to changing the file.

file source member name specifies the source member containing the file description (F-specifications) and record input specifications (I-specifications) that describe the file to be processed. This member can contain one or more sets of file description and input specifications, or an entire RPG II program. The file description and input specifications that correspond to the file are taken as the data description.

This parameter is prompted for if it was not specified. It is required if the the specified DFU program does not exist.

records specifies the number of records the file is to be extended when it becomes full. If no number is specified, 0 (zero) is assumed and the file will not be extended.

D, Z, or B indicates whether unkeyed numeric fields are to be filled with zeros (hex F0) or blanks. The only allowed entries are **D**, **Z**, or **B**. If no parameter is specified, D is assumed.

D or B specifies a data file with blank fill of unkeyed numeric fields.

Z specifies a data file with zero fill of unkeyed numeric fields.

NN, NY, YN, YY, or GO specifies how the source specifications are to be processed. This 2-character parameter is appropriate if the DFU program does not exist and that job setup must be performed. It indicates whether the DFU specifications for this job are already stored in the library, and whether they are to be stored in the library when the job setup is complete. This source member of DFU specifications is stored or looked for in the current library, unless a library name is specified.

NN Stored DFU specifications are not used by the setup function. You are prompted to create the specifications, but they are not saved in the library.

NY Stored DFU specifications are not used by this job setup. You are prompted to create the specifications; once created, they are stored in the library.

YN Stored DFU specifications are used by the setup function. You can update the retrieved specifications before the DFU program is changed. If you change the specifications at the display station, only this job is affected; the changes are not stored in the library.

YY Stored DFU specifications are used by the setup function. You can update the retrieved specifications before the DFU program is changed. The specifications in the library are replaced by the updated specifications.

GO Stored DFU specifications are used by the setup function. You cannot update the specifications before the DFU program is changed unless errors are found. If errors are found in the specifications, you must correct the specification before the DFU program is changed. However, the stored DFU specifications are not changed.

dfu source member name specifies the source member that contains or will contain saved DFU specifications. This parameter is required if the DFU source processing parameter is not NN.

library name specifies the library that contains or will contain the DFU specifications. All library members associated with the DFU job are looked for, or stored, in this library. If this parameter is not specified, the current library is assumed.

display source member name specifies the source member in which DFU is to store the display format source specifications that were created when the DFU job was set up.

name of file on disk specifies the name in the disk VTOC of the file to be changed, if it is different from the name specified in the DFU program. If this parameter is specified, you can have several programs that refer to different, logical files change the same file actually disk. This is an optional parameter. If you specify the name of a file on disk and fail to specify a file to be changed by the DFU program, you are prompted for that parameter.

Example

This example shows how to change a disk file named FILE1. The DFU specifications named DFUSETUP are to be used to process the file. The library MYLIB contains DFU specifications.

```
UPDATE FILE1,FILEFMT,,,,GO,DFUSETUP,,MYLIB
```

WSFLOAD Procedure

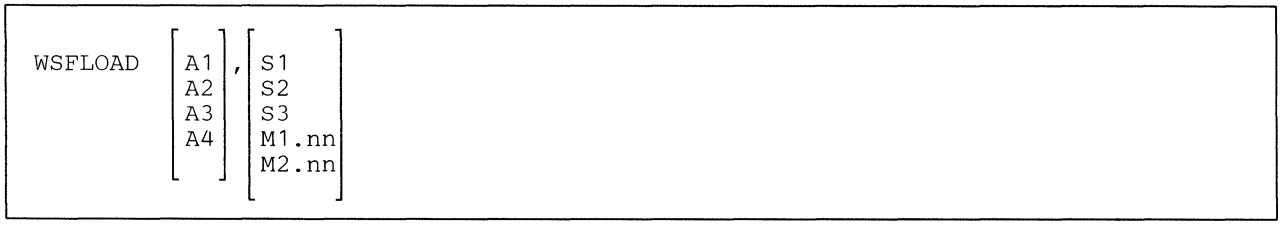
The WSFLOAD procedure creates a library named #IWLIB2 and copies the PC Support/36 work station feature from diskette into that library and the system library, #LIBRARY. One file, #IWPCLD4, is also copied from diskette.

The WSFLOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the WSFSAVE procedure. See the "WSFSAVE Procedure" on page 4-553 for information about how to save the PC Support/36 work station feature on diskette.

The WSFLOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the WSFLOAD procedure to restore support that has been saved by WSFSAVE.

If PC Support/36 is not currently on the system, you must use the TOLIBR procedure to copy the diskette file IWS into #LIBRARY before running WSFLOAD.

If #LIBRARY is backed up with PC Support/36 on the system and then replaced before WSFLOAD is run, you do not have to copy the diskette file IWS using the TOLIBR procedure.



S9020632-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and the file #IWPCLD4 and library #IWLIB2 are placed on the least-used disk unit. If no location is specified and the system has more than one disk unit, the files are placed on the least-used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01. Specifying M2 is the same as specifying M2.01.

Example

Create the file #IWPCLD4 and library #IWLIB2 on disk and copy the PC Support/36 work station feature from diskette.

```
WSFLOAD
```

WSFSAVE Procedure

The WSFSAVE procedure copies the PC Support/36 work station feature to diskette. The work station feature from the libraries #IWLIB2 and #LIBRARY and the file #IWPCLD4 is copied. You should use the "WSFLOAD Procedure" on page 4-552 to load the PC Support/36 work station feature from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPWSF and be located in diskette slot S1.

```
WSFSAVE
```

S9020633-0

The WSFSAVE procedure has no parameters.

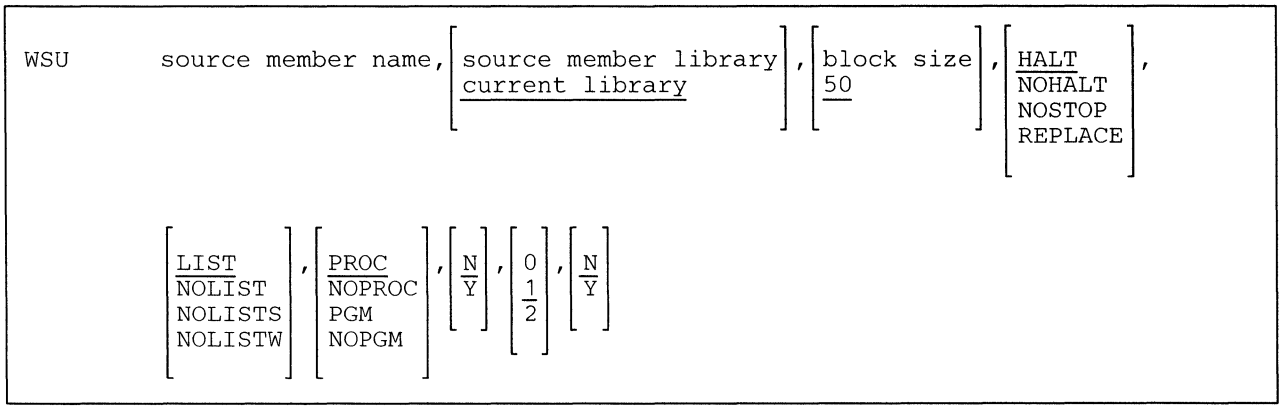
Example

Copy the PC Support/36 work station feature onto diskette.

```
WSFSAVE
```

WSU Procedure

The WSU procedure generates a work station utility (WSU) program from specifications in a library source member. WSU is an interactive data-entry utility that allows you to create and update transaction files and to update master files. For more information about WSU, see the *WSU Guide*.



S9020285-1

source member name specifies the library source member that contains the WSU source program specifications.

source member library specifies the library that contains the source member. If no library name is specified, the current library is assumed.

block size specifies the number of blocks to be used for a work file. Any number from 1 through 9999 can be entered. If no number is entered, 50 blocks are assumed.

HALT specifies that if any terminating errors are encountered in the program, an error message should be displayed. If no parameter is entered, HALT is assumed. If any library members are found with the same names as the library members generated by the WSU procedure, a displayed message allows you to either replace the existing member or cancel the WSU procedure.

NOHALT specifies that if any terminating errors are encountered in the program, no error message should be displayed, and the WSU procedure should continue processing. If any library members are found with the same names as the library members generated by the WSU procedure, a displayed message allows you to either replace the existing member or cancel the WSU procedure.

NOSTOP is a combination of the NOHALT and REPLACE functions. The WSU procedure continues processing if a terminating error is encountered, and any existing library members are automatically replaced with any newly generated members.

REPLACE specifies that any existing library members that have the same names as the generated WSU program, procedure, display format source member, or display format load member are automatically replaced by the newly created members.

LIST specifies that a complete WSU program generation listing is to be produced. This listing includes the WSU program generation information, WSU source program information, diagnostic information, and display format information. See the *WSU Guide* for more information about the information in the listing. If no parameter is specified, LIST is assumed.

NOLIST specifies that only the diagnostic information is to be listed.

NOLISTS specifies that the information included in the LIST parameter is to be listed, but the display format information is to be omitted from the listing.

NOLISTW specifies that only the diagnostic information and display format information is to be listed.

PROC specifies that the procedure that calls the generated WSU program is to be created (as well as the WSU program and display formats). If no parameter is specified, PROC is assumed.

NOPROC specifies that no procedure is to be created, but the WSU program and display formats are to be generated.

PGM specifies that only the WSU program is to be generated. A procedure and the display format source and load members are not to be created.

NOPGM specifies that no members are to be created or replaced. You can use this parameter to create the listings associated with the generation of the program, without actually creating the program or the other library members.

N specifies that the WSU program generation job should not be placed on the input job queue, but should be run from the display station requesting the WSU procedure. If no parameter is entered, N is assumed.

Y specifies that the WSU program generation job should be placed on the input job queue.

0, 1, or 2 specifies how much information is to be included in the WSU program generation listing. If no parameter is entered, 1 is assumed. See the *WSU Guide* for more information.

0 specifies that the following information is to be listed:

Information about the WSU program generation
Information about the WSU source program
Diagnostic information
Undefined indicators
Field names defined more than once
Undefined field names
Main storage requirements
Disk storage requirements
The procedure created for running the generated WSU program
Diagnostic text

1 specifies that the following information is to be listed:

All the information listed for parameter 0
The indicators used
Unreferenced indicators
Message member message identification codes used
Mode-level data field names used
Session-level data field names used
Job-level data field names used
Program label names used
Unreferenced field names

2 specifies that the following information is to be listed:

All the information listed for parameter 1
Indicator name cross-reference
Field name and label cross-reference

N specifies that the display format source member created by WSU is not saved. If no parameter is entered, N is assumed.

Y specifies that the display format source member created by WSU is to be saved in the same library as the generated program.

Example

This example shows how to generate a WSU program named PROG1, a full listing, and the procedure used to run the program. The source member is in the current library.

```
WSU PROG1,,,,,,,2
```

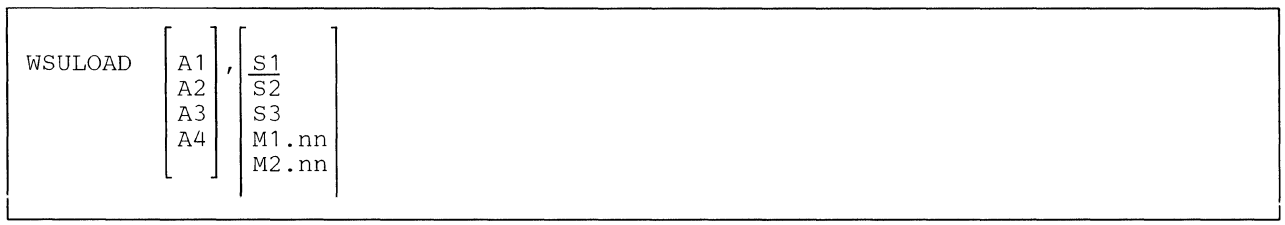
WSULOAD Procedure

The WSULOAD procedure creates a library named #WSULIB and copies the work station utility (WSU) support from diskette into that library. WSULOAD copies additional support into the system library (#LIBRARY). The WSULOAD procedure can copy either diskettes obtained through software distribution or diskettes created by the WSUSAVE procedure. See the "WSUSAVE Procedure" on page 4-558 for information about how to save the WSU support on diskette.

The WSULOAD procedure could change the master configuration record, with the result that there may not be a matching configuration member. To change the configuration member to match the master configuration record, see the manual *Changing Your System Configuration*. You should normally use the CNFIGSSP procedure to add support to the system. You should use the WSULOAD procedure to restore support that has been saved by WSUSAVE.

If WSU is not currently on the system, you must use the TOLIBR procedure to copy the diskette file WSU into #LIBRARY before running WSULOAD.

If #LIBRARY was backed up with WSU on the system and then replaced before WSULOAD is run, you do not have to copy the diskette file WSU using the TOLIBR procedure.



S9020286-0

A1, A2, A3, or A4 indicates the disk preference; that is, the disk unit that is searched first for available disk space. If not enough space is available on that unit, the other units (if they exist) are checked, and #WSULIB is placed on the least used disk unit. If no location is specified and the system has more than one disk unit, #WSULIB is placed on the least used disk unit.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If no diskette position is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a decimal value from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

Create #WSULIB on disk and copy the WSU support from diskette.

```
WSULOAD
```

WSUSAVE Procedure

The WSUSAVE procedure copies the work station utility (WSU) support to diskette. The WSU support from the libraries #WSULIB and #LIBRARY is copied. You should use the "WSULOAD Procedure" on page 4-557 to load the WSU support from the backup diskette. The diskette to contain the saved copy must have a volume ID of PPUTIL and be located in diskette slot S1.

```
WSUSAVE
```

S9020287-0

The WSUSAVE procedure has no parameters.

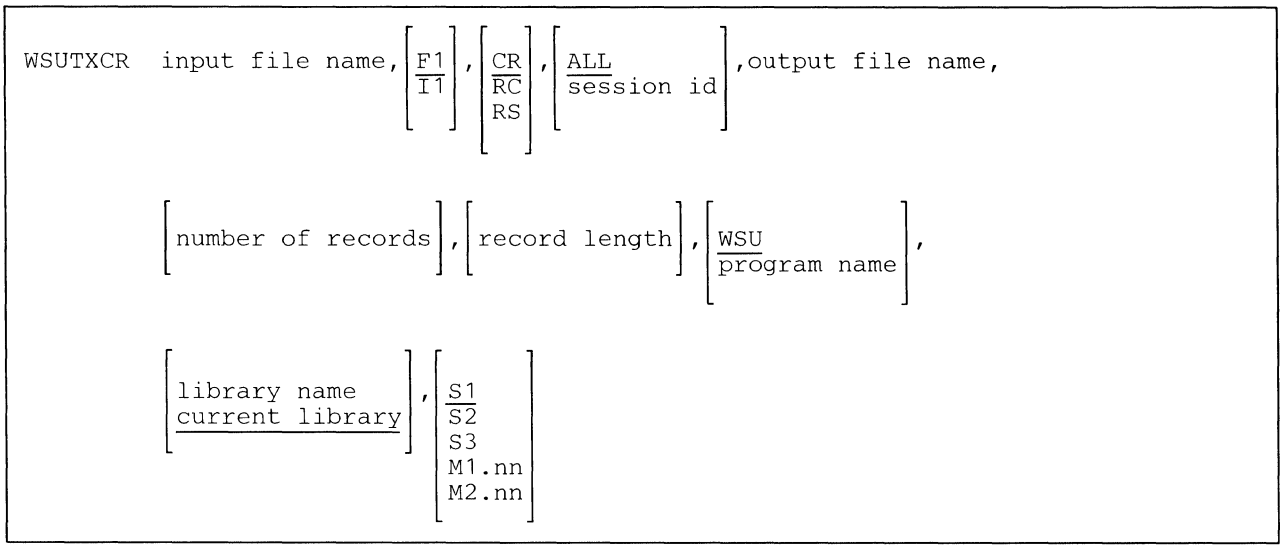
Example

Copy the WSU support to diskette.

```
WSUSAVE
```

WSUTXCR Procedure

The WSUTXCR procedure creates a work station utility (WSU) transaction file from a non-WSU sequential, indexed, or direct disk file. The procedure can also be used to recreate a WSU transaction file from another WSU transaction file. For more information about WSU and this procedure, see the *WSU Guide*.



S9020288-0

input file name specifies the input file.

F1 specifies that the input file is a disk file. If no parameter is entered, F1 is assumed.

I1 specifies that the input file is a diskette file. If the specified input file name matches the name of an existing disk file, a message is displayed.

CR specifies that a WSU transaction file is to be created from the non-WSU input file. If no parameter is entered, CR is assumed.

RC specifies that a WSU transaction file is to be recreated from the WSU input file.

RS specifies that a WSU transaction file is to be created from a WSU transaction file.

ALL specifies that the session ID in each input file record is to be copied unchanged to the output file. If no parameter is specified, ALL is assumed.

session id specifies the session ID that is to be placed in each output record.

output file name specifies the file that the WSUTXCR procedure is to create. If a file exists on disk with the specified name, an error message is displayed.

number of records specifies the number of records that are to be reserved for the output file. Any number from 1 through 8000000 can be specified. If the number of records is not specified, the number of records actually in the input file is used.

record length specifies the length of the records for the output file. If CR is specified, the record length should allow for the 13-byte trailer at the end of each record in the WSU transaction file. If no record length is entered, the record length of the input file as used.

WSU specifies that the generated WSU program is not yet available, and that all nonblank records in the output file are to be identified as detail records. If no parameter is assumed, WSU is assumed.

program name specifies the name of the generated WSU program that is to use the newly created or rebuilt transaction file.

library name specifies the library that contains the WSU program. If no parameter is entered, the current library is assumed.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

Example

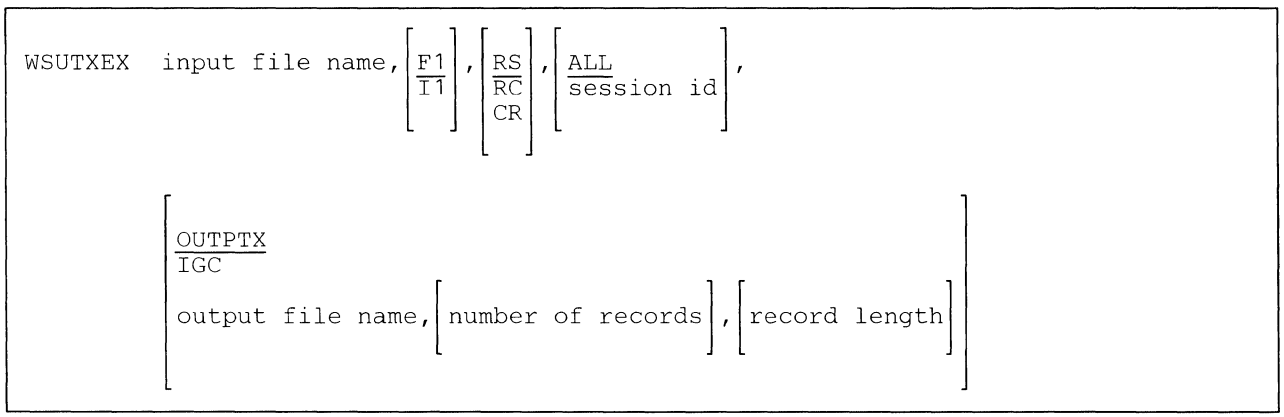
This example shows how to create a WSU transaction file named WSUTRAN from a sequential disk file named TRANSACT. A WSU program named WSUPROG, in the library PROGLIB, is to use the file. The file is to have 200 records and a record length of 300.

```
WSUTXCR TRANSACT,,CR,,WSUTRAN,200,300,WSUPROG,PROGLIB
```

WSUTXEX Procedure

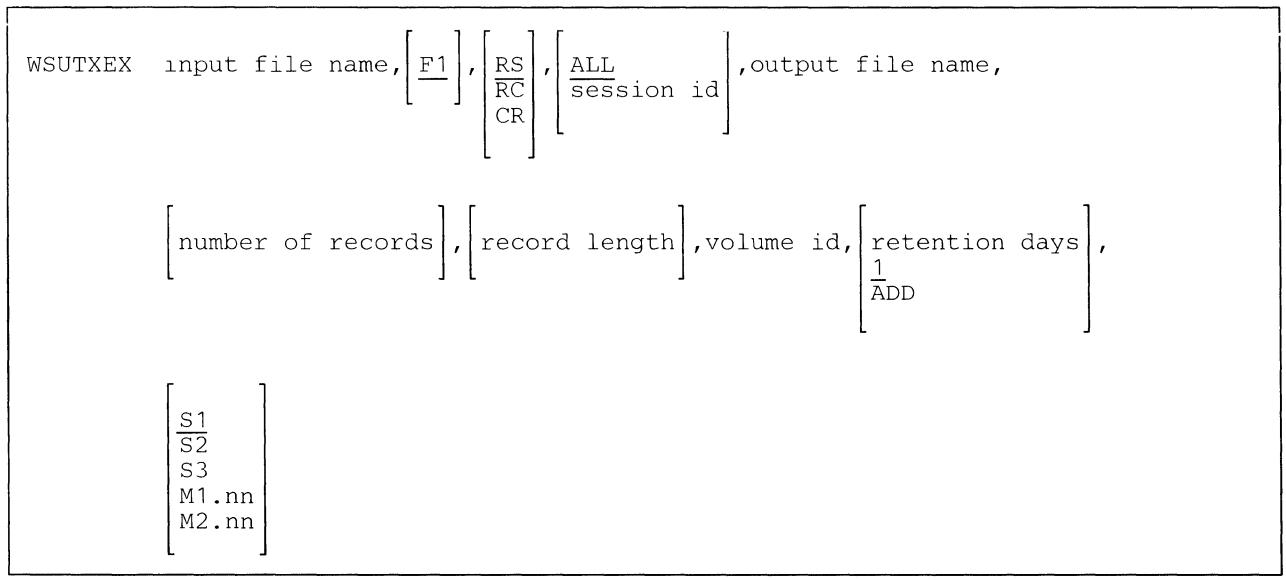
The WSUTXEX procedure copies records from a work station utility (WSU) transaction file or a non-WSU sequential, direct, or indexed file. The copied records can be listed, placed in a disk file, or placed in a diskette file. For more information about WSU and this procedure, see the *WSU Guide*.

To copy records from a file and list them or copy them to a disk file:



S9020289-0

To copy records from a file and add them or copy them to a diskette file:



S9020290-0

input file name specifies the name of the input file.

F1 specifies that the input file is a disk file. If no parameter is entered, F1 is assumed.

I1 specifies that the input file is a diskette file. If the specified input file name matches the name of a file on the disk, a message is displayed.

RS specifies that the header and detail records are to be copied in logical order from a WSU transaction file. Unchained (logically deleted) records are not to be copied. If no parameter is entered, RS is assumed.

RC specifies that the header and detail records are to be copied in physical order from a WSU transaction file or a non-WSU file.

CR specifies that all records that contain at least one nonblank position are to be copied in physical order from a non-WSU file.

ALL specifies that all header and detail records are to be copied.

session id specifies that only those header and detail records that have the specified session ID are to be copied.

OUTPTX specifies that copied records are to be listed on the current system list device. You can use the STATUS SESSION command to determine the current system list device. if no parameter is entered, OUTPTX is assumed.

IGC specifies that extracted records contain ideographic characters that are to be printed or displayed on the current system list device.

output file name specifies the file the WSUTXEX procedure is to create. If a file exists on disk with the specified name, an error message is displayed.

number of records specifies the number of records that are to be allocated for the output file. Any number from 1 through 8000000 can be specified. If the number of records is not specified, the number of records actually in the input file is used.

record length specifies the length of the records for the output file. If the record length is not specified, the record length of the input file is assumed.

volume id specifies the volume ID of a diskette to receive the copied records.

retention days specifies that a new diskette file is to be created and the number of days the file is to be retained. Any decimal number from 0 through 999 can be specified. If no retention period is entered, 1 day is assumed.

ADD specifies that an existing diskette file is to have the copied records added to it.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

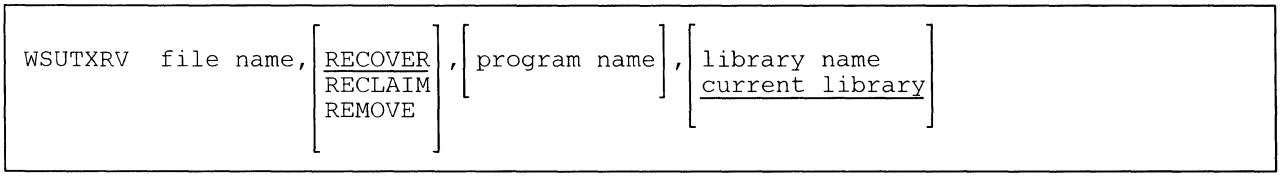
Example

This example shows how to copy records from a WSU transaction disk file named WSUTRAN and list those records on the system list device.

```
WSUTXEX WSUTRAN
```

WSUTXRV Procedure

The WSUTXRV procedure recovers a work station utility (WSU) transaction file that is causing problems when a WSU program is running. For more information about WSU and this procedure, see the *WSU Guide*.



S9020291-0

file name specifies the file to be recovered.

RECOVER specifies that control records are to be restored so there is no flag to indicate that records have been deleted.

RECLAIM specifies that partially inserted records are to be reclaimed; that is, operators will be able to use the records.

REMOVE specifies that partially inserted records and deleted records are to be removed from the file.

program name specifies the generated WSU program that is to be used to determine which records are correct or not correct.

library name specifies the library that contains the WSU program. If no parameter is entered, the current library is assumed. If a library name is specified, the program name must be specified.

Example

This example shows how to reclaim a WSU transaction disk file named WSUTRAN. The program that uses the file is named WSUPROG and is in the library named PROGLIB.

```
WSUTXRV WSUTRAN,RECLAIM,WSUPROG,PROGLIB
```

XREST Procedure

The IBM System/34 XREST procedure is not supported. See the "RESTEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-380 for information about restoring the extended character file. See the "SAVEEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-425 for information about saving the extended character file.

XSAVE Procedure

The IBM System/34 XSAVE procedure is not supported. See the "SAVEEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-425 for information about saving the extended character file. See the "RESTEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-380 for information about restoring the extended character file.

Chapter 5. OCL Statements

This chapter describes the System/36 operation control language (OCL) statements. The OCL statements allow you to run programs and control how the system runs programs. The following information is given for each statement:

- The function of the statement.
- The placement of the statement in relation to other statements in a procedure and the circumstances under which it is needed.
- The syntax format of the statement. For a description of the rules used to describe the syntax formats, see "Conventions Used for Describing Syntax Formats" on page 1-3.
- Descriptions of the statement's parameters.
- One or more examples of how to use the statement.

Placement of OCL Statements

OCL statements make up a job or job step. The end of a job step is indicated by a RUN OCL statement or (if the program being run reads utility control statements) an END utility control statement. Certain OCL statements can only be placed between the LOAD and RUN OCL statements for a job step; for example, the COMM OCL statement. Other OCL statements can be placed anywhere among the OCL statements.

The following example shows OCL statements that make up two job steps. The first job step (which runs the ACTREC program) ends with the first RUN OCL statement; the second job step (which runs the $COPY utility program) ends with the END utility control statement.

```
* First job step
// LOAD ACTREC
// FILE NAME-DATA
// RUN
*
* Second job step
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-DATA,DISP-SHR
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

Types Of Information Contained In OCL Statements

The OCL statements contain two types of information: an **identifier** and one or more **parameters**. The identifier distinguishes one OCL statement from another; a parameter supplies information to the SSP. The general form of an OCL statement is:

```
// identifier parameter1,parameter2, . . .
```

Identifiers

Every OCL statement except INCLUDE requires a statement identifier. All OCL statements begin with // followed by one or more blanks. For example, in the statement:

```
// LOAD PROGRAM
```

the OCL statement identifier is LOAD. The program to be loaded is named PROGRAM.

The end of data statement:

```
/*
```

does not require the //.

OCL Parameters

Parameters are either **positional** or **keyword** parameters. A positional parameter is a value. A keyword parameter is a keyword followed by a value. In the following statement, PROG1 and MYLIB are **positional parameters**, the name of a program to be loaded and the library that contains the program:

```
// LOAD PROG1,MYLIB
```

Positional parameters must be entered in the order shown in the syntax formats.

In the following statement, NAME-COPYIN and LABEL-FILE1 are **keyword parameters**:

```
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
```

A keyword parameter contains a **keyword** (NAME and LABEL are the keywords in the previous OCL statement) that distinguishes the parameter from other parameters, just as statement identifiers distinguish one OCL statement from another. In addition to a keyword, a keyword parameter usually contains a **value** (COPYIN and FILE1 are values in the previous OCL statement).

Keyword parameters can be entered in any order, not necessarily in the order shown in the syntax formats.

Procedure Parameters

A parameter can be any combination of characters. Question marks (?), commas (,), apostrophes ('), slashes (/), hyphens (-), greater than signs (>), equal signs (=), plus signs (+), and blanks should be used with caution because they have special meanings within procedures.

If a parameter is to contain embedded blanks or commas (,), it must begin with an apostrophe and end with an apostrophe. A parameter enclosed in apostrophes is considered to be a character string. All data between the apostrophes is processed as one parameter. The beginning and ending apostrophes are not processed as part of the parameter. The length of the parameter is set to equal the number of characters between the apostrophes. The following example shows the character string *June 17, 1985* as the first parameter for PROCA. The length of the parameter is 13.

```
PROCA 'June 17, 1985'
```

If an apostrophe is to be included within a character string, it must be specified as two apostrophes.

```
'one o''clock'
```

The length of the character string is set to equal the number of characters between the beginning and ending apostrophes, except two apostrophes within the string are counted as one. The following example shows how you would enter the character string *It's one o'clock*. The length of the character string is 16.

```
'It''s one o''clock'
```

If a parameter does not begin with an apostrophe, any other apostrophes encountered in the parameter are considered to be data, not character strings. The following example shows parameters with apostrophes as data. The first parameter is *Mary's* and the second parameter is *o'clock*.

```
PROCA Mary's,o'clock
```

General OCL Coding Rules

See "Conventions Used for Describing Syntax Formats" on page 1-3 for a description of how to enter procedures, commands, and OCL statements. Additional rules for OCL statements are:

- The // must be entered in positions 1 and 2.
- One or more positions between the // and the statement identifier must be blank. For example:

  ```
  // LOAD
  ```

  or:

  ```
  //   LOAD
  ```

- One or more positions between the statement identifier and the first parameter must be blank. For example:

  ```
  // LOAD PROG1,MYLIB
  ```

  or:

  ```
  //  LOAD   PROG1,MYLIB
  ```

- If you need to include more than one parameter, use a comma to separate the parameters. No blanks are allowed within or between parameters. Anything following the first blank after a parameter is considered to be a comment (see "Comments" on page 5-5).
- If you are entering keyword parameters, place the keyword first and use a hyphen (-) to separate the keyword from the value.
- If a value is not specified with a keyword parameter, the parameter is ignored. For example:

  ```
  // FILE NAME-FILEA,DATE-,UNIT-F1
  ```

  When this statement is processed, the DATE keyword parameter is ignored.

Continuing OCL Statements

Expressing a single statement in two or more lines is called **continuation**. Any OCL statement that contains keyword parameters can be continued.

Rules for using continuation are:

- Begin each new line with // in positions 1 and 2.
- Leave one or more blanks between the // and the first parameter in the line.
- Place a comma after the last parameter in every line except the last line. The comma, followed by a blank, tells the SSP that the statement is continued on the next line.

In the following example of a continued FILE OCL statement, five lines are used to express the statement.

```
// FILE NAME-INPUT,
//      LABEL-FILE1,
//      RECORDS-250,
//      RETAIN-J
```

The line is interpreted by the SSP as:

```
// FILE NAME-INPUT,LABEL-FILE1,RECORDS-250,RETAIN-J
```

See "Continuing the Lines of a Procedure" on page 2-7 for another way to continue OCL statements. This method uses a plus sign (+) in the last nonblank position in the line to indicate the line is continued. For example, the above example could also have been entered as the following:

```
// FILE NAME-INPUT,+
   LABEL-FILE1,+
   RECORDS-250,+
   RETAIN-J
```

And would be interpreted as:

```
// FILE NAME-INPUT,LABEL-FILE1,RECORDS-250,RETAIN-J
```

Note that it takes the system longer to read statements that are continued.

If a record ends with a shift-in character just before the continuation expression, and the first nonblank character of the next record is a shift-out character, both the shift-in and shift-out characters will be removed.

Comments

Comments can be used to explain the purpose of the statements in a procedure. Comments in a procedure do appear when the procedure is listed on a printer or display station. Comments do not appear when the procedure runs. Comments can contain any combination of characters. However, the continuation symbol (+ sign) is recognized in a comment if the statement does not have an * in column 1. Comments can be included in the following places:

- Following the * on the comment statement. For example:

  ```
  * THIS IS AN EXAMPLE OF A COMMENT STATEMENT
  ```

  The comment here is `THIS IS AN EXAMPLE OF A COMMENT STATEMENT.`

- After the last parameter in a statement. Leave one or more blanks between the last parameter and your comment. For example:

  ```
  // LOAD $COPY    LOAD THE COPY UTILITY
  ```

  In this example, the comment is `LOAD THE COPY UTILITY.` Another example:

  ```
  // RUN    RUN THE PROGRAM
  ```

  The comment here is `RUN THE PROGRAM`; the RUN statement has no parameters.

- After the comma that follows the last parameter in a line that is continued. For example:

```
// FILE NAME-INPUT,      FILE STATEMENT FOR
//      LABEL-FILE1,     INPUT FILE
//      RECORDS-250,
//      RETAIN-J
```

  In this example, the first two lines of the FILE statement contain comments.

- After the procedure name in an INCLUDE OCL statement if the statement has parameters but none of them are specified. Leave a blank after the procedure name, enter a comma, leave a blank after the comma, and enter the comment. For example:

```
PAYROLL , RUN PAYROLL PROCEDURE
```

  or:

```
// PAYROLL , RUN PAYROLL PROCEDURE
```

  or:

```
// INCLUDE PAYROLL , RUN PAYROLL PROCEDURE
```

  The comment in these statements is `RUN PAYROLL PROCEDURE.`

Notes:

1. *Substitution expressions contained in a comment statement will be evaluated, except comment statements preceded by an asterisk (*).*

2. *An INCLUDE OCL statement that calls either a multiple requester terminal (MRT) procedure or a procedure that passes data, not parameters, cannot contain a comment.*

ABEND OCL Statement

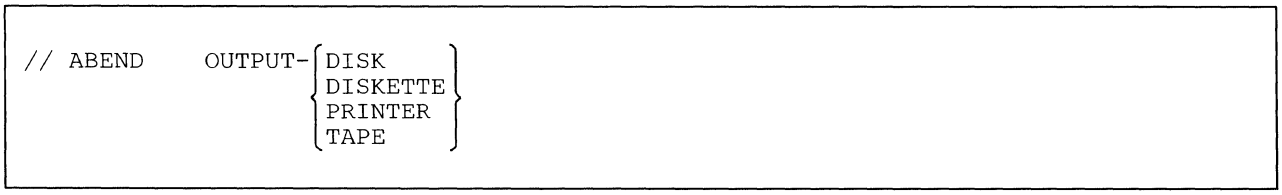
The ABEND OCL statement specifies the device to be used for a task dump in case a program check occurs. A program check is usually caused by an error in the program; the SSP stops the program and issues an error message. A task dump causes all the program's storage areas to be printed or copied to a disk or diskette file. The disk or diskette file can then be processed by the DUMP or APAR procedure.

The output device specified by the ABEND statement remains in effect until the job ends or until another ABEND statement is processed.

If no ABEND OCL statement is specified, the output is copied to a disk file.

The ABEND OCL statement is ignored for both the SETDUMP procedure dump and the dumps produced when the MSP Stop key on the system control panel is pressed (the Alter/Display function).

Placement: The ABEND OCL statement can be placed anywhere among the OCL statements.



S9020292-0

OUTPUT specifies where the task dump data is to be sent.

DISK specifies that a new disk file is to be created, and all the storage areas of the failing task are to be copied into the disk file. The file can then be processed by the DUMP or APAR procedure. Dump files are named #DUMP.nn on disk, where nn is a number from 00 through 99.

DISKETTE specifies that a new diskette file is to be created. The operator is prompted to insert a diskette into the diskette drive, and all the storage areas of the failing task are to be copied into the diskette file. The file can then be processed by the DUMP or APAR procedure.

PRINTER specifies that all storage areas of the failing task are to be printed on the operator's session printer. You can use the STATUS SESSION command to determine the session printer.

TAPE specifies that a new tape file is to be created. The operator is prompted to specify a tape drive for the tape to be mounted on, and all the storage areas of the failing task are to be copied into the tape file. The file can then be processed by the APAR, DFA, or DUMP procedure.

Example

To make the storage areas of the program PAYROLL be written to diskette (in case a program check occurs), you could use the following:

```
// ABEND OUTPUT-DISKETTE
// LOAD PAYROLL
// FILE NAME-EMPLOYES
// RUN
```

ALLOCATE OCL Statement

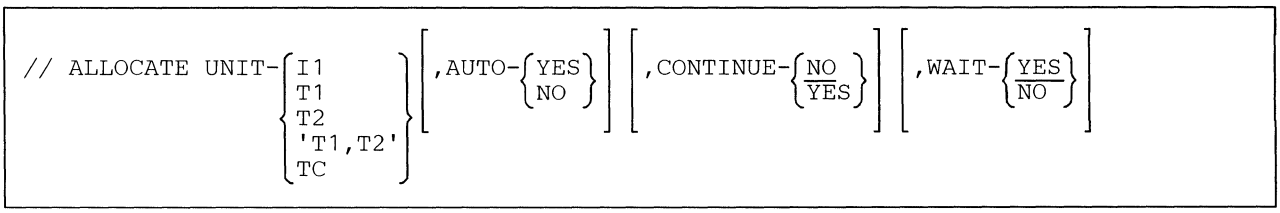
The ALLOCATE OCL statement causes the diskette drive or tape drives to be allocated to a job; that is, after the ALLOCATE statement is processed, no other procedure can use the drive. This allows your procedure to retain control of the drive during a job that uses the drive several times, without losing control to another job on the system. After the ALLOCATE OCL statement is processed and the diskette drive is allocated to your job, the Diskette in Use light is turned on.

This statement also allows you to write or read a series of diskette or tape files such that each job step that uses the diskette drive or tape drives will begin from the position used by a previous job step.

The drive remains allocated to the job until the job ends, or until a DEALLOC OCL statement is processed in your job.

Note: The ALLOCATE OCL statement is not allowed in a multiple requester terminal (MRT) procedure.

Placement: The ALLOCATE OCL statement can be placed anywhere among the OCL statements.



S9020293-1

UNIT specifies whether the diskette drive, a specific tape drive, or both tape drives are being allocated.

I1 specifies that the diskette drive is being allocated.

T1 specifies that tape drive 1 is being allocated.

T2 specifies that tape drive 2 is being allocated.

'T1,T2' specifies that both tape drive 1 and tape drive 2 are being allocated. If one of the tape drives is not available, neither tape drive is allocated. This option must be enclosed within single quotes and separated by a comma.

TC specifies that the tape cartridge is being allocated.

AUTO specifies whether the diskette drive should automatically advance to the next slot or the tape drive should automatically advance from one drive to another. This parameter overrides the AUTO and NOAUTO parameters of procedures and the AUTO parameter of the FILE OCL statement.

Examples of procedures that use FILE OCL statements are FROMLIBR and TOLIBR; examples of procedures that do not use FILE OCL statements are CATALOG and INIT. See the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48 for a list of the procedures and system utilities that use the FILE OCL statement for diskette files and tape files.

The AUTO parameter is ignored for systems that do not have a diskette magazine drive or multiple tape drives.

YES for diskette magazine drives, specifies that the next diskette slot should be advanced to automatically when another diskette needs to be processed. For tape drives, YES specifies that if the end of the tape is reached while processing a file, the next tape drive should automatically be used. If that tape drive cannot be allocated, the original tape drive is used for the next tape.

NO for diskette magazine drives, specifies that the next diskette slot should not be advanced to automatically when another diskette needs to be processed. For tape drives, NO specifies that the next tape drive should not be advanced to automatically when another tape needs to be processed.

CONTINUE specifies whether the job steps that use the diskette drive will process a sequential series of diskette files; that is, whether each job step that uses the diskette drive will continue from the ending diskette position of the previous job step. If the ALLOCATE statement is used but the CONTINUE parameter is omitted, CONTINUE-NO is assumed. The CONTINUE parameter is ignored if the system does not have a diskette magazine drive.

NO specifies that the diskette position specified in the procedure or on the diskette FILE OCL statement is to be used.

YES specifies that a sequential series of diskette files are to be created or read.

This method of operation remains in effect for as long as the diskette drive is used for either input or output, **and** the same location is specified in each procedure, diskette FILE OCL statement, or diskette-processing procedure control expression (VOLID-, ?VOLID?, or DATAI1) that uses the drive. Whenever one of these conditions changes, the SSP assumes a new series of diskette files is being processed and the position specified on that procedure or diskette FILE OCL statement should be used.

When diskette files are being created, only the first diskette can contain active files. When diskette files are being read, and a file cannot be found in the current position, the SSP assumes a new series of diskette files is being processed and the position specified on that procedure or diskette FILE OCL statement should be used.

If CONTINUE-YES is specified, AUTO-YES must also be specified.

Note: CONTINUE does not apply if a tape drive is allocated; an error message is issued.

WAIT specifies whether the procedure should wait for the drive to become available. If the ALLOCATE statement is used but the WAIT parameter is omitted, WAIT-YES is assumed.

YES specifies that if the drive is already allocated or is being used, the procedure should wait for the drive to become available. If the drive is allocated to or being used by an interrupted or waiting program, a program with the never-ending program attribute, or a job that has allocated the drive using the ALLOCATE OCL statement, a message will be displayed and the operator can either retry the allocation operation or cancel the job.

NO specifies that if the drive is already allocated or is being used, the procedure should continue. The ?CD? substitution expression can then be used to see whether the drive was allocated to the procedure. See "?CD? (Return Code) Expression" on page 3-16 for more information about the values substituted by this expression.

Example 1

To create a series of diskette files containing libraries you could use the following. The diskettes to contain the libraries are in diskette slots S1 and S2 and have a volume IDs of VOL001.

```
// ALLOCATE UNIT-I1,CONTINUE-YES,AUTO-YES
SAVELIBR MYLIB,,VOL001
SAVELIBR YOURLIB,,VOL001
SAVELIBR OURLIB,,VOL001
SAVELIBR PAYLIB,,VOL001
```

or:

```
// ALLOCATE UNIT-I1,CONTINUE-YES,AUTO-YES
// LOAD $MAINT
// FILE NAME-MYLIB,UNIT-I1,PACK-VOL001
// FILE NAME-YOURLIB,UNIT-I1,PACK-VOL001
// FILE NAME-OURLIB,UNIT-I1,PACK-VOL001
// FILE NAME-PAYLIB,UNIT-I1,PACK-VOL001
// RUN
// COPYLIBR FROM-MYLIB,TO-DISK,FILE-MYLIB
// COPYLIBR FROM-YOURLIB,TO-DISK,FILE-YOURLIB
// COPYLIBR FROM-OURLIB,TO-DISK,FILE-OURLIB
// COPYLIBR FROM-PAYLIB,TO-DISK,FILE-PAYLIB
// END
```

Example 2

To allocate the diskette drive to a job and not wait for the drive to become available you could use the following. The diskettes to contain the files are in diskette magazine slot M1 and have a volume ID of VOL002.

```
// ALLOCATE UNIT-I1,CONTINUE-YES,WAIT-NO,AUTO-YES
// IFF ?CD?=2032 IFF ?CD?=2033 GOTO OK
// PAUSE 'Diskette drive allocated to another job.'
// CANCEL
// TAG OK
SAVE ALL,,,VOL002,ALL,M1
SAVELIBR LIBR1,,VOL002,M1
SAVELIBR LIBR2,,VOL002,M1
```

Example 3

To allocate tape drive 1 to a job and not wait for the drive to become available, you could use the following. The tape to contain the files is on tape drive 1 and has a volume ID of VOL002.

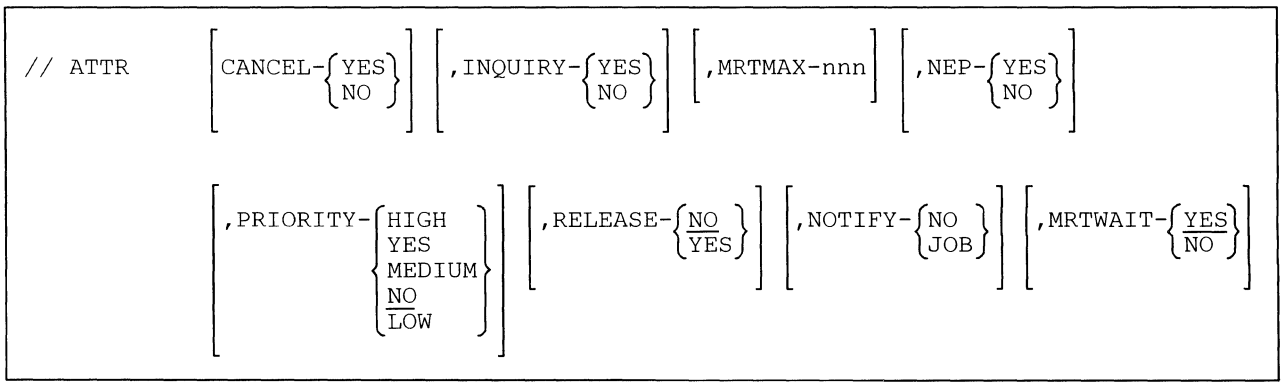
```
// ALLOCATE UNIT-T1,WAIT-NO,AUTO-YES
// IFF ?CD?=2035 IFF ?CD?=2036 GOTO OK
// PAUSE 'Tape drive 1 allocated to another job.'
// CANCEL
// TAG OK
SAVE ALL,,,VOL002,ALL,T1
SAVELIBR LIBR1,,VOL002,,,T1
SAVELIBR LIBR2,,VOL002,,,T1
```

ATTR OCL Statement

The ATTR OCL statement:

- Indicates whether an operator can cancel a job; that is, whether an operator can end a job before the job has completed normally.
- Indicates whether another job can be started at a display station while a job is interrupted.
- Changes the maximum number of requester terminals for a multiple requester terminal (MRT) program.
- Changes the never-ending program indicator for a program.
- Assigns a priority to a job or job step.
- Releases the requester display station from the next job step when the next job step begins running.
- Informs you if job has ended normally or abnormally.

Placement: The ATTR OCL statement can appear anywhere among the OCL statements except between the LOAD and RUN OCL statements.



S9020294-3

CANCEL specifies whether a job can be canceled by an operator pressing the Attn key and taking either option 2 or 3 from the inquiry display. This parameter only affects the options that are displayed on the inquiry display; that is, the operator can press the Attn key and temporarily interrupt the job, but options 2 or 3 to cancel the job are not available. The job can still be canceled by using the CANCEL control command at the system console.

This condition remains in effect until changed by another ATTR statement, or until the job ends.

YES specifies that options 2 and 3 of the inquiry display will appear, and that the operator can cancel the job by taking either option 2 or 3.

NO specifies that options 2 and 3 of the inquiry display will not appear, and that the job cannot be canceled using the Inquiry display.

INQUIRY specifies whether another job can be started by an operator pressing the Attn key and taking option 1 from the inquiry display or whether an alternate job can be started by pressing the Attn key and taking option 7 from the display. This parameter only affects the options that are displayed on the inquiry display; that is, the operator can press the Attn key and temporarily interrupt the job, but option 1 to request the command display and option 7 to resume an alternate interrupted job are not available. This condition remains in effect until changed by another ATTR statement, or until the job ends.

If the INQUIRY parameter is specified, the program's noninquirable indicator is ignored (this indicator can be set on by the overlay linkage editor or by a programming language such as RPG).

YES specifies that option 1 and option 7 of the inquiry display may appear. The operator can start another job by taking option 1 and entering the command or menu number required to start another job or take option 7 to resume an alternate interrupted job.

NO specifies that option 1 and option 7 of the inquiry display will not appear, thus preventing the operator from starting another job or resuming an alternate interrupted job.

MRTMAX specifies the number of multiple requester terminals (MRTs, display stations or SSP-ICF sessions) that can be attached to the program run in the next job step. Only one ATTR statement in a job step can specify the MRTMAX parameter.

nnn changes the MRTMAX value specified on the COMPILE statement when the program was compiled. nnn cannot exceed the MRTMAX value specified on the COMPILE statement or in the RPG II file description specification. MRTMAX is allowed only if an MRTMAX value of one or more was specified on the COMPILE statement. Leading zeros need not be entered.

NEP specifies whether the program is a never-ending program (NEP). NEP is defined as a long-running program. Other jobs cannot use the system resources, except for shared files and the spool file, that are allocated to an NEP.

YES specifies that the program is an NEP.

NO specifies that the program is not an NEP.

You should specify NEP-YES for a program that uses one or more system resources that cannot be shared (for example, a disk file that cannot be shared) for a longer time than a display station operator should have to wait for the SSP to start a program. If an operator attempts to run a program that uses a nonsharable system resource being used by an NEP, the SSP does not wait for the resource to become available but issues a message indicating that the requested program cannot be started because an NEP is using the required resource. The operator can then cancel the job or retry allocating the resource.

If an operator attempts to run a job that uses a nonsharable system resource being used by a long-running program for which NEP-YES was not specified, the SSP waits for the resource to become available. While the SSP is waiting, the display station operator may use Inquiry mode (by pressing the Attn key) to cancel the job or to run other jobs.

If the NEP parameter is specified, the NEP value specified when the program was compiled is ignored. Only one ATTR statement in a job step can specify the NEP parameter.

The following special considerations exist for jobs run from the job queue and for multiple requester terminal (MRT) programs:

- All programs run from the job queue are run as NEPs, unless NEP-NO is specified on the ATTR OCL statement.

- For an MRT program, if NEP-YES is not specified on the ATTR OCL statement and if NEP was not specified when the program was compiled, other programs will not wait for nonshared resources being used by the MRT program. However, when the MRT program releases its last requesting display station, the program goes to end-of-job processing.

- If NEP-YES is specified for an MRT program, other programs will not wait for nonshared resources being used by the MRT program. The MRT program will not end when it releases its last requesting display station. Instead, the MRT program will wait until it is requested by another display station. Normally, an MRT program with NEP-YES specified will not end until after the system operator enters the STOP SYSTEM control command and all display stations using the MRT program are released. (The system operator can end the MRT program at any time by using the CANCEL command.)

PRIORITY specifies the processing priority for a job or job step. The system assigns system resources in order of decreasing priority. The order of priority is: HIGH or YES, MEDIUM, NO, and LOW. For example, if PRIORITY-MEDIUM is specified, system resources are assigned to the job or job step after they are assigned to any higher priority (HIGH or YES) job or job step, but before they are assigned to any lower priority (NO or LOW) job or job step. PRIORITY-YES and PRIORITY-HIGH are equivalent. The PRIORITY parameter can be specified more than once in a job; this parameter takes effect as soon as it is encountered.

If an operator enters the PRTY command with HIGH, MEDIUM, LOW, YES, or NO specified before submitting a job, any PRIORITY parameters on the ATTR statement are ignored. If the system operator enters the PRTY command with HIGH, MEDIUM, LOW, YES, or NO specified after the job has started, any following PRIORITY parameters on the ATTR statement are ignored.

NO priority jobs can have their processing priority automatically changed by the system. HIGH, MEDIUM, and LOW priority jobs do not automatically change priority. See the *Concepts and Programmer's Guide* for more information about job priority.

RELEASE specifies whether the display station remains allocated to the next job step. The RELEASE parameter is ignored for jobs on the job queue. If RELEASE is not specified, RELEASE-NO is assumed.

NO specifies that the next job step is not released and that the display station remains allocated to the job step.

YES releases the display station when the next job step begins running. If that job step is the last or only step in a procedure, the command display appears at the display station and the operator can start another job. If that job step is not the last step in a procedure, only that job step is released. The released step runs at the same time as following steps of the same procedure. The requesting display station remains allocated to the steps that follow the released step. If RELEASE-YES is specified, the following points should be considered:

- Existing job files cannot be passed to the released step. For information about job files, see the "FILE OCL Statement (for Disk Files)" on page 5-32.
- Continued print files cannot be passed to the released step. For more information about continued print files, see the "PRINTER OCL Statement" on page 5-83.
- Only those files referenced by FILE OCL statements between the following LOAD and RUN OCL statement pair will be passed to the released job step. All other FILE OCL statements, including FILE OCL statements that have JOB-YES specified, will continue to be owned by the original job and can be used in following job steps. This means that a job step released by an ATTR OCL statement cannot own files that are specified outside of that job step's LOAD and RUN OCL statements.
- Scratch files are passed to the released job step unless a given file has been specified as a resident (by a FILE OCL statement that has RETAIN-T specified) outside the LOAD and RUN OCL statement pair.
- Job files created by the released step are treated as scratch files; that is, those files cannot be used by following steps in this procedure.
- The RELEASE-YES parameter is ignored if the OCL statements for the job step contain a WORKSTN statement that specifies REQD-YES for the requesting display station.

- System messages issued by the released step are displayed at the system console, not at the display station that released the job step.
- A released step uses a copy of the user program status indicator (UPSI) switches for the requesting display station and the display station local data area as they exist when the released step is started. If the released step changes the display station local data area or the switches, the changes remain in effect only during the job step. The changes are not seen by following steps in the procedure or by later jobs submitted from the display station.
- If RELEASE-YES is specified for a job step that runs a multiple requester terminal (MRT) program that is also defined as a never-ending program (NEP), the MRT program is started but has no display stations. The MRT program then waits for the next requesting display station.
- If RELEASE-YES is specified for a procedure that is evoked through the Interactive Communications Feature (SSP-ICF), the SSP-ICF session is released from the evoked procedure.

NOTIFY specifies whether you want a message when the job termination has ended.

JOB specifies that you get a message when the job has started via

```
// EVOKE,NRT(// ATTR RELEASE-YES),
or // JOBQ has ended.
```

NO specifies that no message is sent.

MRTWAIT specifies, if a request causes the maximum number of MRTs to be exceeded, whether the operator is to wait until the MRT can be attached. If MRTWAIT is not specified, MRTWAIT-YES is assumed.

YES specifies that the operator will not get control back until the MRT can be attached.

NO specifies that the operator will get control back, and a return code of 2045 will be returned. This return code can be checked by using the ?CD? substitution expression.

Example 1

To prevent an operator from ending the payroll program by taking option 2 or 3 from the inquiry display, you could do the following:

```
// ATTR CANCEL-NO
// LOAD PAYROLL
// FILE NAME-EMPLOYES
// PRINTER NAME-PRINT,DEVICE-P2,FORMSNO-CHCK,ALIGN-YES
// RUN
```

Example 2

To assign high processing priority to a job and release it from the display station, you could do the following:

```
// ATTR PRIORITY-HIGH,RELEASE-YES
// LOAD PROG1
// RUN
```

CANCEL OCL Statement

The CANCEL OCL statement cancels one or more spool file entries. You can use the CANCEL OCL statement to cancel:

- All of your spool file entries
- All of your spool file entries with a specific forms number

If you control one or more printers, you can also cancel:

- All spool file entries for a specific printer that you control
- All spool file entries for all printers that you control
- All spool file entries with a specific forms number for all printers that you control
- All spool file entries with a specific user ID for all printers that you control

Note: If you are operating the system console or a system service display station, you control all printers. You also control all printers if password security is active and you have system operator authority or higher. If you are operating a subconsole display station, you control those printers for which the display is a subconsole.

Placement: The CANCEL OCL statement can be placed anywhere among the OCL statements. It may be evoked or submitted to the job queue. However, if you evoke or submit a CANCEL OCL statement to the job queue, it is assumed you do not control any printers. The only exception to this is, if password security is active and you have system operator authority or higher, you control all printers.

```
// CANCEL   PRT,{printer id          }
            (P) {ALL                 }
                {FORMS,forms number  }
                {USER,user id        }
```

S9020615-0

PRT or P specifies that one or more entries are to be canceled from the spool file.

printer id specifies the 2-character ID of a printer for which all spool file entries are to be canceled. You must control the printer.

ALL specifies that all spool file entries for all of the printers that you control are to be canceled. If you do not control any printers, only your spool file entries are canceled.

FORMS specifies that spool file entries with the specified forms number are to be canceled. If you control any printers, all entries with the specified forms number are canceled for all printers that you control. If you do not control any printers, only your spool file entries with the specified forms number are canceled.

forms number specifies the 1- to 4-character forms number of the spool file entries that are to be canceled.

USER specifies that spool file entries with the specified user ID are to be canceled. If you specify your own user ID, all of your spool file entries are canceled. If you specify a user ID other than your own, all spool file entries with the specified user ID are canceled for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries that are to be canceled. You can determine the user ID by using the STATUS PRT or STATUSF PRT control commands.

Example 1

This example shows how all spool file entries to be printed on printer P2 are canceled.

```
// CANCEL PRT,P2
```

or:

```
// CANCEL P,P2
```

Example 2

This example shows how all spool file entries with forms number 1234 are canceled for all printers that you control.

```
// CANCEL PRT,FORMS,1234
```

or:

```
// CANCEL P,FORMS,1234
```

CHANGE OCL Statement

The CHANGE OCL statement changes entries on the spool file. You can use the CHANGE OCL statement to change:

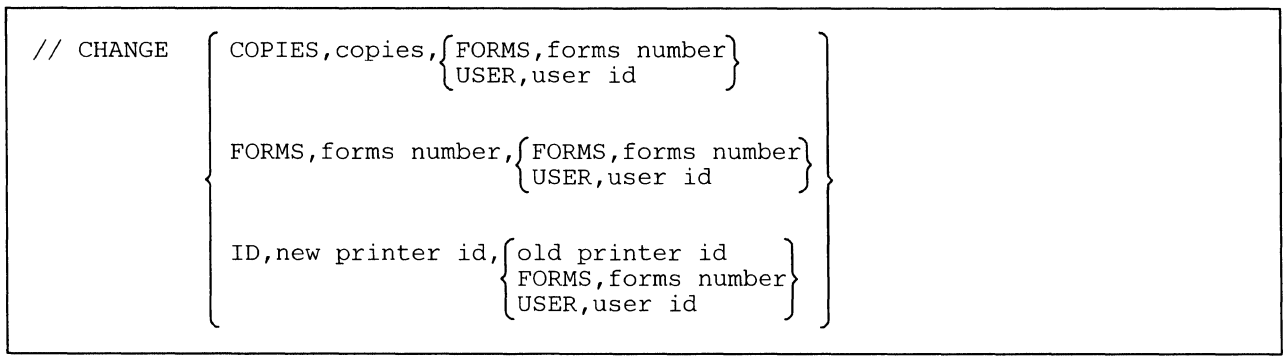
- The number of copies to be printed of your spool file entries
- The number of copies to be printed of all of your spool file entries with a specific forms number
- The forms number to be used for your spool file entries
- The new forms number to be used for all of your spool file entries with a specific forms number
- The printer to be used for your spool file entries
- The printer to be used for all of your spool file entries with a specific forms number

If you control one or more printers, you can also change:

- The number of copies to be printed of all spool file entries with a specific forms number for all printers that you control
- The number of copies to be printed of all spool file entries with a specific user ID for all printers that you control
- The new forms number to be used for all spool file entries with a specific forms number for all printers that you control
- The forms number to be used for all spool file entries with a specific user ID for all printers that you control
- The printer to be used for all spool file entries from among the printers that you control
- The printer to be used for all spool file entries with a specific forms number for all printers that you control
- The printer to be used for all spool file entries with a specific user ID for all printers that you control

Note: If you are operating a system console or a system service display station, you control all printers. You also control all printers if password security is active and you have system operator authority or higher. If you are operating a subconsole display station, you control those printers for which the display is a subconsole.

Placement: The CHANGE OCL statement can be placed anywhere among the OCL statements. It may be evoked or submitted to the job queue. However, if you evoke or submit a CHANGE OCL statement to the job queue, it is assumed you do not control any printers. The only exception to this is, if password security is active and you have system operator authority or higher, you control all printers.



COPIES specifies that the number of copies of printed output for one or more spool file entries is to be changed.

copies specifies the number of copies to be printed and can be any number from 1 to 255.

FORMS specifies that the number of copies of spool file entries with the specified forms number is to be changed. If you control any printers, all entries with the specified forms number are changed for all printers that you control. If you do not control any printers, only your spool file entries with the specified forms number are changed.

forms number specifies the 1- to 4-character forms number of the spool file entries for which the number of copies is to be changed.

USER specifies that the number of copies of spool file entries with the specified user ID is to be changed. If you specify your own user ID, all of your spool file entries are changed. If you specify a user ID other than your own, all spool file entries with the specified user ID are changed for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries for which the number of copies is to be changed. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

FORMS specifies that the printout form for one or more spool file entries is to be changed.

forms number specifies the new 1- to 4-character forms number of the printout form to be used.

FORMS specifies that the printout form for spool file entries with the specified forms number is to be changed. If you control any printers, all spool file entries with the specified forms number are changed for all printers that you control. If you do not control any printers, only your spool file entries with the specified forms number are changed.

forms number specifies the current 1- to 4-character forms number of the spool file entries for which the printout form is to be changed.

USER specifies that the printout form for spool file entries with the specified user ID is to be changed. If you specify your own user ID, all of your spool file entries are changed. If you specify a user ID other than your own, all spool file entries with the specified user ID are changed for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries for which the printout form is to be changed. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

ID specifies that the printer to be used for selected spool file entries is to be changed.

Note: If you direct your printed output to another printer with different printer characteristics, this may result in printing or programming errors, or your output may not print properly. (DW/36 jobs may not complete successfully.)

new printer id specifies the 2-character ID of the new printer to be used.

old printer id specifies the 2-character ID of the current printer. All spool file entries to be printed using this printer will use the new printer. If a spool file entry is currently printing on the old printer, it will continue to print on that printer. You must control the printer.

FORMS specifies that the new printer is to be used by spool file entries with the specified forms number. If you control any printers, all spool file entries with the specified forms number are changed for all printers that you control. If you do not control any printers, only your spool file entries with the specified forms number are changed.

forms number specifies the 1- to 4-character forms number of the spool file entries for which the printer is to be changed.

USER specifies that the new printer is to be used by spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries are changed. If you specify a user ID other than your own, all spool file entries with the specified user ID are changed for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries for which the printer is to be changed. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

Example 1

This example shows how to change the number of copies of printout for all spool file entries with user ID AA120199. Only one copy of these spool file entries is supposed to be printed, but you want three copies to be printed.

```
// CHANGE COPIES,3,USER,AA120199
```

Example 2

This example shows how to change the printout form for all spool file entries with current forms number 1324. These spool file entries were supposed to be printed on printout form 1324, but you want them to be printed on printout form 6978.

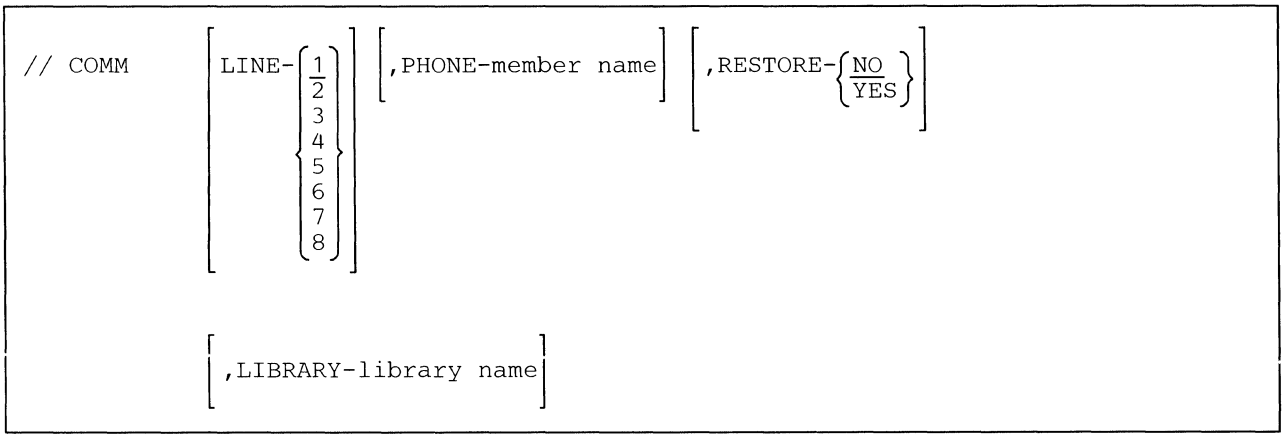
```
// CHANGE FORMS,6978,FORMS,1324
```

COMM OCL Statement

The COMM OCL statement is used by batch BSC programs to:

- Assign a line number to a program using communications
- Assign a phone list to a program using the autocall feature
- Specify whether the phone list should be restored for the program using the autocall feature
- Specify a library to be searched for the phone list.

Placement: The COMM OCL statement must be placed between the communication program's LOAD and RUN OCL statements.



S9020295-0

LINE specifies the communications line number to be associated with the communications program. If no line number is specified, 1 is assumed.

PHONE specifies the member name of the phone list (created by the DEFINEPN or DEFINX21 procedure) that is to be used by the autocall feature.

RESTORE specifies whether the phone list used by a previous step in the job should be restored to its original state; that is, whether the phone list will be set to indicate no calls have been attempted.

NO specifies that the phone list specified by the PHONE parameter and used in the previous job step is not to be restored to its original state. Note that the same phone list used in the previous job step is used in this job step (provided their names are the same); that is, no check will be made to determine whether the previous phone list and the specified phone list are from different libraries. If no parameter is specified, NO is assumed.

YES specifies that the phone list specified by the PHONE parameter is to be restored to its original state. The SSP first searches the current library, then the system library (#LIBRARY), for the phone list.

LIBRARY specifies the library to be searched initially for the phone list. If the phone list is not found in the specified library, the system library (#LIBRARY) is searched. If this parameter is not specified, the system first searches the current library, then the system library, for the phone list. If the phone list is not found in any of these libraries, a message is issued.

Example

This example shows how to assign a communications line number and phone list to a program. The program will run for each number in the phone list.

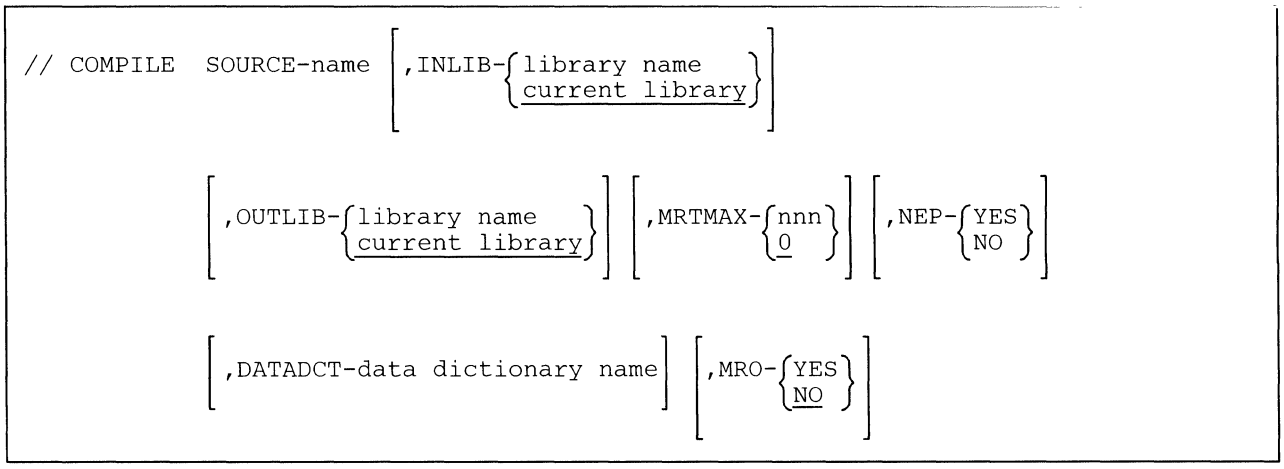
```
// TAG CALL
// LOAD CALLPROG
// COMM LINE-1,PHONE-PHONELST,RESTORE-NO
// RUN
// IFF LISTDONE-PHONELST GOTO CALL
* If the list is not completed,
* run the program to call the next number
```

COMPILE OCL Statement

The COMPILE OCL statement supplies information required by the SSP when a library source member is compiled. The COMPILE OCL statement does the following:

- Identifies the library source member that contains the source program to be compiled. A source program is a collection of statements, such as RPG II specifications, that can be compiled into a running program.
- Identifies the library that contains the source program.
- Identifies the library that will contain the compiled load member. The load member can then be loaded and run using the LOAD and RUN OCL statements.
- Specifies the maximum number of requesting display stations that can be attached to the program, if the program is a multiple requester terminal (MRT) program.
- Specifies whether the program is a never-ending program (NEP).
- Specifies the data dictionary to be used by the program.
- Specifies whether the compiler is to use memory resident overlays.

Placement: The COMPILE OCL statement must be placed between the LOAD and RUN OCL statements of the job step that compiles the source program. If the source program is in the procedure or keyboard job stream, the COMPILE OCL statement may be omitted.



S9020296-1

SOURCE specifies the name of the source member that contains the source program to be compiled.

INLIB specifies the name of the library that contains the source program. If INLIB is specified, only that library is searched. If INLIB is not specified, the current library is assumed.

OUTLIB specifies the name of the library that will contain the compiled load member. If OUTLIB is not specified, the current library is assumed. The name of the load member is either specified in the source program or is the same as the source program.

MRTMAX identifies the program as a multiple requester terminal (MRT) program.

nnn specifies the maximum number of requesting display stations that can be attached to the program (leading zeros need not be entered).

If nnn is 0 or if MRTMAX is not specified, the object program is not an MRT program. If MRTMAX is specified, the maximum number of requesting terminals can be changed by an ATTR statement when the object program is run.

NEP specifies whether the program is a never-ending program (NEP). NEP is defined as a long-running program. Any system resources, except for shared files and the spool file, that are allocated to a NEP are not available to other jobs.

YES specifies that the program is a NEP.

NO specifies that the program is not a NEP.

The NEP indicator can be changed by an ATTR statement when the program is run.

The following special considerations exist for jobs run from the job queue and for multiple requester terminal (MRT) programs:

- All programs run from the job queue are run as NEPs, unless NEP-NO is specified on the ATTR OCL statement.
- For a MRT program, if NEP-YES is not specified on the ATTR OCL statement and if NEP-YES is not specified on the COMPILE OCL statement, other programs will not wait for non-shared resources being used by the MRT program. However, when the MRT program releases its last requesting display station, the program goes to end-of-job processing.
- If NEP-YES is specified for a MRT program, other programs will not wait for non-shared resources being used by the MRT program. The MRT program will not end when it releases its last requesting display station. Instead, the MRT program will wait until it is requested by another display station. Normally, a MRT program with NEP-YES specified will not end until after the system operator enters the STOP SYSTEM command and all requesting terminals are released. (The system operator can end the job at any time by using the CANCEL command.)

For information about NEPs, SRTs, and MRTs, see the *Concepts and Programmer's Guide*.

DATADCT specifies the name of the data dictionary that contains the format definitions to be used by the program.

MRO specifies whether the compiler is to use memory resident overlays. If MRO is not specified, memory resident overlays are not used.

YES specifies that the compiler should use memory resident overlays.

NO specifies that the compiler should not use memory resident overlays.

Example

The source member named PROG3, in the library named MYLIB, is to be compiled. The compiled load member will be placed in the system library. The program PROG3 will be run as a NEP.

```
// COMPILE SOURCE-PROG3,INLIB-MYLIB,OUTLIB-#LIBRARY,NEP-YES
```

DATE OCL Statement

A DATE OCL statement entered anywhere among the OCL statements other than between LOAD and RUN OCL statements changes the session date. (A session begins when an operator signs on and, normally, ends when the operator enters the OFF command.) If no DATE statement or DATE procedure establishes a session date, the system date specified during initial program load is used as the session date.

A DATE OCL statement that is entered between LOAD and RUN OCL statements specifies the program date (also known as the job step date). When the job step ends, the date is set back to the session date. If a DATE statement is not entered or if the DATE procedure is not run between a LOAD statement and a RUN statement, the session date is used as the program date. If two or more DATE statements are between a LOAD statement and a RUN statement, the last DATE statement is used.

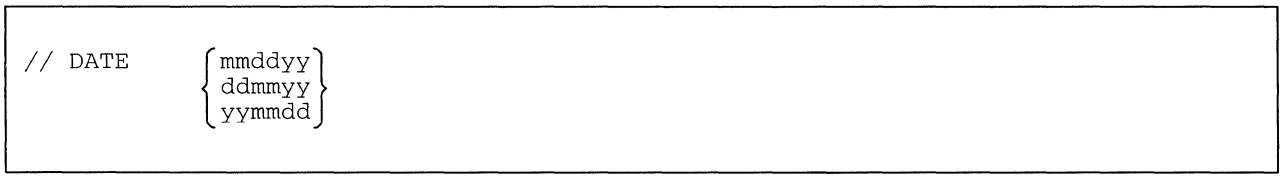
Notes:

1. *The program date is used to determine how long a diskette file is to be retained and is printed on printed output. The program date is also used as the creation date of any disk or diskette files created by the program.*

2. *If a job is placed on the job queue, the program date when the job was placed on the queue is assigned to the job.*

3. *If 2400 hours occurs, the system date is automatically updated, but the session date and program date are not updated for active sessions or programs.*

Placement: A DATE statement can appear anywhere among the OCL statements.



S9020297-0

The date specified on the DATE statement must be in the current session date format. The session date can be in any of three formats: month-day-year (mmddyy), day-month-year (ddmmyy), or year-month-day (yymmdd). The STATUS command can be used to determine the current session date format, and the SET procedure can be used to change the current session date format.

Month, day, and year must each be 2-digit numbers, but leading zeros in month and day can be omitted when punctuation is used. In the punctuated form, any characters except commas (,), apostrophes ('), numbers, and blanks can be used as punctuation. However, question marks (?) should not be used as punctuation.

The date can be entered with or without punctuation. For example, July 24, 1983 can be specified in any of the following ways:

Date	Date Format
7-24-83	mm-dd-yy
24-7-83	dd-mm-yy
83/7/24	yy/mm/dd
072483	mmddyy
240783	ddmmyy
830724	yymmdd

A date of all zeros (000000) is invalid.

Example 1

The DATE statement for July 1, 1983:

```
// DATE 070183
```

or:

```
// DATE 7-1-83
```

Example 2

To specify a program date (job step date) for the PAYROLL program, you could place the DATE statement between the LOAD and RUN statements for the PAYROLL procedure. When the PAYROLL program ends, the session date is used for the PAYPRNT program.

The PAYROLL program reads the EMPLOYES file and calculates the paychecks. Included on the checks is the date. By specifying the program date using the DATE statement, you can run the program before the actual payroll date (indicated by parameter 1, the ?1R? expression). The PAYPRNT program does not use the date.

The example PAYROLL procedure contains the following statements:

```
* PAYROLL  procedure
* Calculate paychecks using date specified
// LOAD PAYROLL
// DATE ?1R?
// FILE NAME-EMPLOYES,DISP-OLD
// FILE NAME-CHECKS,RETAIN-J,BLOCKS-30
// RUN
*
* Print check information on check forms (CHCK)
* Allow the checks to be aligned in the printer
// LOAD PAYPRNT
// FILE NAME-CHECKS,RETAIN-J
// PRINTER NAME-PRINT,SPOOL-NO,LINES-25,FORMSNO-CHCK,
//         ALIGN-YES,DEVICE-P3
// RUN
```

To run the procedure and specify a payroll date of 7-1-84, you would enter:

```
PAYROLL 7-1-84
```

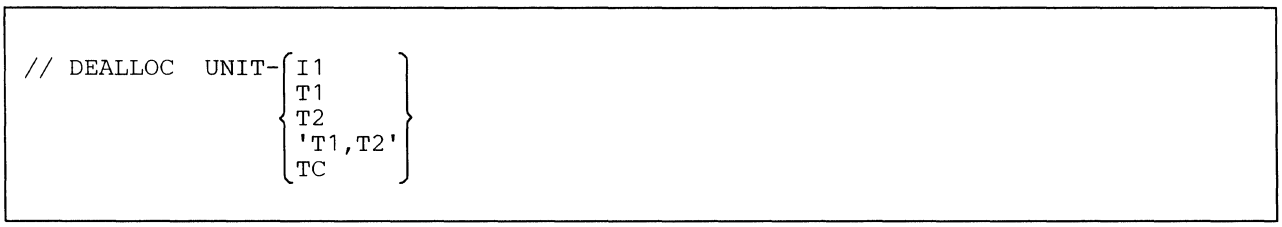
and the DATE statement would be processed as:

```
// DATE 7-1-84
```

DEALLOC OCL Statement

The DEALLOC OCL statement frees the diskette drive or tape drives after an ALLOCATE OCL statement has allocated the drive. This allows other jobs on the system to use the drive.

Placement: The DEALLOC statement can be placed anywhere among the OCL statements.



S9020298-1

UNIT specifies which drive is to be freed so that other programs can use it.

I1 specifies that the diskette drive is to be freed.

T1 specifies that tape drive 1 is to be freed.

T2 specifies that tape drive 2 is to be freed.

'T1,T2' specifies that both tape drive 1 and tape drive 2 are to be freed. This option must be enclosed within single quotes and separated by a comma.

TC specifies that the tape cartridge is to be freed.

Example

In this example, the diskette drive is allocated for three procedures and is then freed before program PROG1 is loaded so that other jobs can use the drive.

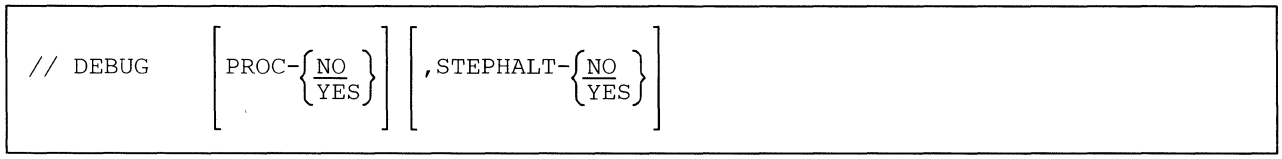
```
// ALLOCATE UNIT-I1,WAIT-YES
SAVE FILE1,,,VOL001
SAVE FILE2,,,VOL001
SAVE FILE3,,,VOL001
// DEALLOC UNIT-I1
// LOAD PROG1
// RUN
```

DEBUG OCL Statement

The DEBUG OCL statement specifies whether the procedure control expressions and OCL statements in your procedures should be listed as they are evaluated, and whether the procedures should stop after each job step. This allows you to follow the logic path of your procedures.

Placement: The DEBUG OCL statement can be placed anywhere among the OCL statements. If the DEBUG OCL statement is entered from the keyboard, the parameters specified remain in effect until another DEBUG OCL statement is entered, or until you sign off the system (that is, the OFF command is entered).

If the DEBUG OCL statement is specified in a **procedure**, it remains in effect until another DEBUG OCL statement is processed, or until the procedure ends. When the procedure ends, the parameters entered at the **keyboard** (if any) take effect.



S9020299-0

PROC specifies whether the procedure control expressions and OCL statement should be listed as they are evaluated. The information is listed on the system list device. You can determine your present system list device by entering the STATUS SESSION command. The SYSLIST procedure sets the system list device.

If the system list device is the display station, and the listing of the job step being shown does not fill the display, no message is displayed indicating the end of system list data before the next job step is shown. It is possible that other system list output generated by your procedure could be combined with the procedure control expressions and OCL statements.

NO specifies that listing is not to occur. If PROC is not specified, NO is assumed.

YES specifies that listing is to occur.

Note: The output will go to the system list device that was in effect when the DEBUG OCL statement was encountered. That system list device will remain in effect until debugging is turned off; that is, any SYSLIST procedure or OCL statement, or any PRINTER OCL statement having the NAME-$SYSLIST parameter, will be ignored if debugging is on.

STEPHALT specifies whether the procedure should be halted (that is, temporarily stopped) after each job step in the procedure.

NO specifies that the procedure is to run normally. If STEPHALT is not specified, NO is assumed.

YES specifies that a message is to be displayed after each job step in the procedure is completed. The message indicates the name of the first-level procedure (if you are running a procedure) and the name of the program that was run, and the operator can select an option to either continue or to cancel the procedure.

Example

This example shows how to use the DEBUG statement to view the procedure control expressions used during a sample procedure named SAMPLE.

The SAMPLE procedure contains the following statements:

```
* SAMPLE Procedure
*  Parameter 1:  File name (required, prompted for if omitted)
*  Parameter 2:  File's creation date (optional)
// * 'SAMPLE PROCEDURE RUNNING'
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-?1R?,
//   IFF ?2?= DATE-?2?,
//   UNIT-F1
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

These statements are used to run the procedure and produce the debug listing:

```
// DEBUG PROC-YES
SAMPLE FILE1
```

The following shows the output produced by the DEBUG statement:

```
SAMPLE FILE1
* SAMPLE Procedure
*  Parameter 1:  File name (required, prompted for if omitted)
*  Parameter 2:  File's creation date (optional)
// * 'SAMPLE PROCEDURE RUNNING'
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-?1R?,
   FILE NAME-COPYIN,LABEL-FILE1,
//   IFF ?2?= DATE-?2?,
     IFF = DATE-?2?,
//   UNIT-F1
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

EVOKE OCL Statement

The EVOKE OCL statement can be used to start a procedure (that is, the procedure is evoked). When a procedure is evoked, it begins running as a separate job, and control returns immediately to the calling procedure. Thus, it is possible to have several procedures running at once as a result of several EVOKE OCL statements. The EVOKE OCL statement may be most useful for jobs that require no operator input or response, or for jobs whose output is not required by a subsequent job step.

When a job is evoked, the priority of the evoked job is the same as the priority of the job that issued the evoke.

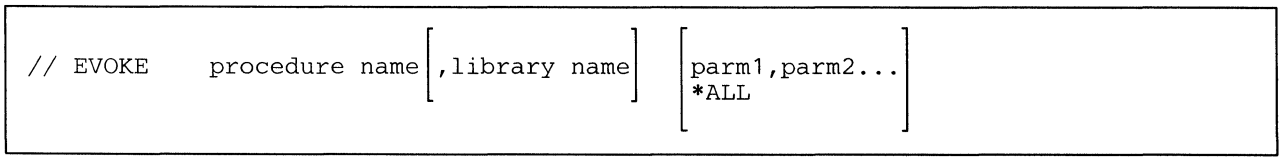
Messages issued from the evoked job are displayed at the system console, not at the display station (if other than the system console) that issued the evoke operation. When a procedure is evoked, the evoked procedure uses a copy of the UPSI switches and the local data area as they exist on the requesting display station. If the evoked procedure changes the UPSI switches or the local data area, the changes are in effect only for the evoked job; the changes are not seen by later jobs run or evoked at the requesting display station.

If no REGION OCL statement is specified in the procedure containing the EVOKE OCL statement, the evoked procedure uses the session region size specified during system configuration or specified by the SET procedure. If a REGION OCL statement is specified in the procedure containing the EVOKE OCL statement, the evoked procedure uses the region size specified in the REGION OCL statement.

An MRT procedure cannot be evoked by the EVOKE OCL statement.

Note: You should not use too many EVOKE OCL statements, because that might slow down the system. A large number of procedures being evoked may cause the system to prevent initiation of procedures.

Placement: The EVOKE statement can be placed anywhere among the OCL statements except between the LOAD and RUN OCL statements.



S9020300-0

procedure name specifies the procedure to be evoked.

library name specifies the library to be searched for the procedure. If the procedure is not found in the specified library, the system library (#LIBRARY) is searched. If no library name is specified, the current library is searched first, and then the system library (#LIBRARY).

parm1,parm2, ... specifies parameters for the procedure. Parameters are not allowed if PDATA-YES was specified when the procedure member was created. The parameters may or may not be required, depending on the procedure they are passed to.

A parameter can be any combination of characters except commas or blanks. Question marks (?), apostrophes ('), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The total number of characters for one parameter must not exceed 128. A maximum of 64 parameters, separated by commas, can be passed with an EVOKE statement.

When you are entering parameters, you can type up to 512 characters. For example, you can type 32 sixteen-character parameters or 64 eight-character parameters. However, the combined total length of **all** parameters cannot exceed 1024 characters; this length can be accomplished by using substitution expressions and the local data area. See "Continuing the Lines of a Procedure" on page 2-7 for information about how to continue input lines to get more characters than 120.

See the "INCLUDE OCL Statement" on page 5-63 for more information about how you can enter parameters. Program data cannot be specified.

***ALL** specifies that all parameters are to be passed from the current procedure level to the procedure being called. *ALL can only be specified within a procedure. If *ALL is specified as the only parameter and it is entered from the keyboard or selected by a menu item, an error message is displayed. If *ALL is specified as one of the parameters, it is treated as a single parameter.

*Note: The history file record of the EVOKE OCL statement will show only *ALL, not the parameters that are passed with the EVOKE statement.*

Example

In the following example, the OCL statements include an EVOKE OCL statement that calls and passes parameters to the procedure PROC1. When PROC1 is evoked, it starts running and control immediately returns to the calling procedure. Thus, both procedures are running at the same time and independently.

```
// EVOKE PROC1 parm1,parm2
// LOAD PAYROLL
// RUN
```

FILE OCL Statement (for Disk Files)

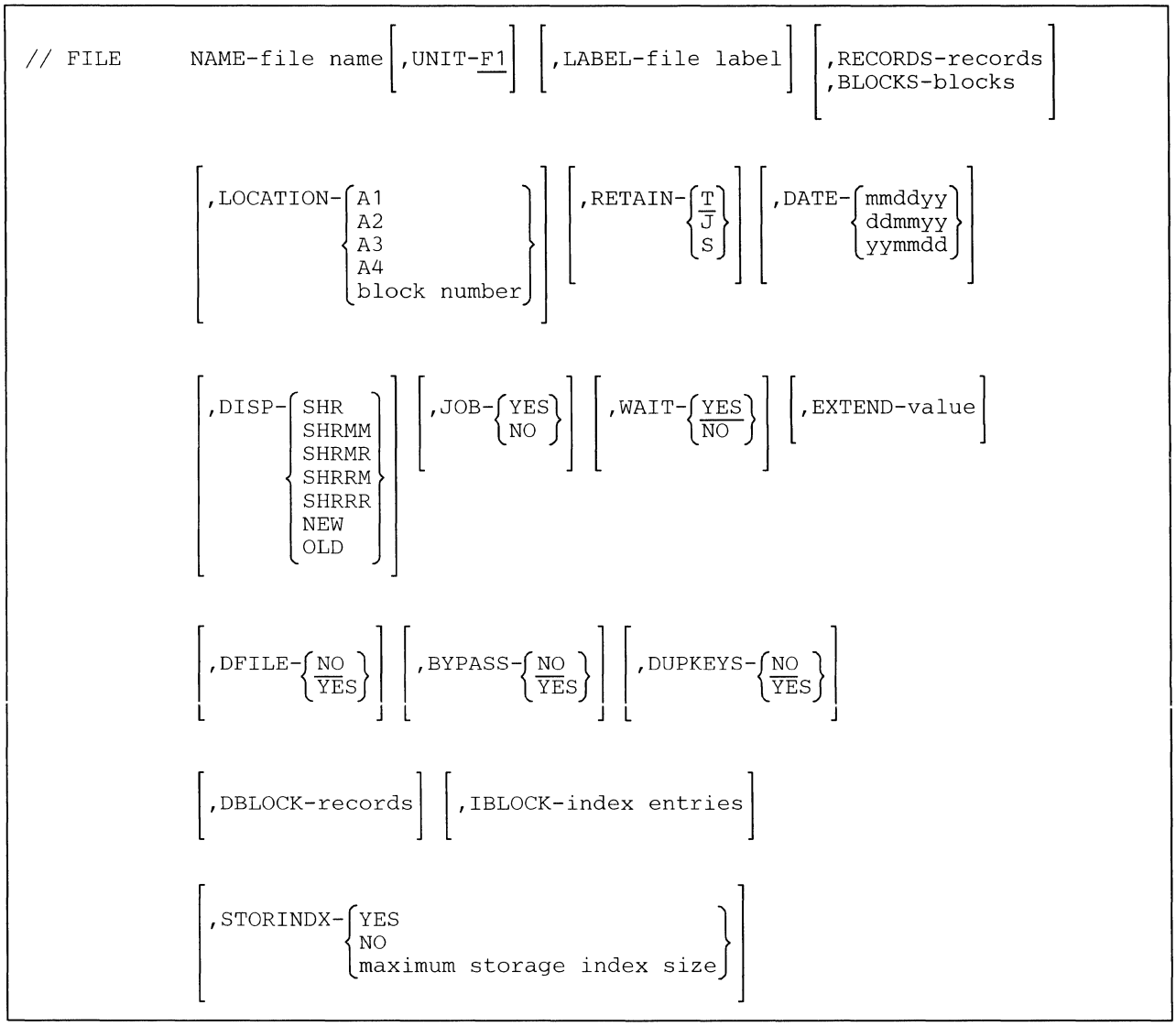
The FILE OCL statement for disk files supplies the SSP with information about disk files. The SSP uses this information to read records from and write records to the disk file.

For information about System/36 disk file concepts, see the manual *Learning about Your Computer* and the *Concepts and Programmer's Guide*.

Placement: A FILE OCL statement is required for each disk file that a program creates or uses. The FILE OCL statement can be placed anywhere among the OCL statements.

If the FILE statement is placed before a LOAD statement, the SSP immediately attempts to make the user the owner of the disk file. (See the *JOB* keyword for information on the duration of file acquisition.) Acquiring a disk file in this way establishes only the ownership level of the file (that is, which programs can use the file), either shared or old. The file is not actually created or used until it is allocated, opened, and used by a program (some programming languages combine the allocate and open steps into one operation). This allows you to determine, prior to starting a series of job steps, whether all the files required by the jobs steps are available. Also, the file is treated as if the program has the never-ending program (NEP) indicator (see the "ATTR OCL Statement" on page 5-11 for information on the NEP indicator). This prevents other jobs from waiting for the file if it cannot be shared.

If the FILE OCL statement is placed between the LOAD and the RUN OCL statements for a job step, the file is acquired when the RUN OCL statement is processed.



S9020301-1

NAME specifies the name that the program uses to refer to the file. The name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. Question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, $, or @). The number of characters in a file name must not exceed eight.

UNIT-F1 specifies that the file is on the disk. This parameter need not be specified for a disk file because F1 is the assumed value for the UNIT parameter.

LABEL specifies the actual name or label by which a file is identified on the disk. You only need to specify the LABEL parameter when the actual name of the disk file is different from the name used in the program. If the LABEL parameter is omitted from a disk FILE statement, the file name from the NAME parameter is used. If a program does not refer to an existing file by the file's actual name on the disk, a LABEL parameter must be supplied.

File names can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. Question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The first character of a name must be alphabetic (A through Z, #, $, or @). The number of characters in a name must not exceed eight.

The name could be a file that belongs to a file group. If the file name contains a period, the file is a member of a file group. The characters in front of the period identify the file group. See the "SAVE Procedure" on page 4-416 for information about file groups.

RECORDS specifies the number of records to reserve for the file. The total space to be reserved is rounded up to the next block (one block contains 2560 bytes), allowing enough space to contain at least the number of records indicated. The smallest disk file unit that can be reserved is one block. For example, if you specify ten 50-byte records (that is, 500 bytes of space are needed for the file), then 2560 bytes (one block) are reserved.

records cannot exceed the value 8000000.

Either RECORDS or BLOCKS, but not both, can appear in the FILE OCL statement.

The RECORDS or BLOCKS parameter must be used for a new file.

BLOCKS specifies the number of disk blocks to reserve for the file. There are 2560 bytes in one disk block.

blocks cannot exceed the configured maximum.

Either RECORDS or BLOCKS, but not both, can appear in the FILE OCL statement.

The RECORDS or BLOCKS parameter must be used for a new file. You can use the CATALOG procedure to determine the number of disk blocks available for files.

LOCATION indicates a preferred disk placement for a new disk file, or specifies the number of the block where a new file should begin. A1, A2, A3, and A4 indicate the disk units (also called spindles). A1 indicates the first disk unit; A2 indicates the second disk unit; A3 indicates the third disk unit; A4 indicates the fourth disk unit.

If you do not specify a preferred disk unit location, or if not enough space is available on the disk unit that you specify, the SSP attempts to place the file on the least used disk unit. This is done to better balance the load between the disk units. Files and libraries can span disk units; for example, a file can begin on A1 and continue onto A2.

A1: If you specify a preferred placement on disk A1 for a system with more than one disk drive, the SSP places the file in the first segment (lowest address) of available storage on disk A1 that is large enough to contain the file.

If you specify A1, or do not specify LOCATION, for a system with a single disk unit, the SSP places the file in the first segment (lowest address) of available storage that is large enough to contain the file.

A2: If you specify A2 for a system with more than one disk drive, the SSP places the file in the last segment (highest address) of available storage on disk unit A2 that is large enough to contain the file.

If you have only one disk unit and you specify A2, disk unit A1 is assumed.

A3: If you specify A3 for a system with more than one disk drive, the SSP places the file in the last segment (highest address) of available storage on disk unit A3 that is large enough to contain the file.

If you have only one disk unit and you specify A3, disk unit A1 is assumed.

A4: If you specify A4 for a system with more than one disk drive, the SSP places the file in the last segment (highest address) of available storage on disk unit A4 that is large enough to contain the file.

If you have only one disk unit and you specify A4, disk unit A1 is assumed.

block number specifies the location of the file; up to 6 digits can be specified. You can use the CATALOG procedure to determine where blocks of available storage are located on the disk. You can use this parameter to create a new file at a specific disk location. If space is not available, an error is issued.

RETAIN classifies the file as a **resident (T)**, **job (J)**, or **scratch (S)** file when it is created or referenced.

If the RETAIN parameter is omitted from the FILE statement when the file is created or referenced, the file is assumed to be a resident file. However, if there is an existing job file with the same label, the job file is used.

T specifies a **resident** file. A resident file remains on the disk when the job ends. The area containing a resident file becomes available for another file only under one of the following conditions:

- A FILE OCL statement containing RETAIN-S is supplied for the resident file to change the file retention to scratch. If the file is used by the program, the file will not exist after the program ends normally; if the program does not end normally, the files will not be changed to scratch files and will still exist as resident files. If the file is not used by the program, the file will remain a resident file after the program ends.
- Another file with the same label is loaded into the area occupied by the resident file, changing the data or the organization of the file. The DISP-OLD parameter must be specified.
- The DELETE procedure is used to delete the file.

The total number of resident files that can exist at any one time is limited by the maximum number of VTOC entries specified when the system was configured. Any additional scratch and job files can exist. The actual number of files that can be placed on the disk depends upon the size of the disk and the size of the files.

J specifies a **job** file. A job file, after it is created, can be used by all of the remaining job steps in a job. The job file is defined only within the job and does not exist after the job ends. If a job step is released (using the RELEASE parameter on the ATTR OCL statement) or runs a multiple requester terminal (MRT) program, that step cannot use job files defined by other steps of the job. Any job files created by a released job step or a job step that runs an MRT program are treated as scratch files and cannot be used by other steps in the job. Two or more jobs using job files with the same name can run at the same time because the job files are defined only within the individual jobs.

When a file is created with RETAIN-J specified, following steps in the job can refer to that file by not specifying a RETAIN parameter or by specifying RETAIN-J or RETAIN-S. If a following step specifies RETAIN-J or does not specify RETAIN, the file is kept for later use by other job steps. However, if a following step specifies RETAIN-S for an existing job file, that file is removed at the end of that step and cannot be used by following steps in the job.

If a file is created with RETAIN-J, a following job step can use or create another disk file with the same label by specifying RETAIN-T. A resident disk file cannot be processed in the same job step with a job file having the same label.

Note: If a system failure occurs, the contents of a job file or a new scratch file are lost.

S specifies a **scratch** file. A scratch file or a resident/job file being scratched cannot be stored. It does not exist after the job step has ended.

DATE specifies the creation date of an existing file and can be used to ensure that the proper version of a file is used. The DATE parameter is not allowed when DISP-NEW is specified.

When a file is created on disk, its label and creation date are written on the disk as identification. The job step date when the file is created is the date used as the creation date. If date differentiated files was specified during system configuration, then more than one resident file can be given the same label; however, the creation dates of these files must be different. To refer to such a file, specify its label and date. If the date is not specified, the file having the latest date is used.

The date can be entered in one of three formats: month-day-year (mmddyy), day-month-year (ddmmyy), or year-month-day (yymmdd). However, the format chosen must be the same as the session date format. The STATUS SESSION command can be used to determine the session date format.

DISP (disposition) specifies that the file is a new file or an old file, or that the file can be shared by other jobs running on the system. The DISP parameter is not allowed if RETAIN-J is specified. If DISP is not specified, the system determines whether a file is new or old based upon whether the file is in the disk VTOC. This is not the same, however, as assigning NEW or OLD as the DISP parameter. DISP-OLD must be specified to overlay a file, even if the system has determined that the file is old.

SHR specifies that the file already exists and can be shared by other programs running on the system. Read, update, delete, and add operations can be performed on the file. SHR is the same as SHRMM.

A description of programming considerations for file sharing can be found in the *Concepts and Programmer's Guide*.

SHRMM specifies that the program using the file can modify the file (that is, records can be read, updated, deleted, or added). Other programs that are sharing the file can also modify the file. SHRMM is the same as SHR.

SHRMR specifies that only the program using the file can modify the file (that is, records can be read, updated, deleted, or added). Other programs that are sharing the file can only read records from the file.

SHRRM specifies that the program using the file only needs to perform read operations on the file (that is, no records will be updated, deleted, or added). Other programs that are sharing the file can modify the file (that is, they can read, update, delete, or add records to the file).

SHRRR specifies that the program using the file only needs to perform read operations on the file (that is, no records will be updated, deleted, or added). Other programs that are sharing the file can also only read records from the file.

NEW specifies that the file is new. If a file already exists with the same label and creation date as the new file, an error message is displayed. The new file can be created using any disk file organization. The file cannot be shared by any other programs until the program that created it ends.

OLD specifies that the file already exists, and is not to be shared until the program that is using it ends. If the file does not exist, an error message is displayed. DISP-OLD allows you to process an existing file as an output file. When an existing file is overlaid (that is, the file contains totally new information), DISP-OLD must be specified. The creation date of the overlaid file is changed to the job step date, and all parameters are reset and must be respecified except the RECORD/BLOCKS and LOCATION parameters.

JOB specifies whether the disk file is to be acquired for the entire job. If JOB-NO is specified, or if a file is not already acquired and no JOB parameter is specified, the file will be acquired only until the job step ends.

YES specifies that the file should be acquired for the entire job. The share level specified by the DISP parameter will be used throughout the job.

JOB-YES can only be specified on a FILE OCL statement that is outside of a LOAD and RUN OCL statement pair.

If JOB-YES is specified, the other parameters specified are assumed each time the same file is used by programs in the following job steps. That is, if you use the same file in a following job step, you need not specify another FILE OCL statement for that job step.

If a subsequent FILE OCL statement with the JOB-YES parameter is encountered that has the same NAME parameter as a previous FILE OCL statement that had JOB-YES specified, and the file has not yet been created, all parameters specified on the current FILE OCL statement will permanently override the original defaults. Any parameters that are not specified on the current FILE OCL statement but were specified on the previous FILE OCL statement are lost. This, in effect, replaces the original parameter defaults.

If a subsequent FILE OCL statement without the JOB-YES parameter is encountered with the same NAME parameter as a previous JOB-YES FILE OCL statement, and the file has not yet been created, all parameters specified will override the original default specifications. Any parameters not specified on the current FILE OCL statement will default to what was specified on the previous JOB-YES FILE OCL statement. When the current job step ends, all the original default specifications for the file (except LOCATION, BLOCKS, and RECORDS) will again become the parameter defaults if the file was not created. The LOCATION, BLOCKS, and RECORDS parameters used to create the file are always from the most recently processed FILE OCL statement.

JOB-YES can be specified for a new file. The specified file will be acquired for exclusive use of the job and cannot be shared. The file is not actually created by specifying JOB-YES; the file is actually created when it is allocated, opened, and used by a program (some programming languages combine the allocate and open into one operation). If the LOCATION, BLOCKS, or RECORDS parameters are specified, they are only used when the file is created.

If RETAIN-S and JOB-YES are specified on a FILE OCL statement, the scratch file can be used only by the job step creating it and does not exist after the job step has ended.

If an ATTR OCL statement is specified with the RELEASE-YES parameter, only those files referenced by FILE OCL statements between the following LOAD and RUN OCL statement pair will be passed to the released job step. All other FILE OCL statements, including FILE OCL statements that have JOB-YES specified, will continue to be owned by the original job and can be used in following job steps. This means that a job step released by an ATTR OCL statement cannot own files that are specified outside of that job step's LOAD and RUN OCL statements.

When JOB-YES is specified for a single requester terminal procedure, and that procedure calls a multiple requester terminal (MRT) procedure that uses the same file, the share levels must be compatible on both FILE OCL statements in both procedures.

NO specifies that the file is to be used only for that job step.

WAIT specifies whether the procedure should wait for the file to become available.

YES specifies that if the file is already acquired by another job, the procedure should wait for the file to become available. If the file is allocated to or being used by an interrupted or waiting program, or a program with the never-ending program attribute, a message will be displayed and the operator can either retry the allocation operation or cancel the job.

NO specifies that if the file is already acquired by another job, the procedure should continue. NO can only be specified on a FILE OCL statement that is outside of a LOAD and RUN OCL statement pair. The ?CD? substitution expression can then be used to see whether the file was acquired by the procedure. See "?CD? (Return Code) Expression" on page 3-16 for more information about the values substituted by this expression.

EXTEND specifies that the file may be extended whenever additional space is needed.

Files can also be extended by specifying a value when the file is created (by this parameter or by the BLDFILE procedure). If an extend value is specified when a file is created, this EXTEND parameter need not be specified when the file is being updated.

value is a 1- to 8-digit block or record value that specifies the amount of additional space to use for the extension. When the file was created, the size of the file was specified in either blocks or records; use the same unit of measure here. You can use the CATALOG procedure to determine how the size of the file was specified. The extension should be large enough to contain at least one record. If EXTEND-0 is specified, no file extension will occur.

For all file types, and on add operations for sequential and indexed files, the file will be extended when the current size of the file is not large enough. The file will be extended on a read operation for a direct file if the record to be read is beyond the end of the file. If FILE OCL statements for more than one job specify values for the extension, the value from the FILE OCL statement in the job that caused the extension will be used. Note that when files are being shared, if one user causes an extension to occur, all users of that file can take advantage of the additional file space, regardless of whether they specified EXTEND on their FILE OCL statements.

Notes:

1. *The EXTEND parameter, if used on file statements for the system utilities $BICR, $POST, and $MAINT, is ignored. This parameter should not be used on Sort Utility work files.*

2. *To use the EXTEND parameter with job (J) files, you must specify EXTEND when the file is created. All following references within the job to that job file use the extend specification referenced when the file was created. If the EXTEND parameter is specified on following statements, it will be ignored.*

DFILE specifies whether the file should be delete-capable. With a delete-capable file, you can use the delete operation in your program to delete a record from the file (the record is not actually removed, but is made unreadable). The DFILE parameter is only allowed for new files or existing files that are to be completely reloaded with new information.

NO specifies that the file is not to be delete-capable. No delete operations can be performed by the programs using this file.

YES specifies that the file is to be delete-capable.

If DFILE-NO is specified for an existing delete-capable file, or if DFILE-YES is specified for an existing file that is not delete-capable, a message may be displayed, and you can either continue or cancel the job.

BYPASS specifies whether the SSP is to allow a program to add a record to an indexed file without checking for a duplicate index key.

NO specifies that the SSP should check for a duplicate index key before adding a record to the file. If no parameter is specified, BYPASS-NO is assumed.

YES specifies that the SSP should not check for a duplicate index key before adding a record to the file.

DUPKEYS specifies, for a new indexed file, whether duplicate keys are to be allowed. If the DUPKEYS parameter is specified for a file that is not a new indexed file, the DUPKEYS parameter is ignored.

NO specifies that the file should not contain duplicate keys. If an attempt is made to add a record that has the same key as a record in the file, an error occurs. If no parameter is specified, DUPKEYS-NO is assumed.

YES specifies that the file can contain duplicate keys. Checking for duplicate keys in the index is not performed even if BYPASS-NO is specified.

DBLOCK specifies the number of records to be moved between main storage and a disk file. This allows you to specify a record blocking value that is different from (overrides) the value specified in the program. If the DBLOCK parameter is not specified, the value specified in the program is used.

See the *Concepts and Programmer's Guide* for considerations about file record blocking.

records must be a number from 1 through 65535.

IBLOCK specifies the number of index entries to be moved between main storage and an indexed file. This allows you to specify an index entry blocking factor that is different (overrides) from the value specified in the program. If the IBLOCK parameter is not specified, the value specified in the program is used. If the IBLOCK parameter is specified for a file that is not an indexed file, the IBLOCK parameter is ignored.

index entries must be a number from 1 through 65535.

STORINDX specifies whether a storage index is to be built for an indexed disk file. If no parameter is specified, the SSP determines whether a storage index should be built. The storage index is also called a master track index.

YES specifies that a storage index should be built (if it does not already exist) for the indexed disk file. If a storage index already exists for the file, that storage index is used.

NO specifies that no storage index should be built for the indexed file. However, if a storage index already exists it will be used to process the file.

maximum storage index size specifies the maximum size in K-bytes of the storage index being built by the SSP. The maximum storage index size must be a number from 1 through 16. The specified number is rounded up to an even number. If a storage index already exists for the file, that storage index is used.

Example 1

A program is creating a disk file. Assume the following facts about the file:

- The name the program uses to refer to the file is OUTPUT.
- The label of the file on the disk is TRANSACT.
- The file contains 200 records and will be extended by 100 records whenever the file becomes full.
- The system chooses the disk area to contain the file.

The FILE statement that defines the file is:

```
// FILE NAME-OUTPUT,LABEL-TRANSACT,RECORDS 200,EXTEND-100
```

Example 2

A program writes over (that is, overlays) the contents of an existing disk file labeled IVENTORY. The file is located at block number 500 and is 12 blocks long. The file name used by the program is OUTFILE. The file is opened (in the program) for output.

```
// FILE NAME-OUTFILE,LABEL-IVENTORY,RETAIN-T,DISP-OLD
```

Example 3

A program is to create a delete-capable file named FILE1.

```
// FILE NAME-FILE1,DFILE-YES,RECORDS-250
```

Example 4

Programs PROG1 and PROG2 both need to use the file TRANSACT. You could use the following:

```
// FILE NAME-TRANSACT,JOB-YES,DISP-SHRMM,WAIT-NO
// IF ?CD?=0000 GOTO OK
// * 'The file is being used, do you want to wait (Y or N)?'
// IF ?64R?=N RETURN
// FILE NAME-TRANSACT,JOB-YES,DISP-SHRMM,WAIT-YES
// TAG OK
// LOAD PROG1
// RUN
// LOAD PROG2
// RUN
```

Example 5

This example shows how to use the DBLOCK parameter to specify that a larger number of records are to be transferred than was specified in the program.

```
// LOAD PAYROLL
// FILE NAME-PAYROLL,DISP-OLD,DBLOCK-30
// RUN
```

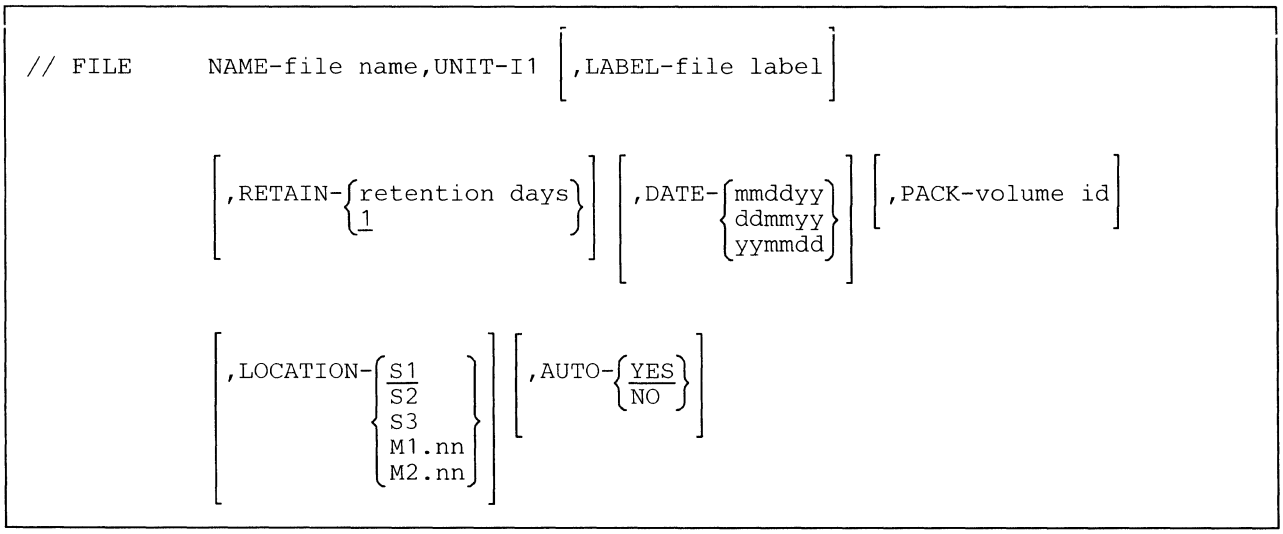
FILE OCL Statement (for Diskette Files)

The FILE OCL statement for diskette files supplies the SSP with information about a diskette file. The SSP uses this information to read from and write to the diskette. For more information about System/36 diskette file concepts, see the manual *Learning about Your Computer.*

The SSP utility programs that use FILE OCL statements for diskette files are the following:

Utility Program	Program Description	Procedures
$BICR	Basic data exchange	TRANSFER
$COPY	Disk file copy and display	COPYDATA, DISPLAY, LISTDATA, ORGANIZE, RESTORE, SAVE
$DUPRD	Diskette copy	COPYI1
$MAINT	Library maintenance	ALOCLIBR, FROMLIBR, RESTLIBR, SAVELIBR, TOLIBR
$POST	Special E format data exchange	POST

Placement: A FILE OCL statement is required for each diskette file that a program creates or uses. The FILE OCL statement for diskette files must follow the LOAD OCL statement and come before the RUN OCL statement.



S9020302-0

Note: The LOCATION and AUTO parameters are ignored for systems without a diskette magazine drive.

NAME specifies the name that the program uses to refer to the file. The file name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. However, question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, $, or @). The number of characters in a file name must not exceed eight.

UNIT-I1 specifies the file is on one or more diskettes. UNIT-I1 must be specified for a diskette file. If the UNIT parameter is omitted, UNIT-F1 is assumed.

LABEL specifies the actual name or label by which the file is identified on the diskette. If the LABEL parameter is omitted from a diskette FILE statement, the name specified by the NAME parameter is used. If the file is an existing file, a LABEL parameter is required when the name the program used to refer to the file differs from the name that identifies the file on the diskette.

The name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. However, question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The first character of a name must be alphabetic (A through Z, #, $, or @). The number of characters in a name must not exceed 17.

If more than one file on a diskette has the same name, those files must have different creation dates. In all cases (except when you are using the SAVE procedure or the $COPY utility program to save all files), the SSP displays a warning message before it will create a file with the same name as an existing file on the diskette. You can then (1) allow the SSP to create the file with a different creation date, (2) insert another diskette and then allow the SSP to create the file on that diskette, or (3) cancel the job. If you are using the SAVE procedure or the $COPY utility program to save **all** files, the creation date of each disk file becomes the creation date of the corresponding diskette file. Therefore, if more than one disk file exists with the same name, the diskette(s) will also contain those files with the same name and different creation dates. The SSP does not display a warning message in this case.

If more than one diskette file has the same name specified on a FILE statement and you do not specify the DATE parameter, the program processes the **first file** in the diskette VTOC with the specified name. Unlike the disk processing, there is no search for the file with the most recent date.

RETAIN specifies the number of days a file is to be retained. The RETAIN parameter is used to calculate an expiration date. Whenever RETAIN is specified for a file, the SSP determines the expiration date of the file by adding the job step date to the number of days specified by the RETAIN parameter. If the program date is invalid, an error message is displayed and you can cancel the job or continue processing. If you decide to continue, the expiration date is set equal to the creation date.

retention days can be any number from 0 through 999. If RETAIN is not specified when a new file is created, 1 day is assumed. If any number up to 998 is specified, the file is retained for the specified number of days. If 999 is specified, the file is considered permanent and can be deleted only by the DELETE procedure.

When the SSP creates a diskette file, the SSP writes the creation date and the calculated expiration date of the file in international format (yymmdd). If an existing diskette file that is not permanent is referenced by a FILE OCL statement with a RETAIN parameter, the expiration date of the file is changed to the date determined by the RETAIN parameter.

If an existing permanent diskette file is referenced by a FILE OCL statement with a nonpermanent RETAIN parameter, a message is displayed and you can either cancel the job or continue processing. If you decide to continue processing after the message is displayed, the file remains a permanent file.

Whenever the SSP is creating a file on a diskette or adding to an existing file on a diskette, all other files on the diskette whose expiration dates are the same as or earlier than the job step date and all expired files are deleted automatically. If a file being added to has expired, it is not deleted.

When the SSP checks whether a file has expired, each expiration date is checked for the international format (yymmdd) by comparing the first two characters to 70. If the first two characters are less than 70, it is assumed that the expiration date is not in the international format and is in the same format as the session date. (International format dates earlier than 1970 and expiration dates not in the international or session date formats, will not be read correctly by the SSP. This could cause the SSP to delete unexpired files or to retain expired files.)

When a new file is created on a diskette, the new file starts at the first available sector beyond the last unexpired file. The space that was occupied by a deleted file is not used again if any unexpired files follow it.

DATE specifies the creation date of an existing file and ensures that the proper version of a file is processed.

If you specify both a name and a creation date on the FILE OCL statement, the SSP displays a message if it cannot find a file with that name and date on the inserted diskette. You can then insert another diskette, and the SSP will look for the specified file on that diskette.

When a file is created on diskette, its name, expiration date, and creation date (job step date) are written on the diskette as identification. When a diskette file is created on the System/36, the SSP writes both the creation date and the expiration date in the international format (yymmdd). The SSP converts the job step date to the international format before writing it on the diskette. If the DATE parameter is specified for an existing diskette file, the SSP converts the specified date to international format before using the file. To ensure correct processing, files created by other systems should be created so that the creation date and the expiration date are written in the international format.

For information about the job step date, see the "DATE OCL Statement" on page 5-25.

PACK specifies the volume ID of the diskette, and is required when a program is creating a file or adding to a file on a diskette. The volume ID is put on the diskette by the INIT procedure and can be any combination of 6 or less alphameric characters.

The volume ID specified by the PACK parameter is compared with the volume ID of the diskette; if they are not the same, a message is displayed. You can then continue processing (the volume ID is ignored), insert the correct diskette, or cancel the job.

If the PACK parameter is not supplied on the diskette FILE statement for a diskette file that is being created or added to, a message is displayed and you must cancel the job.

The PACK parameter is not required for a diskette file that is being read; however, you should ensure that the proper diskette is inserted.

LOCATION specifies a diskette location.

S1, S2, or S3 identifies an individual diskette slot. If the file is on more than one diskette, the specified slot must contain the first diskette of the file. If LOCATION is not specified, S1 is assumed.

M1.nn or M2.nn identifies a magazine location. M1 indicates the first magazine; and M2 indicates the second magazine. nn is a number from 01 through 10 that identifies the location within the magazine. Specifying M1 or M2 is the same as specifying M1.01 or M2.01, respectively.

For output to diskettes, the first diskette to receive output must be in the specified location. If the diskette in that location is not capable of receiving the output, the system will not search the other slots for another diskette. Instead, an error message will be displayed and the operator will have to insert a diskette that is capable of receiving the output into the specified slot.

AUTO specifies how diskette files contained on more than one diskette are to be processed. You can use the ALLOCATE OCL statement to override this parameter; see the "ALLOCATE OCL Statement" on page 5-8 for more information.

YES specifies:

- If S1, S2, or S3 is specified in the LOCATION parameter, the program can use all three individual slots. Processing begins with the diskette in the slot specified on the LOCATION parameter. After the diskette in slot S3 has been processed, the SSP displays a message; the operator must then insert the next diskette. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified in the LOCATION parameter, the program can use both magazine slots. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes remain to be processed, the SSP displays a message; the operator must then insert the next magazine. Processing resumes at location M1.01.

NO specifies:

- If S1, S2, or S3 is specified in the LOCATION parameter, the program uses only the specified slot. After a diskette has been processed, the SSP displays a message; the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified in the LOCATION parameter, the program uses only the specified magazine. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes remain to be processed, the SSP displays a message; the operator must insert the next magazine in the same magazine slot. Processing resumes at the first location in the magazine.

Example

This example shows how FILE OCL statements are used by the $COPY utility program to copy a disk file to diskette. The name of the disk file is TRANS1, which is to be the name of the diskette file. The $COPY program refers to the input file (the disk file in this example) as COPYIN; it refers to the output file (the diskette file) as COPYO. The diskette has a volume ID of VOL001, and the diskette file is to be retained for 8 days.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-TRANS1
// FILE NAME-COPYO,UNIT-I1,LABEL-TRANS1,RETAIN-8,PACK-VOL001
// RUN
```

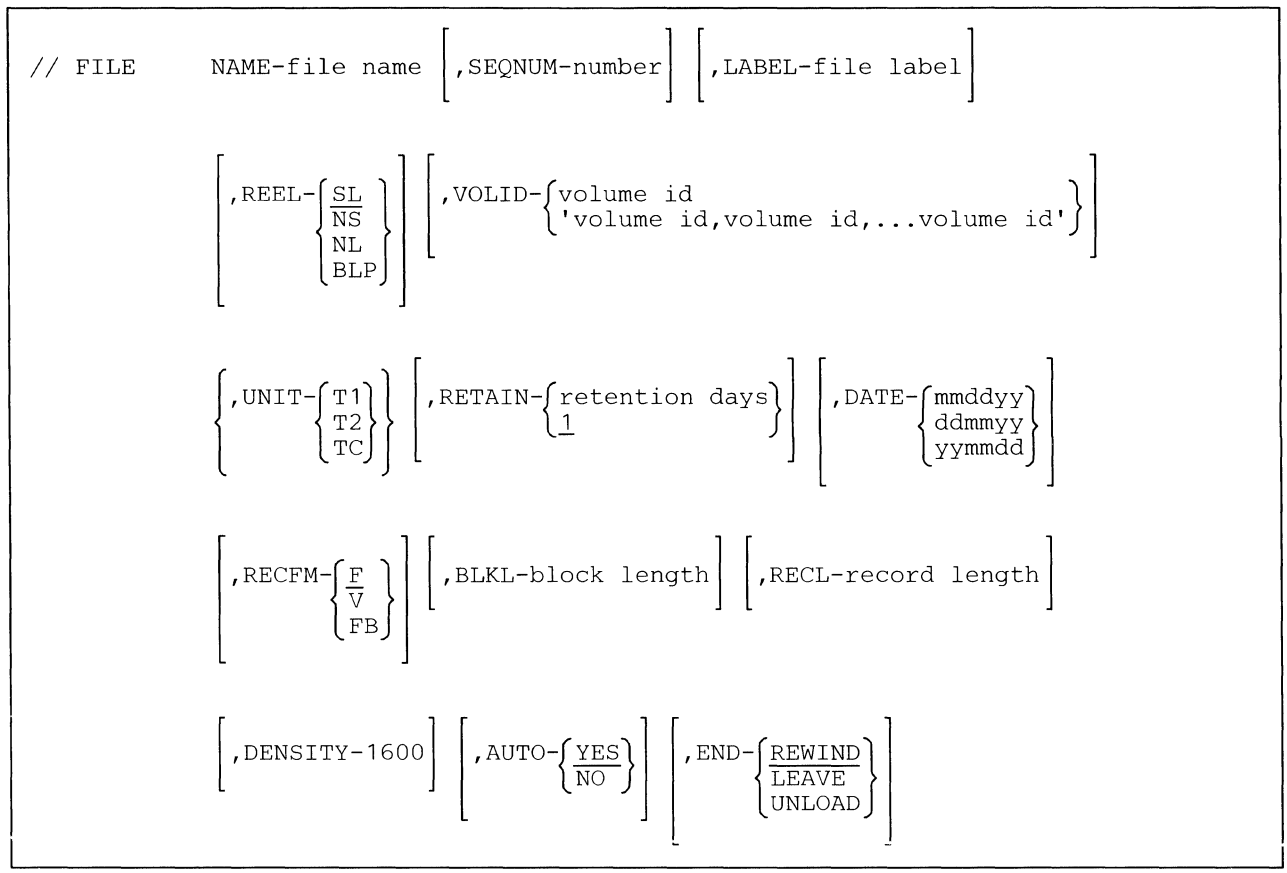
FILE OCL Statement (for Tape Files)

The FILE OCL statement for tape files supplies the SSP with information about a tape file. The SSP uses this information to read from and write to the tape. For more information about System/36 tape file concepts, see the manual *Learning about Your Computer* and the *Concepts and Programmer's Guide*.

The SSP utility programs that use FILE OCL statements for tape files are the following:

Utility Program	Program Description	Procedures
$TCOPY	Basic data exchange	TAPECOPY
$COPY	Disk file copy and display	COPYDATA, LISTDATA, LISTFILE, RESTORE, SAVE, SECSAVE, SECREST
$MAINT	Library maintenance	FROMLIBR, RESTLIBR, SAVELIBR, TOLIBR, BLDLIBR, JOBSTR

Placement: A FILE OCL statement is required for each tape file that a program creates or uses. The FILE OCL statement for tape files must follow the LOAD OCL statement and come before the RUN OCL statement.



S9020303-1

Note: The AUTO parameter is ignored for systems with only one tape drive.

NAME specifies the name that the program uses to refer to the file. The file name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. However, question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, $, or @). The number of characters in a file name must not exceed 8.

SEQNUM specifies the placement of the file when a tape contains more than one file. SEQNUM can be any number from 1-9999. For example, if the input file you want to use is the fifth file on the tape, you specify SEQNUM-5 to access the file. This parameter, as opposed to the LABEL parameter, is used mainly with nonstandard labeled and nonlabeled tapes, and tapes with bypass label processing.

If SEQNUM is specified and the LABEL parameter is also specified for an input file, the FILE OCL statement will process the SEQNUM parameter first. The header label name from the file found by the SEQNUM parameter is compared to the label name specified in the FILE statement. If the label names are the same, processing continues. If they are different, the FILE statement will then look for the file using the LABEL parameter.

If neither the SEQNUM parameter nor the LABEL parameter is specified, the file name is used as the label name.

LABEL specifies the actual name or label by which the file is identified on the tape. If the LABEL parameter is omitted from a tape FILE statement, the name specified by the NAME parameter is used. If the file is an existing file, a LABEL parameter is required when the name the program used to refer to the file differs from the name that identifies the file on the tape. The LABEL parameter is valid only with standard label tapes: if the REEL parameter is not REEL-SL, the LABEL parameter is ignored.

The name can be any combination of characters (numeric, alphabetic, and special) except commas (,), apostrophes ('), and blanks. However, question marks (?), slashes (/), equal signs (=), plus signs (+), greater than signs (>), and hyphens (-) should be used with caution because they have special meanings within procedures. The first character of a name must be alphabetic (A through Z, #, $, or @). The number of characters in a name must not exceed 17.

When two or more files on a tape have the same name, (except when you are using the SAVE procedure or the $COPY utility program to save all files), the SSP displays a warning message before it will create a file with the same name as an existing file on the tape. You can then (1) allow the SSP to create the file, (2) insert another tape and then allow the SSP to create the file on that tape, or (3) cancel the job. If you are using the SAVE procedure or the $COPY utility program to save **all** files, the creation date of each disk file becomes the creation date of the corresponding tape file. Therefore, if two or more disk files exist with the same name, the tape will also contain those files with the same name and different creation dates. The SSP does not display a warning message in this case.

If two or more tape files have the same label specified on a FILE statement and you do not specify the DATE parameter, the program processes the **first file** on the tape with the specified label. If two or more tape files have the same label and creation date, the program processes the first file on tape with the specified label and creation date. To process a subsequent tape file with the same label and creation date, specify the file placement with the SEQNUM parameter. Unlike disk processing, there is no search for the file with the most recent date.

REEL specifies the type of label processing to be performed for the tape.

SL specifies standard tape labels. If REEL is not specified, SL is assumed.

NS specifies nonstandard tape labels. If UNIT-TC is specified, REEL-NS is not valid.

NL specifies nonlabeled tape. If UNIT-TC is specified, REEL-NL is not valid.

BLP specifies that the label processing is to be bypassed, but the tape must have standard labels. If UNIT-TC is specified, REEL-BLP is not valid.

VOLID specifies the volume ID of the tape or tapes being processed. The VOLID parameter is valid only for tapes with standard labels (REEL-SL). The name can contain up to 6 alphameric characters (A to Z, 0 to 9, @, #, $).

volume id specifies the volume ID of the tape to be processed. If a multivolume output file is being processed and only one volume ID is specified in the VOLID parameter, the volume ID of the first tape processed must match the volume ID specified. The volume IDs of the second and subsequent tapes are not checked and do not have to match the specified volume ID. At least one volume ID must be specified for a multivolume output file.

If a multivolume input file is being processed and only one volume ID is specified in the VOLID parameter, the volume ID of the first tape processed must match the volume ID specified. The volume ID of the second and subsequent tapes does not have to match the volume ID of the first tape.

'volume id,volume id,...volume id' specifies that if a multivolume file is being processed and more than one volume ID is to be specified in the VOLID parameter, the volume IDs must be enclosed within single quotes (') and separated by a comma (,); for example, **'vol1,vol2,vol3'**. Up to 30 volume IDs can be specified. The volume ID of each tape processed must then match and be in the same order as the volume IDs specified.

UNIT specifies on which tape drive the processing is to begin. UNIT-T1, UNIT-T2, or UNIT-TC must be specified for a tape file. If the UNIT parameter is omitted, UNIT-F1 is assumed.

T1 specifies that tape processing is to begin on tape drive 1.

T2 specifies that tape processing is to begin on tape drive 2.

TC specifies that tape processing is to be on the tape cartridge.

RETAIN specifies the number of days a file is to be retained. The RETAIN parameter is used to calculate an expiration date for standard label tapes only, and once written to tape it cannot be changed. Whenever RETAIN is specified for a file, the SSP determines the expiration date of the file by adding the job step date to the number of days specified by the RETAIN parameter. If the program date is invalid, an error message is displayed and you can cancel the job or continue processing. If you decide to continue, the system adjusts the creation date to a valid Julian date, and then adds the number of retention days you specify to the valid Julian date to equal an expiration date.

retention days can be any number from 0 through 999. If RETAIN is not specified when a new file is created, 1 day is assumed. If any number up to 998 is specified, the file is retained for the specified number of days. If 999 is specified, the file is considered permanent.

When the SSP creates a tape file, the SSP writes the creation date and the calculated expiration date of the file in Julian format (cyyddd). If an existing tape file is referenced by a FILE OCL statement with a RETAIN parameter, the RETAIN parameter is ignored.

The expiration date is not checked when the tape file is allocated, but only when a tape is initialized (by the TAPEINIT procedure) and when writing between tape volumes. Only the first file on the tape is checked against the expiration date.

When a new file is created on a tape, the new file is placed after the last file on the tape.

DATE specifies the creation date of an existing file and ensures that the proper version of a file is processed.

If you specify both a name and a creation date on the FILE OCL statement, the SSP displays a message if it cannot find a file with that name and date on the inserted tape. You can then mount another tape, and the SSP will look for the specified file on that tape.

When a file is created on tape, its name, expiration date, and creation date (job step date) are written on the tape as identification. When a tape file is created on the System/36, the SSP writes both the creation date and the expiration date in the Julian format (cyyddd). The SSP converts the job step date to the Julian format before writing it on the tape. If the DATE parameter is specified for an existing tape file, the SSP converts the Julian date to the session date format before using the file. To ensure correct processing, files created by other systems should be created so that the creation date and the expiration date are written in the Julian format.

For information about the job step date, see the "DATE OCL Statement" on page 5-25.

RECFM specifies the format of the input and output file records. If the RECFM parameter is not specified, F is assumed unless you are reading or adding to a standard label tape file (REEL-SL) and the file being processed contains a HDR2 label.

F specifies the format to be fixed length, unblocked records. If UNIT-TC is specified, RECFM-F is not valid.

V specifies the format to be variable length, unblocked records (valid only for input files). If UNIT-TC is specified, RECFM-V is not valid.

FB specifies the format to be fixed length, blocked records.

BLKL-block length specifies the number of bytes in a physical block of data on tape. The parameter can be any value from 18 bytes to 32767 bytes. The BLKL parameter can only be used for fixed length blocked records (the RECFM-FB parameter is specified). The BLKL parameter is required for all output files that have fixed length blocked records and for all input files that have fixed length blocked records not being processed by standard labels with an HDR2 label (that is REEL-NS, REEL-NL, REEL-BLP, or REEL-SL and the file being processed does not have an HDR2 label is specified).

RECL-record length specifies the number of bytes in a logical data record on tape. The parameter can be any value from 18 bytes to 4096 bytes. The RECL parameter is required for input files not being processed by standard labels with an HDR2 label (that is REEL-NS, REEL-NL, REEL-BLP, or REEL-SL and the file being processed does not have an HDR2 label is specified). If the RECL parameter is specified for a standard label tape (REEL-SL is specified and the file being processed has an HDR2 label), it must match the record length contained within the tape label. If it does not match, an error message is issued.

DENSITY-1600 specifies the number of bytes per inch at which files are to be written or read from tape. If UNIT-TC is specified, DENSITY-1600 is not valid.

AUTO specifies how tape files contained on more than one tape are to be processed. You can use the ALLOCATE OCL statement to override this parameter; see the "ALLOCATE OCL Statement" on page 5-8 for more information.

YES specifies that if you run out of tape while processing a file, an attempt is made to continue processing the file by allocating the other tape drive. If that tape drive cannot be allocated, the original tape drive will be used for the next volume. If the AUTO parameter is not specified, AUTO-YES is assumed.

NO specifies that an attempt should not be made to continue a file on the next tape drive.

END specifies the position of the tape after the file has been processed.

REWIND specifies that the tape is to be rewound after the file is processed. If the END parameter is not specified, REWIND is assumed.

LEAVE specifies that the tape is to remain in the position of the last access.

When a job step using tape is successfully completed and LEAVE was specified, the next job step using tape may then take advantage of the position of the tape:

- If the next job step is writing a new standard labeled (SL) tape file, the system checks to ensure that the tape is positioned at the end of the tape prior to writing the data to tape. If a sequence number is specified, the system searches from its current position toward the specified sequence number, validating that the sequence number can be written on this tape. If a tape file exists at that sequence number, it will be written over by the new tape file.

  Note: An attempt to overwrite files on the 6157 Tape Drive causes a diagnostic message to be issued.

- If the next job step is reading a standard labeled (SL) tape file, the system validates that it is positioned to the correct file by checking the file label and the creation date (if specified). If the file label and creation date do not match, the system rewinds and searches the tape for the specified file. If a sequence number is specified, the system searches from its current position to the specified sequence number, performing validation of the file label and the creation date prior to reading the data.

  Note: If the sequence number requested for the 6157 Tape Drive is less than the current position, the tape is rewound and searched from the beginning of the tape. If the sequence number requested is greater than the current position, the tape is searched in a forward direction from its current position.

- If the next job step is reading a standard labeled (SL) tape file in bypass label processing (BLP) mode, the system searches from its current position to the specified sequence number.

- If the next job step is reading or writing to a nonlabeled (NL) tape file and no sequence number has been specified, the system begins reading or writing data with no checking. If a sequence number is specified, the tape is rewound. Then the sequence number is located.

- If the next job step is reading a nonstandard labeled (NS) tape file, the system rewinds and reads the first file's data.

The PCE expressions of VOLID and DATAT cause the tape to rewind and search, starting from the beginning of the tape.

UNLOAD specifies that the tape is to be rewound and unloaded after the file is processed.

Example

This example shows how FILE OCL statements are used by the $COPY utility program to copy a disk file to tape on tape drive 1. The name of the disk file is TRANS1, which is to be the name of the tape file. The $COPY program refers to the input file (the disk file in this example) as COPYIN; it refers to the output file (the tape file) as COPYO. The tape has a volume ID of VOL001, is to be processed as a standard label tape in the format of fixed length, blocked records, with a block length of 24576 bytes. If the file runs out of tape, the other tape drive should not be used to finish copying the file. After the file has been copied, the tape should be rewound and unloaded for removal. The tape file is to be retained for 8 days.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-TRANS1
// FILE NAME-COPYO,LABEL-TRANS1,REEL-SL,VOLID-VOL001,+
   UNIT-T1,RETAIN-8,RECL-256,RECFM-FB,BLKL-24576,DENSITY-1600,+
   AUTO-NO,END-UNLOAD
// RUN
```

FORMS OCL Statement

The FORMS OCL statement specifies information about the printer to be used during a display station session. The FORMS OCL statement could be used to set how output is to be printed for jobs that contains several printing steps, without having to specify the same information several times on PRINTER OCL statements.

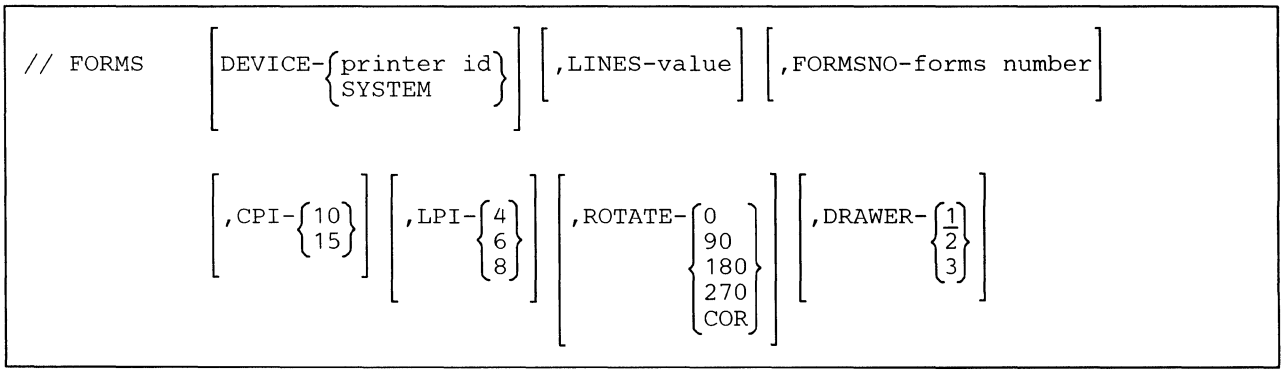
- The printer ID of the printer to be used for all printed output
- The number of lines to be printed per page
- The forms number
- The horizontal print density (characters per inch)
- The vertical print density (lines per inch)
- The orientation or size of the printed output on the page (rotation or reduction)
- The printer drawer from which paper is to be used

Changes made by the FORMS OCL statement remain in effect until the display station session ends or they are changed by:

- Another FORMS statement
- The PRINTER statement (for that job step only)
- The LINES procedure
- The PRINT procedure

A job placed on the job queue uses the values that were in effect when the job was placed on the queue. However, if a procedure running from the job queue contains a FORMS OCL statement, the FORMS OCL statement in the procedure is used.

Placement: The FORMS OCL statement can appear anywhere among the OCL statements.



S9020304-1

If the FORMS statement is used, at least one parameter must be specified.

DEVICE specifies the printer to be used for printed output. System list output, Print key output, and any other printed output from the session is all printed on the specified printer.

If no DEVICE parameter is specified, the printer assignments are not changed. The DEVICE parameter of the PRINTER OCL statement can override this parameter.

printer id specifies the work station ID of the printer to be used. You can use the STATUS WORKSTN command to determine the printer IDs.

SYSTEM specifies that the system printer is to be used.

LINES specifies the number of print lines per page. The maximum number of lines that can be specified per page is 112. If the LINES parameter is not specified and the number of lines per page was not previously set during the session, the system uses the value specified for the display station during system configuration, by the SET procedure, or by the $SETCF utility program. If a line counter specification is used in a program, that specification remains in effect only for the duration of that program.

For IBM-supplied programs and most user-written programs, the printer skips to a new page when six less than the number of lines specified are printed. For example, if LINES-66 is specified, the printer skips to a new page after printing line 60. Also, the printing starts on line 6 of the new page. Therefore, if you specify 66 lines per page, you actually get 54 lines of output per page (66 - 12 = 54). If LINES-13 is specified, one line is printed per page. When 12 or less lines are specified, printing occurs on every line of each page.

For print operations from your programs, the SSP indicates an overflow condition when six less than the number of lines specified (either in the program or in the FORMS statement) are printed.

FORMSNO specifies the forms number of the printer forms to be used for printed output from the display station session. (Each type of form should have a unique, user-assigned forms number.) The forms number can be any combination of up to 4 characters except commas (,), apostrophes ('), and blanks. However, question marks (?), slashes (/), equal signs (=), plus signs (+), and hyphens (-) should be used with caution because they have special meanings within procedures.

If a forms number is specified, the SSP prompts the operator controlling the printer to install the forms with the specified forms number in the printer if the specified forms are not already installed.

CPI specifies the horizontal print density, that is, characters per inch, to use for printed output from the display station session. The values that can be specified are 10 or 15. If CPI-15 is used, the output should be printed on a 4214, 4224, 4234, 5225, or 5224 Printer. If CPI-15 is attempted on another printer, a message is displayed and the operator controlling the printer can either cancel or continue the printing. CPI is ignored if the output is printed on a 5219 Printer.

The PRINTER statement can be used to change the CPI value for a particular job step, but it stays in effect only for that job step. If the CPI parameter is not specified and the characters per inch was not previously set during the session, the system uses the value that was set when the printer was configured (either 10 or 15). (Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is actually printed, not when it is intercepted in the spool file.) If no CPI value was specified when the printer was configured, the CPI value is controlled by a switch on the printer.

LPI specifies the vertical print density (that is, lines per inch) to use for printed output from the display station session. The values that can be specified are 4, 6, and 8. If the output is printed on a printer other than the 5224, 5225, 4214, 4224, 4234, or 4245 Printer, the LPI parameter is ignored. If LPI-4 is specified on a 4245 printer, it actually prints at LPI-6.

The PRINTER statement can be used to change the LPI value for a particular job step, but it stays in effect only for that job step. If the LPI parameter is not specified and the lines per inch was not previously set during the session, the system uses the value that was set when the printer was configured (either 4, 6, or 8). (Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is actually printed, not when it is intercepted in the spool file.) If no LPI value was specified when the printer was configured, the LPI value is controlled by a switch on the printer.

ROTATE specifies that the output is to be rotated on the page or that the size of printed output is to be reduced. This parameter is valid only if output is printed on a 3812 Printer.

0 specifies that printed output is to be rotated zero degrees on the page.

90 specifies that printed output is to be rotated 90 degrees on the page.

180 specifies that printed output is to be rotated 180 degrees on the page.

270 specifies that printed output is to be rotated 270 degrees on the page.

COR (computer output reduction) specifies that the size of the printed output is to be reduced. Output normally printed on 14-inch wide paper will print on 8-1/2 x 11 inch paper.

DRAWER specifies that the paper used by a printer is to be selected from the specified drawer. The 3812 SNA Control Stream (SCS) and Intelligent Printer Data Stream (IPDS) Printers have two paper drawers, and the 4214 and 5219 Printers have three paper drawers.

1 specifies that output is to be printed on paper from drawer 1. If the DRAWER parameter is not specified, 1 is assumed.

2 specifies that output is to be printed on paper from drawer 2.

3 specifies that output is to be printed on paper from drawer 3. Specify drawer 3 if you want to select envelopes on the 5219 Printer.

Example 1

The FORMS statement in the following procedure specifies that the forms length for the job is 20 lines per page and that form CHEC is to be used.

```
// FORMS LINES-20,FORMSNO-CHEC
// LOAD PRNTCHEC
// PRINTER NAME-CHECKS,SPOOL-NO,ALIGN-YES
// RUN
```

Example 2

The following FORMS statement specifies:

- The printer to be used has a work station ID of P3 (the printer happens to be a 5225 Printer).
- The vertical print density is to be 8 lines per inch.
- The length of the forms is 11 inches (27.9 cm). Therefore, the lines per page is to be 88 (11 x 8 = 88).
- The horizontal print density is to be 15 characters per inch.

The programs to be run are named TEST and TEST1.

```
// FORMS DEVICE-P3,LINES-88,LPI-8,CPI-15
// LOAD TEST
// RUN
// LOAD TEST1
// RUN
```

IMAGE OCL Statement

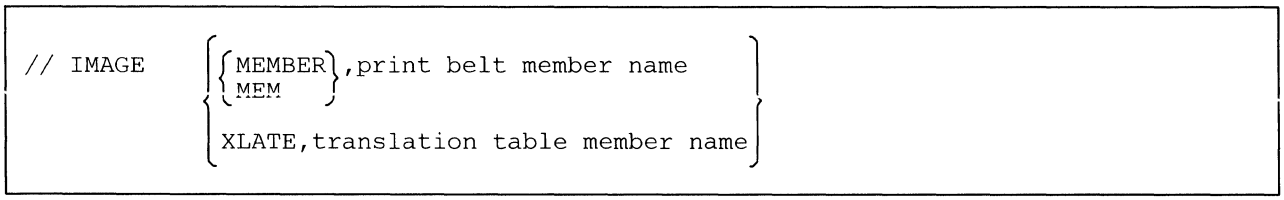
The IMAGE OCL statement specifies the print belt image to be used for all output to the 3262 Printer from the requesting display station. The IMAGE statement affects only data for the 3262 Printer. Data to be printed by other printers on the system is not affected.

You can also use the IMAGE OCL statement to translate lowercase characters to uppercase characters for those print belts that cannot print lowercase characters.

For each display station, a special table called the **printer specification table** contains the name of the print image member to be used when output for the display station is printed. That image name can be changed at any time by the IMAGE statement. When output from the display station is directed to a 3262 Printer, the SSP checks that the print belt matches the image name in the printer specification table. The SSP comes with several print belt images; if none of these images can be used for your print belt, you will have to create your own print belt image, see "Creating Your Own Print Belt Members" on page 5-61. If the belt does not match, the SSP displays a message instructing the system operator to change the belt.

The IMAGE statement allows you to specify a 3262 Printer translation table for each display station or job on the job queue. When a job is started from a display station or the job queue, it will use the system translation table unless one is specified by an IMAGE OCL statement.

Placement: The IMAGE OCL statement can appear anywhere among the OCL statements.



S9020305-0

MEMBER or MEM indicates the new print belt image member to be used. The member must be in a system library (#LIBRARY) source member.

print belt member name specifies the source member that contains the new print belt image. The SSP includes the following print belt members:

BELT48
BELT48HN
BELT64B
BELT64C
BELT96
BELT188B

For information about the characters included in these print belts, see "3262 Print Belts" on page E-1.

XLATE specifies that a translation table will be replaced. When you are printing information that includes lowercase characters using a print belt other than BELT96 or BELT188B, translation tables allow you to have all lowercase characters translated to uppercase characters.

translation table member name specifies the source member in the system library that contains a translation table. The SSP includes the following translation tables:

#96E48 Translates the 96-character set to a 48-character print belt.

#96E64 Translates the 96-character set to a 64-character print belt.

#188E48 Translates the 188-character set to a 48-character print belt.

#188E64 Translates the 188-character set to a 64-character print belt.

#188E96 Translates the 188-character set to a 96-character print belt.

#188E188 Resets the character translation.

For information about the characters translated in these tables, see "3262 Translation Tables" on page E-7. To turn off character translation, specify translation table #188E188.

To specify a translation table of your own design, the first record within the source member must be in the form:

```
pp,nnn
```

pp is a 2-character representation of 1 hexadecimal byte specifying the position of the first character to be replaced in the translation table. pp must be between hex 40 and FE.

nnn is a 1- to 3-character number that specifies the number of characters being supplied as translation table data. The allowable range for this value is 2 through 384.

The second and following source statements must contain 2-character representations of each hexadecimal byte to be replaced in the translation table. The number of characters must be equal to the value specified in nnn, and each representation must have a value greater than or equal to hex 40 and less than or equal to hex FE. Except for the last record, the data must completely fill the record length specified when the translation table was created.

Example 1

The print belt member for the BELT64C print belt is in a system library source member called BELT64C.

```
// IMAGE MEMBER,BELT64C
```

Example 2

Specify a print belt of BELT64C and a translation table of #96E64. This will cause your output to be printed using a 64-character print belt; and any lowercase characters are translated and printed as uppercase characters.

```
// IMAGE MEMBER,BELT64C
// IMAGE XLATE,#96E64
```

Example 3

Specify a translation table using source member ALPHA.

```
// IMAGE XLATE,ALPHA
```

The source member ALPHA would contain the following lines:

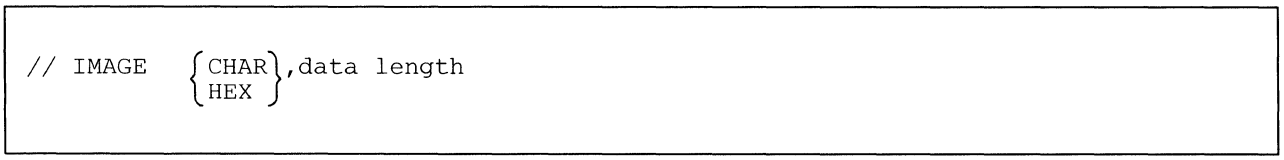
```
C1,6
E7E8E9
```

C1 (the hexadecimal value for the letter A) indicates the first character to be replaced, and 6 specifies the number of characters being supplied. As a result, the characters A, B, and C are changed to X, Y, and Z (hex E7, E8, and E9 respectively).

Creating Your Own Print Belt Members

If you need to use a print belt that does not have an IBM-supplied print belt member, you will have to create your own print belt source member. The source member consists of a special form of the IMAGE statement (called an IMAGE member statement), and one or more lines that indicate the characters on your print belt.

The IMAGE member statement has the following format:



S9020306-0

CHAR indicates the one or more lines that follow the IMAGE statement are in character form.

HEX indicates the one or more lines that follow the IMAGE statement are in hexadecimal form. See "3262 Print Belts" on page E-1 for the hexadecimal forms of the characters.

If the print belt contains characters that cannot be entered from the keyboard, HEX must be specified.

data length specifies the number of characters to be entered. (If HEX is specified for the first parameter, each character on the print belt requires 2 hexadecimal numbers to be entered.) For the IBM 3262 Printer, the data length must not exceed 288 when characters are entered and must not exceed 576 when hexadecimal numbers are entered.

The one or more lines of characters have the following format:

- The characters must begin in the first position.
- Consecutive positions must be used, and the characters must be entered in the sequence they appear on the print belt. For example, the sequence of characters on a sample 50-character print belt could be the following (this information is included with your print belt):

  ```
  1234567890#@/STUVWXYZ&,JKLMNOPQR-$*ABCDEFGHI+.'?=>
  ```

- All positions in a line must contain characters before the characters can be continued on a following line.

For example, to create a print belt member for the sample 50-character print belt shown above, you could create the following print belt member in either character form:

```
// IMAGE CHAR,50
1234567890#@/STUVWXYZ&,JKLMNOPQR-$*ABCDEFGHI+.'?=>
```

or in hexadecimal form:

```
// IMAGE HEX,100
F1F2F3F4F5F6F7F8F9F07B7C61E2E3E4E5E6E7E8E9506BD1D2D3D4D5D6D7D8D9
605B5CC1C2C3C4C5C6C7C8C94E4B7D6FBF6E
```

To actually create the member, see "$MAINT Utility" on page A-55 or the *SEU Guide*.

INCLUDE OCL Statement

The INCLUDE OCL statement identifies a procedure member containing OCL and utility control statements to be placed into the job stream. If the procedure is not a multiple requester terminal (MRT) procedure or if PDATA-YES was not specified when the procedure member was created, the INCLUDE statement can pass **parameters** to the procedure.

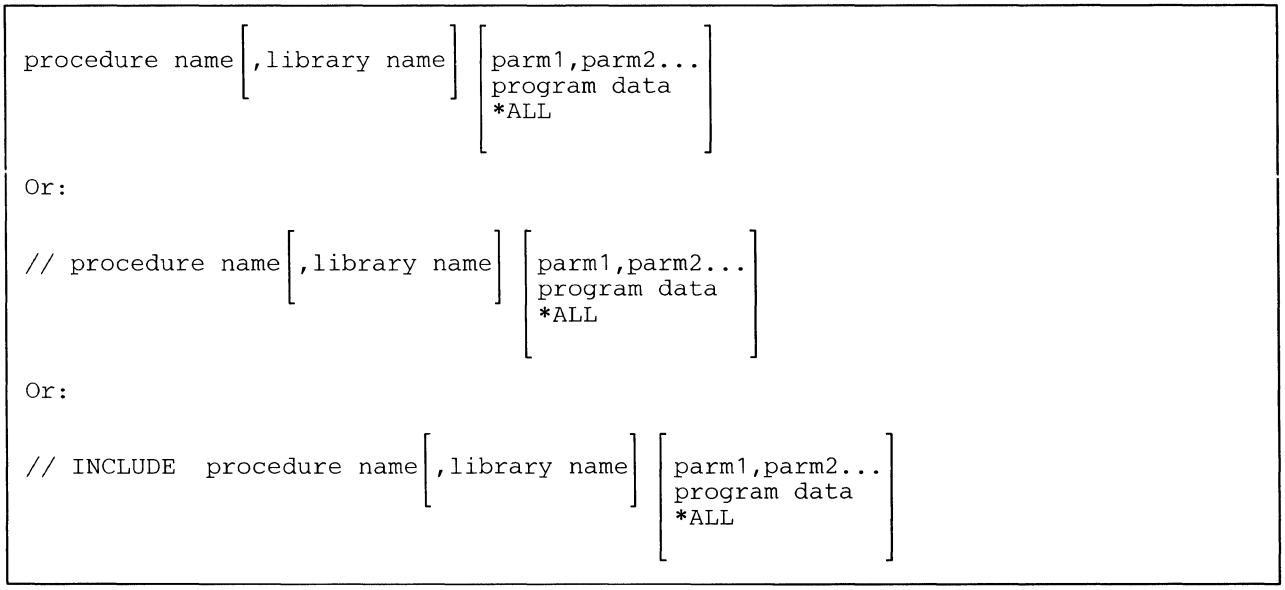
If the procedure is an MRT procedure or if PDATA-YES was specified when the procedure member was created, the INCLUDE statement can pass only **data** to a program.

For general information and programming considerations about MRT programs and procedures, see the *Concepts and Programmer's Guide*. For information about creating procedures, see Chapter 2, "Making Your Own Procedures" on page 2-1.

The first form of the INCLUDE statement should be used if the procedure name is the same as an OCL statement identifier. For example, if the procedure name is FILE, the following format is correct:

```
// INCLUDE FILE FILEA,FILEB
```

Placement: The INCLUDE statement can be placed anywhere among the OCL statements.



S9020307-0

procedure name specifies the procedure member to be called.

library name specifies the library to be searched for the procedure. If a library name is specified, that library is searched and then the system library (#LIBRARY) is searched. If a library name is not entered, the current library is searched and then #LIBRARY is searched.

parm1,parm2, ... specifies parameters for the procedure. Parameters are not allowed if the procedure is an MRT procedure or if PDATA-YES was specified when the procedure member was created. The parameters may or may not be required, depending on the procedure they are passed to.

The number of characters in a parameter must not exceed 128. A maximum of 64 parameters, separated by commas, can be passed with an INCLUDE statement. See "Procedure Parameters" on page 5-3 for more information on procedure parameters.

When you are entering parameters, you can type in up to 512 characters. For example, you can type in 32 sixteen-character parameters or 64 eight-character parameters. However, the combined total length of **all** parameters cannot exceed 1024 characters; this length can be accomplished by using substitution expressions and the local data area. See "Continuing the Lines of a Procedure" on page 2-7 for information about how to continue input lines to get more characters than 120.

program data specifies data, not parameters, to be passed at the first read operation in the program. The data starts with the first nonblank character following the procedure name and ends with the last nonblank character in the statement. The data is passed to the program on its first input operation from the display station. (The input record in the program would contain this data.) Up to 508 characters can be specified using continuation. See "Continuing the Lines of a Procedure" on page 2-7 for information on how to continue input lines for more than 120 characters.

***ALL** specifies that all 64 parameters are to be passed from the current procedure level to the procedure being called. *ALL can only be specified within a procedure. If *ALL is specified as the only parameter and it is entered from the keyboard or selected by a menu item, an error message is displayed. If *ALL is specified as one of the parameters, it is treated as a single parameter.

Example 1

In this example, ACCTS and EMPNUM are two parameters that are interpreted by the PAYROLL procedure. Parameter 2 is omitted.

```
PAYROLL ACCTS,,EMPNUM
```

Example 2

In this example, MRTPROC is an MRT procedure that causes an MRT program to be run. The number 126 is data passed to the MRT program on its first input operation from the requesting display station.

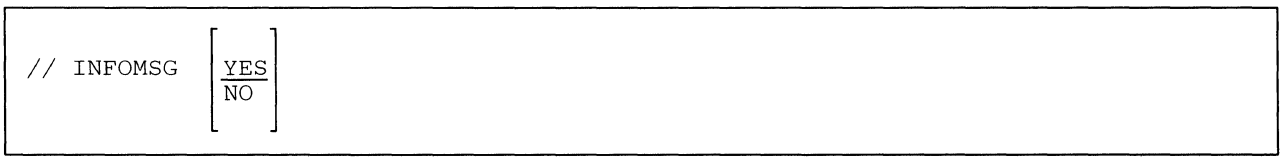
```
MRTPROC 126
```

INFOMSG OCL Statement

The INFOMSG OCL statement specifies whether informational messages are to be displayed. For example, you might create a procedure that uses several system procedures. Informational messages are all messages that do not require a response. Most system procedures issue messages stating that they are running, and such messages might be confusing to some operators who are running application programs.

Also, an informational message sent to a remote display station results in longer response times because the current display is saved, and then shown again when the procedure ends. By preventing the informational messages from being displayed, you can decrease the time needed to run the procedure. When the INFOMSG statement is specified in a procedure, it remains in effect until another INFOMSG statement is processed or until the procedure ends. When the INFOMSG statement is entered from the keyboard, it remains in effect until another INFOMSG statement is entered or until the operator signs off.

Placement: The INFOMSG statement can be placed anywhere among the OCL statements. The INFOMSG OCL statement cannot be used in an interactive communication feature session, in a job running from the job queue, or in a job that was evoked.



S9020308-0

YES specifies that informational messages are to be displayed. If no parameter is specified, YES is assumed.

NO specifies that informational messages are not to be displayed.

Example

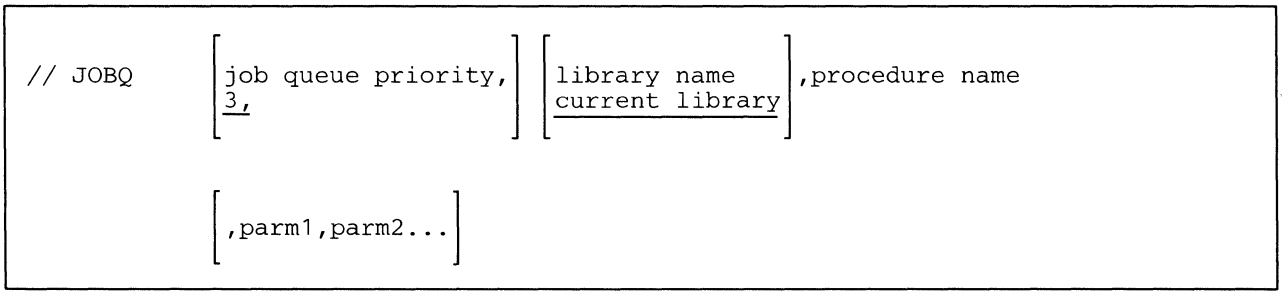
This example shows how you can display a general informational message, prevent the messages issued by the SAVE procedures from being displayed, and then display a message at the end of the procedure.

```
// * 'Saving files FILE1, FILE2, and FILE3'
// INFOMSG NO
SAVE FILE1,,VOL001
SAVE FILE2,,VOL001
SAVE FILE3,,VOL001
// INFOMSG YES
// * 'Save complete'
```

JOBQ OCL Statement

The JOBQ OCL statement places a job on the job queue. The JOBQ OCL statement may be most useful for jobs that require no operator input or response, or for jobs whose output is not required by a subsequent job step. The maximum number of characters you can enter on the JOBQ statement is 120.

Placement: The JOBQ statement can appear anywhere among the OCL statements.



S9020309-1

job queue priority specifies job queue priority; that is, the job's order of processing from the job queue. The job queue priority can be any number from 0 through 5. When choosing the next job to run, the system considers jobs with higher priority numbers before jobs with lower priority numbers. For example, all jobs with a job queue priority of 5 are considered before any other jobs in the job queue. Jobs with the same job queue priority are considered in the order they were placed in the job queue. Jobs with a job queue priority of 0 are the last jobs considered by the system. Job queue priority 0 is usually stopped, that is, any jobs placed on the job queue with a priority of 0 will not be considered until the system operator starts priority 0.

If no parameter is specified, a job queue priority of 3 is assigned to the job, and the comma shown in this parameter should not be specified.

library name specifies the library for the job. The system searches the specified library and then the system library for the procedures, load members, message members, and display formats used in this job. If no parameter is specified, the current library is assumed.

procedure name specifies the procedure that is to be placed on the job queue.

parm1,parm2 . . . specifies the parameters required by the procedure. You can specify at most 120 characters using the JOBQ OCL statement.

Example

Place the PAYROLL procedure, which is located in library PAYLIB, on the job queue with a job queue priority of 4.

```
// JOBQ 4,PAYLIB,PAYROLL
```

LIBRARY OCL Statement

The LIBRARY OCL statement specifies the name of the current library for the display station session or for the duration of a procedure. The current library is the library that the SSP searches first for programs specified on LOAD OCL statements, procedures, menus, message members, and display formats. If the SSP does not find the member in the current library, the SSP then automatically searches the system library (#LIBRARY). However, the search order for program and display formats can be altered by the library name parameter on the LOAD OCL statement. See the LOAD OCL statement.

A library remains current until another LIBRARY OCL statement is processed or until the display station session ends. If the LIBRARY statement is used in a procedure (and SESSION-YES is **not** specified), the current library is changed only while that procedure runs. When the procedure ends, the library that was current when the procedure began is again current.

Once a library is specified as current (when the operator signs on to the system or when a LIBRARY OCL statement is entered from the keyboard), that library remains allocated to the display station at least until a different current library or NAME-0 is specified.

If a different current library or NAME-0 is specified while the display station is still using one or more members from the previous current library (for example, using a menu from that library), the previous current library remains allocated to the display station until the display station is no longer using members from that library. Consequently, certain functions, such as the CONDENSE procedure, cannot be done on the library even though it is no longer the current library at the display station.

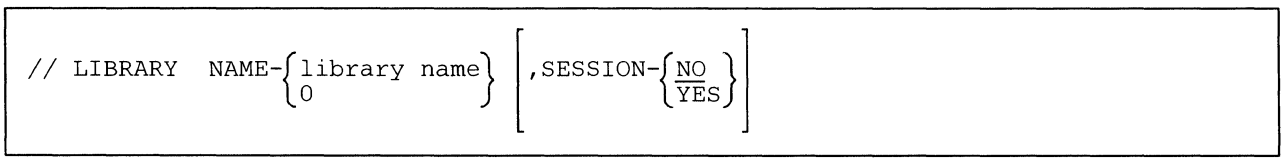
When a procedure is interrupted, the library for the session is current during the inquiry request. The library that was current for the interrupted procedure will again be current when the procedure is resumed. For example, a procedure contains the following statement:

```
// LIBRARY NAME-JOBLIB
```

If a program within the procedure is interrupted, the current library for the session (not JOBLIB) will be current during the inquiry request. For information about interrupting a program, see the manual *Operating Your System* for your system unit.

If a LIBRARY OCL statement is entered from a MENU display, one or more of the items on the displayed menu might not be correct because the procedures corresponding to those items do not exist in the new current library.

Placement: The LIBRARY OCL statement can appear anywhere among the OCL statements except between a LOAD and RUN statement pair.



S9020310-0

NAME specifies the library that will be the current library. If NAME-0 is specified, only the system library (#LIBRARY) is searched by the system.

SESSION specifies whether the session library is to be changed.

NO causes the current library to be changed. If no parameter is specified, NO is assumed. The change remains in effect until the job ends.

YES causes the session library to be changed to the library specified in the NAME parameter (the session library is the library in effect at the keyboard). The change takes effect after the current job ends. (It lasts until changed again or until the session ends.) If the OCL statement is entered from the keyboard, the current library is the session library; therefore, the SESSION parameter has no effect.

Example 1

The following statement specifies that a library called MYLIB is the current library.

```
// LIBRARY NAME-MYLIB
```

Example 2

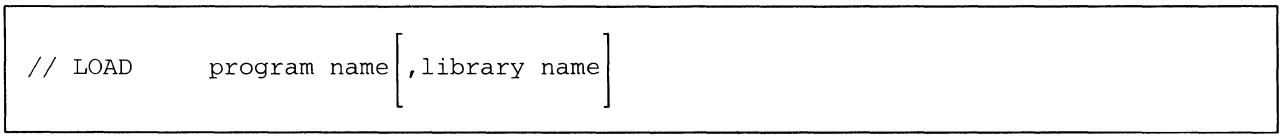
The LIBRARY OCL statement in the following procedure specifies that a library called PAYLIB is the current library for the procedure. The program named PAYROLL and the message member named PAYMSG are contained in that library.

```
// LIBRARY NAME-PAYLIB
// MEMBER USER1-PAYMSG
// LOAD PAYROLL
// FILE NAME-PAYFILE
// RUN
```

LOAD OCL Statement

The LOAD OCL statement identifies a program to be run. The LOAD statement is the first in a set of statements that defines a job step.

Placement: Two LOAD OCL statements cannot be entered without an intervening RUN OCL statement.



S9020311-0

program name specifies the program to be loaded. The program named is a library load member.

library name specifies the library to be searched for the program and display formats. If a library name is specified, that library is searched and then the system library (#LIBRARY) is searched. If a library name is not entered, the current library is searched and then #LIBRARY is searched.

Example 1

In the following LOAD statement, PAYROLL identifies a payroll program.

```
// LOAD PAYROLL
```

Example 2

The following example shows a sample procedure, stored in a library named MYLIB, that runs a program named MYPROG. The program displays messages from a message member named MESSAGES, which is stored in a library named COMMLIB.

```
// MEMBER USER1-MESSAGES,LIBRARY-COMMLIB
// LOAD MYPROG,MYLIB
// RUN
```

Example 3

The following example shows a procedure that runs a program named TEST. The program displays messages from a message member named TESTMSG. The message member and the program are both stored in a library named TESTLIB.

```
// LIBRARY NAME-TESTLIB
// MEMBER USER1-TESTMSG
// LOAD TEST
// FILE NAME-TEMP,LABEL-TEMP?WS?,RETAIN-J,BLOCKS-30
// RUN
```

LOCAL OCL Statement

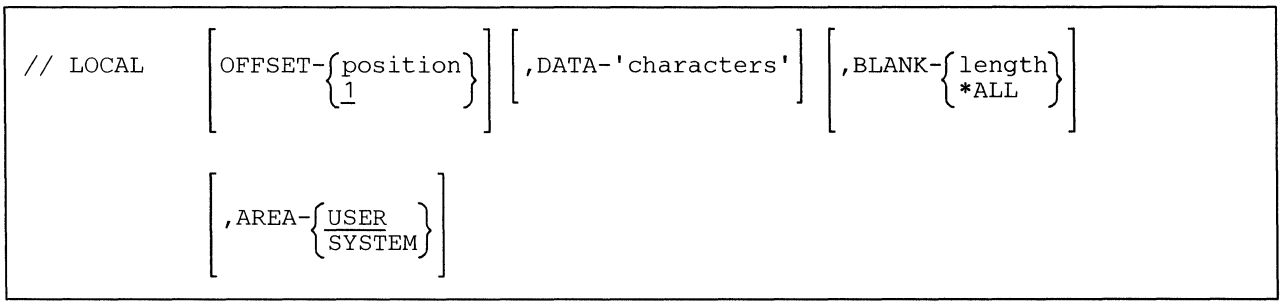
The LOCAL OCL statement is used to modify a specified area in the display station local data area. You can use this statement to pass information between procedures, programs, job steps, and jobs.

A display station local data area exists for each command display station and each running MRT procedure. Each display station local data area is a 512-byte area on disk that can be used to pass information between jobs, job steps, procedures, and programs run during a display station session. For information about using data from the display station local data area to modify a procedure, refer to "?L'position,length'? (Local Data Area) Expression" on page 3-19.

The SSP automatically sets the local data area to blanks at the beginning of a session, that is, when the operator signs on to the system. The local data area is available for use by all jobs run during the session. However, a job placed on the job queue or a released job step uses a copy of the local data area as it was when the job was placed on the queue or when the job was released.

If inquiry mode is used (that is, the operator presses the Attn key and takes option 1), the contents of the local data area are saved and then restored when inquiry mode ends.

Placement: The LOCAL statement can appear anywhere among the OCL statements.



S9020312-0

Although each parameter is optional, either the DATA, BLANK, or AREA parameter must be specified.

OFFSET specifies the first position in the display station local data area to be changed.

position can be any number from 1 through 512. If a position is not entered, 1 is assumed.

DATA specifies the data that is to be placed in the display station local data area.

'characters' must be enclosed in apostrophes ('). If the characters contain embedded apostrophes (such as the apostrophe in o'clock), then enter the embedded apostrophe as two apostrophes (for example: 'o''clock'). The number of characters is limited by the maximum length of the OCL statement. IGC characters are allowed.

BLANK specifies the number of positions, starting with the position specified by OFFSET, to be set to blanks.

length specifies the number of positions to set to blanks.

***ALL** specifies that all positions after the position specified by OFFSET are to be blanked out.

If both BLANK and DATA are specified on the same LOCAL statement, the local data area is first blanked and then the data is placed in the area.

AREA specifies which area is to be used by the current LOCAL statement, and by all following local data area substitution expressions.

USER specifies that the user local data area is to be used. If AREA is not specified on the LOCAL statement, AREA-USER is assumed. The SSP and program products do not affect this area.

SYSTEM specifies that the system local data area is to be used. This area is used by the SSP and program products. **Any data** you place in this area may be lost and may affect the way the SSP and program products function. Therefore, you as a general user should not specify SYSTEM.

Example 1

The following statement places the word PAYROLL in the display station local data area starting in position 18 (through 24).

```
// LOCAL OFFSET-18,DATA-'PAYROLL'
```

Example 2

The following statement appears within a procedure and places the value of the first procedure parameter into the local data area, starting at position 12.

```
// LOCAL OFFSET-12,DATA-'?1?'
```

Example 3

This example clears positions 1 through 8 of the local data area, then places the procedure name in positions 1 through 8.

```
// LOCAL DATA-'?PROC?',BLANK-8
```

Example 4

To blank the entire local data area, enter:

```
// LOCAL BLANK-*ALL
```

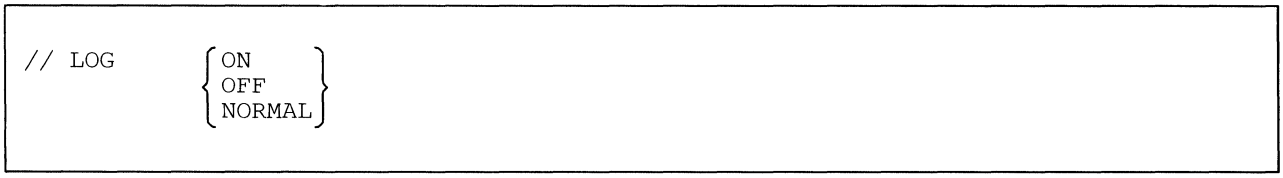
LOG OCL Statement

The LOG OCL statement indicates whether the OCL statements in a procedure are to be logged to the history file, regardless of the OCL statement history logging indicator in the procedure. This allows you to create your procedures with this indicator set to off (for better performance) but still have the OCL statements logged to the history file when debugging the procedure is necessary.

If you want the system help support menu options and user menu options logged to the history file, specify // LOG ON. If you do not want the options logged, specify // LOG OFF or // LOG NORMAL.

All procedure commands are logged to the history file regardless of the LOG OCL statement setting and the procedure's logging attribute.

Placement: The LOG OCL statement can be placed anywhere among the OCL statements. When the LOG OCL statement is specified in a procedure, it remains in effect until another LOG statement is processed or until the procedure ends. When the LOG statement is entered from the keyboard, it remains in effect until another LOG OCL statement is entered, the LOG procedure is entered, or the operator signs off.



S9020313-0

ON specifies that all OCL statements are to be logged to the history file, regardless of the procedure's logging indicator. Also, menu options are logged to the history file.

OFF specifies that no OCL statements are to be logged to the history file, regardless of the procedure's logging indicator. Menu options are not logged to the history file.

NORMAL specifies that the procedure's logging indicator is to be used. Menu options are not logged to the history file.

Example 1

When this example is entered from the keyboard, all processed OCL statements in the procedure are logged to the history file.

```
// LOG ON
```

Example 2

This example shows how to cause the OCL statements for only procedures PROC1 and PROC3 to be logged. The OCL statements for procedure PROC2 are not logged.

```
// LOG ON
PROC1
// LOG OFF
PROC2
// LOG ON
PROC3
```

MEMBER OCL Statement

The MEMBER OCL statement specifies the names of the current message members to be used for procedures and programs. The system gets messages to be displayed from the current message members. The specified members remain current until the system processes another MEMBER OCL statement from the display station or until the display station session is ended. If the MEMBER OCL statement is used in a procedure, the current members are changed only while the procedure runs. When the procedure ends, the message members that were current when the procedure began are again current.

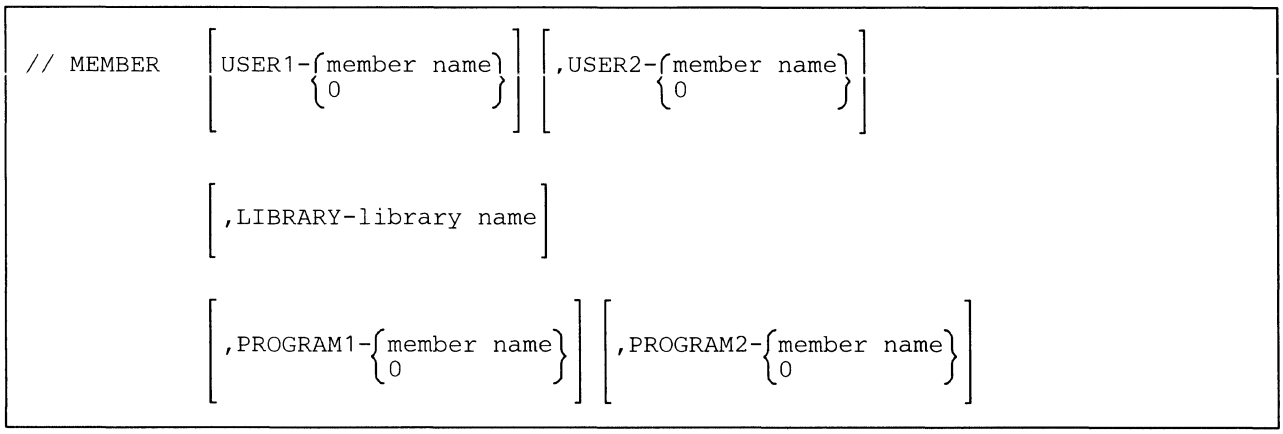
Four types of message members are specified by the MEMBER statement: USER1, USER2, PROGRAM1, and PROGRAM2.

Your program's first-level and second-level messages are retrieved from the USER1 and USER2 members, respectively. First-level messages can be up to 75 characters in length. Second-level messages can be up to 225 characters in length.

For information about creating message members, see the "CREATE Procedure" on page 4-132.

IBM program product (except SSP) first-level and second-level messages are retrieved from the PROGRAM1 and PROGRAM2 members, respectively. Your programs should not use the program product first-level and second-level message members.

Placement: The MEMBER OCL statement can appear anywhere among the OCL statements.



S9020314-0

USER1 specifies the name of the load member used for first-level messages for your programs and procedures. The USER1 message member is also used with the // * statement, the // ** statement, and procedure substitution expressions. If 0 (zero) is specified for member name, there is no current user first-level message member.

USER2 specifies the name of the load member used for second-level messages for your programs and procedures. If 0 (zero) is specified for member name, there is no current user second-level message member.

LIBRARY specifies the library to be searched for the message members. If a library name is specified, that library is searched and then the system library (#LIBRARY) is searched. If a library name is not entered, the current library is searched and then #LIBRARY is searched.

PROGRAM1 specifies the name of the load member used for IBM program product (except SSP) first-level messages. You need not specify this parameter in your procedures.

PROGRAM2 specifies the name of the load member used for IBM program product (except SSP) second-level messages. You need not specify this parameter in your procedures.

Example 1

In this example, the // * statement causes a message to be displayed. The message with a message identification code of 0006 is displayed from the first-level message load member named MESSAGES.

```
// MEMBER USER1-MESSAGES
// * 0006
```

Example 2

The following example shows a sample procedure, stored in a library named MYLIB, that runs a program named MYPROG. The program displays messages from a message member named MESSAGES, which is stored in a library named COMMLIB.

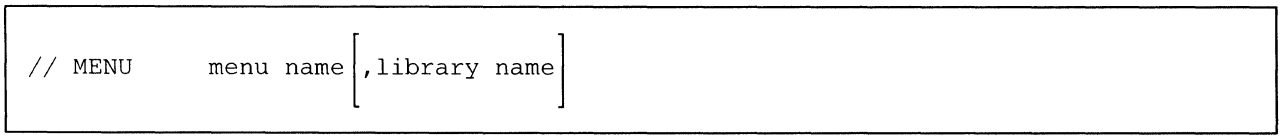
```
// MEMBER USER1-MESSAGES,LIBRARY-COMMLIB
// LOAD MYPROG
// RUN
```

MENU OCL Statement

The MENU OCL statement causes a specified menu to be displayed when the job containing the MENU OCL statement ends.

The menu can contain IGC characters; however, the SSP will not display the menu at a nonideographic display station. If an attempt is made to display an ideographic menu at a nonideographic display station, the SSP issues an error message.

Placement: The MENU statement can appear anywhere among the OCL statements. The MENU OCL statement will be ignored if it occurs in an MRT procedure.



S9020315-0

menu name specifies the menu to be displayed at the end of the job, and can contain from 1 through 6 characters.

library name specifies the library containing the MENU. If no library name is specified, the system searches the current library and then the system library (#LIBRARY) for the menu. The specified library becomes the session library. If the MENU statement is within a procedure, the current library for the procedure is not affected.

Example 1

In this example, the MENU OCL statement causes a menu called DAILY, which resides in the current library, to be displayed when the job containing the MENU statement ends.

```
// MENU DAILY
```

Example 2

In this example, the MENU OCL statement causes a menu called PAYROL, which resides in a library named PAYLIB, to be displayed when the job containing the MENU OCL statement ends. PAYLIB becomes the current library and the session library after the job ends.

```
// MENU PAYROL,PAYLIB
```

MSG OCL Statement

The MSG OCL statement sends a message to the system console, to a selected display station, to a selected display station operator, to all display stations, to personal computer locations in the IBM Token-Ring Network, or to a user on another system.

Messages received by a display station or operator are placed in the system message file (#MESSAGE). When an initial program load (IPL) is done, messages that were sent to a specific display station are removed from the message file. Messages that were sent to a specific operator and are more than seven days old are also removed from the message file when an IPL is done. Because the message file is updated during an IPL, the MSG OCL statement is not allowed in the #STRTUP1 procedure, which requires a dedicated system to start jobs during the IPL process.

The MSGFILE procedure can be used to:

- Define the size and location of the message file.
- List the display stations and users that have messages in the message file.
- Remove messages from the message file.

For more information about the MSGFILE procedure, see the "MSGFILE Procedure" on page 4-299

If an operator is signed on to the display station when a message is sent, the audible alarm sounds and the Message Waiting light comes on. Up to 25 messages can be waiting to be displayed for any display station or operator. If an attempt is made to send a message when 25 messages are waiting to be displayed, the messages are not placed in the message file and the sender is notified that the display station or the operator cannot receive messages at this time.

If an operator is signed on to more than one display station when a message is sent to a specific operator, the audible alarm sounds and the Message Waiting light comes on at all of the display stations that the operator is signed on to. The message is displayed at the first display station that requests that the message be displayed. After the message is displayed, the Message Waiting light is reset at all of the display stations that the operator is signed on to.

If an operator is not signed on when a message is sent and an entry in the user identification file exists for that operator, that message is placed in the message file. The message is displayed when the operator signs on. If the operator does not have an entry in the user identification file, the sender is notified that the message cannot be sent to that operator.

Placement: The MSG statement can appear anywhere among the OCL statements.

Network, as used in the following discussion, can consist of System/36s and PCs attached to System/36s.

```
// MSG      [                          ]
             | display id              |  ,message text
             | user id                  |
             | pc location              |
             | (user id,address)        |
             | (pc location,address)    |
             | group name               |
             | ALL                      |
             [                          ]
```

S9020316-2

display id specifies the work station ID of the display station to which the message is to be sent. The STATUS WORKSTN command can be used to determine the display station IDs. If the first parameter is not specified, the message is sent to the **console** display of the system console. If the work station ID of the system console is specified, the message is sent to the **command** display of the system console.

user id specifies the 1- to 8-character user ID that identifies the operator to whom the message is to be sent. Each display station operator enters a user ID when they sign on to the system. The STATUS WORKSTN command can be used to determine user IDs. If the user ID of the system operator is specified, the message is sent to the **command** display of the system console.

If you specify the user ID of the system operator, the message is sent to the command display of the system console. (The message can be viewed from the console display.)

Note: It is not recommended to have a user ID that includes special characters such as greater than (>) or less than (<) signs and parentheses. These characters may be used as control characters by the system.

pc location specifies the 1- to 8-character personal computer location name in the IBM Token-Ring Network to which the message is to be sent. You can use the STATUS SUBSYS command to determine the personal computer locations that are active.

(user id,address) specifies the 1- to 8-character user ID and the 1- to 8-character address that together identify a user in the network to whom the message is to be sent. The user ID and address must already exist in the directory.

Notes:

1. *This parameter is valid only if DSNX-ND is installed.*

2. *It is not recommended to have a user ID that includes special characters such as greater than (>) or less than (<) signs and parentheses. These characters may be used as control characters by the system.*

(pc location,address) specifies the 1- to 8-character personal computer location name in the IBM Token-Ring Network and the 1- to 8-character address that together identify the personal computer location name in the network to which the message is to be sent. The PC location name and address must already exist in the directory.

Note: This parameter is valid only if DSNX-ND is installed.

group name specifies the name of the distribution list given in a System/36 directory that identifies a group of users in the network to which the message is sent.

Note: This parameter is valid only if DSNX-ND is installed.

ALL specifies that the message is to be sent to all display stations.

message text specifies the message to be sent. Up to 75 alphameric or special characters can be entered. The message text can contain ideographic characters; however, the SSP does not send the message to a nonideographic display station. If an attempt is made to send a message to a nonideographic display station, an error message is displayed.

Example

In this example, a procedure is run from the job queue or evoked, and the operator wants to know when the procedure ends. The MSG OCL statement at the end of the procedure informs the operator at the display station when the procedure ends.

```
// LOAD PROGRAM1
// FILE NAME-FILE1
// RUN
// MSG ?WS?,PROGRAM1 IS COMPLETE
```

?WS? is an expression that causes the work station ID of the display station that placed PROCA on the job queue to be substituted into the MSG statement. For information about substitution expressions, see "Substitution Expressions" on page 3-10.

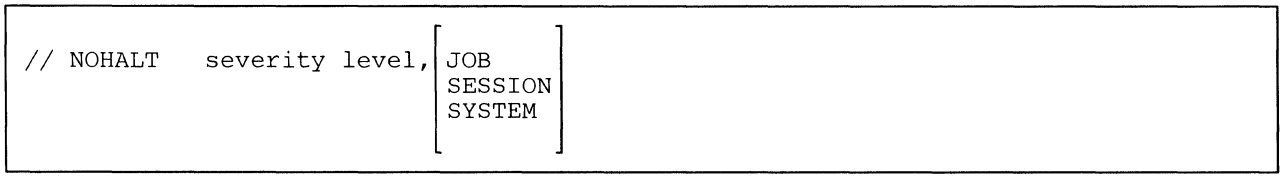
NOHALT OCL Statement

The NOHALT OCL statement specifies the automatic response severity level for the system, a job, or a job step. This allows you to have messages with automatic response values to be responded to by the system, rather than requiring an operator to enter the response to an error message. This is helpful when you are running the system without an operator, for example, overnight.

For more information on automatic response, see the "RESPONSE Procedure" on page 4-374.

Placement: The NOHALT OCL statement can be placed anywhere among the OCL statements.

If NOHALT is entered from the keyboard and the second parameter is not specified, it remains in effect until another NOHALT OCL statement is entered or until the operator signs off the system. If it is specified in a job and the second parameter is not specified, it remains in effect until another NOHALT OCL statement is processed, or until the job ends.



S9020317-0

severity level specifies the automatic response severity level. You can enter 0, 1, 2, 3, or 4.

0 Zero specifies that no messages are to be automatically responded to. If a message is displayed, the operator must enter a response to the message. When 0 is specified, automatic response is turned off.

1 Specifies that any messages having a severity level of 1 are to be automatically responded to by the system.

2 Specifies that any messages having a severity level of 1 or 2 are to be automatically responded to by the system.

3 Specifies that any messages having a severity level of 1, 2, or 3 are to be automatically responded to by the system.

4 Specifies that any messages having a severity level of 1, 2, 3, or 4 are to be automatically responded to by the system.

JOB specifies the severity level to be used for the job. The specified level remains in effect until the job ends or until changed by:

- Another NOHALT OCL statement
- A NOHALT procedure

If the NOHALT statement is within a procedure and no second parameter is specified, JOB is assumed. If the NOHALT statement is entered at the keyboard and JOB is specified, the second parameter is ignored, and the NOHALT status is set for the session.

SESSION specifies the severity level to be used for the display station session. The specified level remains in effect until the display station session ends or until changed by:

- Another NOHALT OCL statement
- A NOHALT procedure

If the NOHALT statement is within a procedure and SESSION is specified, the specified level does not take effect until the current job ends. If the NOHALT statement is entered from the keyboard and no second parameter is specified, SESSION is assumed.

SYSTEM specifies whether the severity level is to establish the system automatic response severity level and the severity level for the display station session. The system severity level establishes the severity level for other display stations that sign on **after** the SYSTEM parameter is processed; other display stations that were already signed on are not affected.

To establish the system severity level before any operators can sign on to the system, include the NOHALT statement in the initial program load (IPL) start-up procedure #STRTUP1. See "#STRTUP1 Procedure" on page 4-4 for more information.

If password security is active, SYSTEM can be specified only by an operator having the classification of system operator or higher. If password security is not active, SYSTEM can be specified only at the system console. If the NOHALT statement is within a procedure and SYSTEM is specified, the specified level takes effect immediately for the system automatic response severity level, but does not take effect for the display station session until the current job ends.

Example 1

To establish an automatic response severity level of 3 for the entire system, you could create a procedure named #STRTUP1 which would contain the following statement:

```
// NOHALT 3,SYSTEM
```

Example 2

To establish an automatic response severity level of 3 for the payroll program, you could specify the following in the payroll procedure:

```
// NOHALT 3
// LOAD PAYROLL
// FILE NAME-EMPLOY,DISP-OLD
// RUN
```

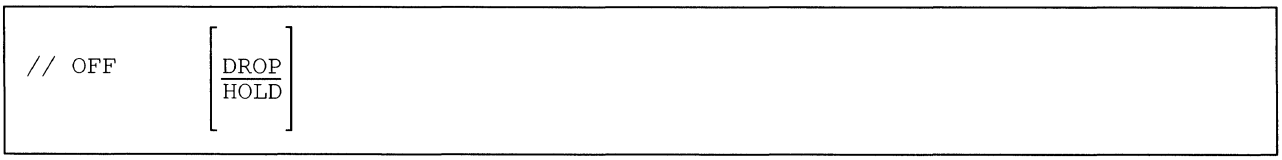
OFF OCL Statement

The OFF OCL statement immediately signs an operator off the system, thus ending a display station session. After the session ends, the sign-on display appears at the display station.

Messages on the message queue for the display station are displayed at sign off unless you specify OFF DROP on a remote work station (if they are not displayed, they remain on the queue and can be displayed later).

The OFF OCL statement cannot be used in inquiry mode or in a multiple requester terminal (MRT) procedure. It also cannot be used in an interactive communication feature session, in a job running from the job queue, or in a job that was evoked.

Placement: The OFF OCL statement can appear anywhere among the OCL statements except between the LOAD and the RUN OCL statements.



S9020318-0

DROP and HOLD are used only for remote display stations on switched or X.25 communications lines and display stations attached through display station pass-through. These parameters are ignored for other display stations.

DROP specifies that the communications session for this display is no longer needed.

If the display is communicating using display station pass-through, the pass-through session will terminate and the display will only communicate with its local system.

If the display is communicating using remote work station support (RWS), the communications line will be dropped if all work stations on that line have completed their activity after all work stations have entered OFF or OFF DROP. For X.25, the difference is that the virtual circuit to the remote work station controller will be disconnected instead of dropping the communications line. This support for X.25 must be configured using CNFIGSSP. If the display is the system console, an alternate console may use the CONSOLE command to transfer the system console function once the communications line has dropped.

HOLD specifies that the communications session for the display is to be held. A sign-on display will appear. If the display is communicating using display station pass-through, the sign-on display will be for the remote system.

Example

In the following example, a procedure runs the program LAST from a display station, and at the end of the job, signs the user off the system.

```
// LOAD LAST
// RUN
*
// OFF
```

POWER OCL Statement

The POWER OCL statement allows you to power off the system. No programs can be running on the system when the POWER OCL is encountered. If there are programs running, the POWER OCL statement is ignored.

The POWER OCL statement cannot be used in inquiry mode.

Placement: The POWER OCL statement can appear anywhere among the OCL statements except between the LOAD and the RUN OCL statements.

```
// POWER OFF
```

S9020319-0

OFF specifies that the system is to be powered off immediately. The OFF parameter is required.

Example

In the following example, the WAIT and POWER OFF OCL statements are used together to automatically do a nightly backup of your master files, and then power off the system.

```
* Nightly Backup Procedure
*
* Wait until 6 p.m. before beginning backup
*  and power off sequence.
// WAIT TIME-180000
*
* Allocate the diskette drive
// ALLOCATE UNIT-I1,CONTINUE-YES
*
* Save the master files on diskette
SAVE CUSTMAST,,,VOL001,M1
SAVE ITEMMAST,,,VOL001,M1
SAVE ACCTMAST,,,VOL001,M1
SAVE SHIPMAST,,,VOL001,M1
*
* Start POWER OFF sequence
// TAG PWRLOOP
// POWER OFF
* If POWER OFF is not successful,
*    then wait 1 minute and try POWER OFF again.
// WAIT INTERVAL-000100
// GOTO PWRLOOP
```

Because the system ignores the POWER OFF statement when it cannot be safely processed (for example, when another job is running), a GOTO and TAG loop is needed in case the POWER statement was not successful. The WAIT OCL statement is included to prevent the system from constantly trying the POWER statement, which would needlessly increase the amount of work the system is doing.

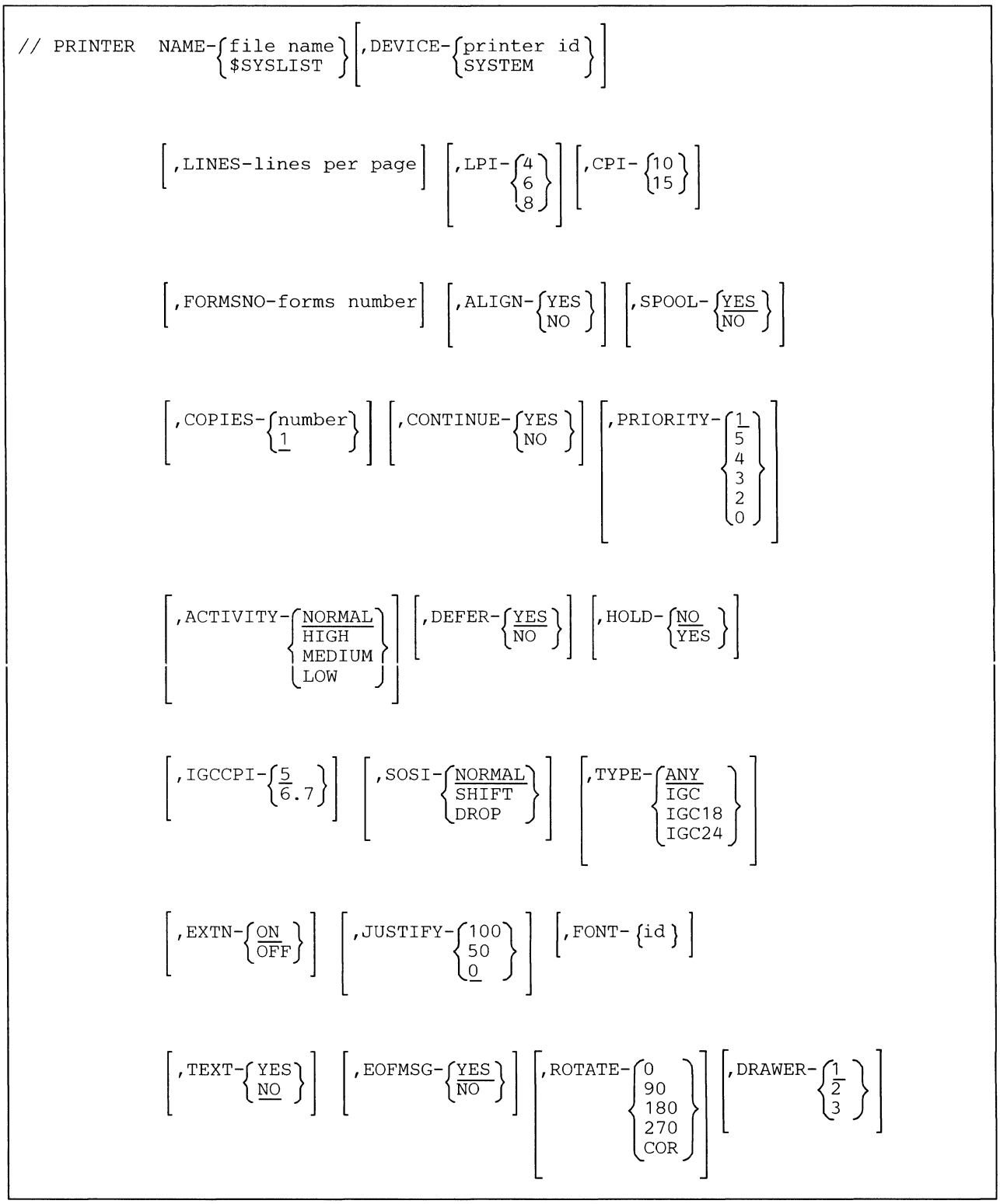
Note that this procedure would require a diskette magazine in slot M1. The magazine would have to be in place before 6 p.m.

PRINTER OCL Statement

The PRINTER OCL statement identifies and controls the output for a printer file created by one or more job steps. The PRINTER OCL statement can be used to make the following changes for a printer file:

- The printer to be used for the printed output
- The number of lines to be printed per page
- The vertical print density (lines per inch)
- The horizontal print density (characters per inch)
- The forms number to be used
- Whether the operator should line up the forms in the printer before printing begins
- Whether printed output should be spooled
- The number of copies of spooled output to be printed
- Whether the printer file is to be continued with following job steps
- The priority of the spooled output
- The activity of a spooled printer file
- Whether printing of spooled output can begin before the job step is complete
- Whether printed output should be held in the spool file after it is printed
- The amount of space to be left for the shift-out and shift-in characters (ideographic only)
- Whether the printing is to be done on a printer that can print ideographic characters
- Whether the extended character task is to be called to process any extended characters to be printed (ideographic only)
- Whether the output from a 5219 Printer is to be right-justified
- The identification number of the printwheel to be used on a 5219 Printer or the identification number of the font to be used on a 4224 Printer
- Whether the output is to be printed in final quality
- Whether the end of forms message is to be issued
- The orientation or size of the printed output on the page (rotation or reduction)
- The printer drawer from which paper is to be used

Placement: The PRINTER OCL statement can appear anywhere among the OCL statements. A PRINTER OCL statement may be specified for each printer file used by the job step. If a PRINTER OCL statement is not used, the SSP uses the PRINTER statement default values when printing the output. The defaults can come from the system configuration, the PRINT procedure, the LINES procedure, the SET procedure, and the FORMS OCL statement.


// PRINTER NAME-{file name / $SYSLIST} [,DEVICE-{printer id / SYSTEM}]
[,LINES-lines per page] [,LPI-{4 / 6 / 8}] [,CPI-{10 / 15}]
[,FORMSNO-forms number] [,ALIGN-{YES / NO}] [,SPOOL-{YES / NO}]
[,COPIES-{number / 1}] [,CONTINUE-{YES / NO}] [,PRIORITY-{1 / 5 / 4 / 3 / 2 / 0}]
[,ACTIVITY-{NORMAL / HIGH / MEDIUM / LOW}] [,DEFER-{YES / NO}] [,HOLD-{NO / YES}]
[,IGCCPI-{5 / 6.7}] [,SOSI-{NORMAL / SHIFT / DROP}] [,TYPE-{ANY / IGC / IGC18 / IGC24}]
[,EXTN-{ON / OFF}] [,JUSTIFY-{100 / 50 / 0}] [,FONT-{id}]
[,TEXT-{YES / NO}] [,EOFMSG-{YES / NO}] [,ROTATE-{0 / 90 / 180 / 270 / COR}] [,DRAWER-{1 / 2 / 3}]

S9020320-1

NAME specifies the name that the program uses to refer to the printer file. The file name can be any combination of characters (numeric, alphabetic, and special). Commas (,), periods (.), apostrophes ('), blanks, question marks (?), slashes (/), hyphens (-), greater than signs (>), plus signs (+), and equal signs (=) should be used with caution because they have special meanings within procedures. The first character of a file name must be alphabetic (A through Z, #, $, or @). The number of characters in a file name cannot exceed eight.

When the PRINTER OCL statement is not between a LOAD and RUN OCL statement pair and the CONTINUE parameter is specified, the NAME parameter cannot be specified.

$SYSLIST specifies that the PRINTER statement controls system list output. For information about system list output, see the "SYSLIST OCL Statement" on page 5-113.

If the NAME parameter does not match a file name specified in the program, the SSP will use the default values of the PRINTER statement when printing the output.

Note: $SYSLIST is ignored if debugging is turned on. For more information, refer to the "DEBUG OCL Statement" on page 5-28.

DEVICE specifies the printer to be used for the print file.

printer id specifies the work station ID of the printer to be used. You can use the STATUS WORKSTN command to determine the printer IDs.

SYSTEM specifies that the system printer is to be used.

If DEVICE is not specified, the following conditions apply:

- If NAME-$SYSLIST is specified, the PRINTER OCL statement overrides the system list device setting. For example, if the current system list device is CRT, the output is printed instead of displayed.
- If a file name other than $SYSLIST is specified, and the procedure is a multiple requester terminal (MRT) procedure, the printer output is sent to the system printer.
- If the job step containing the PRINTER statement is released, evoked, or placed on the job queue, the printed output is sent to the printer that was specified during system configuration. The output is sent either to the session printer or to the system printer.
- If the job step is run as a single requester terminal procedure, the printer output is sent to the session printer.

LINES specifies the number of print lines per page. The number of lines per page can be any number from 1 through 112. If LINES is not specified, the number of lines set previously by a FORMS OCL statement, LINES procedure, or PRINT procedure is used. If the number of lines per page is specified in a program, the program's value is used. If the number of lines per page was not set during the session, the value specified for the display station during system configuration or assigned by the SET procedure or the $SETCF utility program is used.

For SSP utility programs and most user-written programs, the printer skips to a new page when six less than the number of lines specified are printed. For example, if LINES-66 is specified, the printer skips to a new page after printing line 60. Also the printing starts on line 6 of the next page. Therefore, if you specify 66 lines per page you really get 54 lines of output per page (66 - 12 = 54). If LINES-13 is specified, one line is printed per page. When 12 or less lines are specified, printing occurs on every line of each page.

For print operations from your programs, the SSP indicates a printer overflow condition when six less than the number of lines specified are printed (unless the program uses another value).

| When using the 3812 Printer, you can get landscape/rotated printing by specifying LINES-51 (or less).

LPI (lines per inch) specifies the vertical print density to use for printed output from the display station session. The values that can be specified are 4, 6, and 8. If the output is printed on a printer other than the 4214, 4224, 4234, 4245, 5224, or 5225 Printer, the LPI parameter is ignored. If LPI-4 is specified on the 4245 printer, it actually prints at LPI-6.

If LPI is not specified, the system uses the LPI value set previously by a FORMS OCL statement, LINES procedure, or PRINT procedure. If the LPI parameter is not specified and the number of lines per inch was not previously set during the session, the system uses the value specified when the printer was configured (either 4, 6, or 8). (Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is actually printed, not when it is intercepted in the spool file.) If no LPI value was specified when the printer was configured, the LPI value is set through the printer operator panel.

CPI (characters per inch) specifies the horizontal print density to use for printed output. The values that can
| be specified are 10 or 15. If CPI-15 is used, the output should be printed on a 3812 IPDS, 4214, 4224, 4234, 5224, or 5225 Printer. If CPI-15 is attempted on another printer, a message is displayed and the operator controlling the printer can either cancel or continue the printing. CPI is ignored if the output is printed on a 5219 Printer.

If CPI is not specified, the CPI density set previously by a FORMS statement, LINES procedure, or PRINT procedure is used. If the CPI parameter is not specified and the characters per inch was not previously set during the session, the system uses the value specified when the printer was configured (either 10 or 15). (Because output can be directed to any printer from the spool file, this default configuration value is determined when the output is actually printed, not when it is intercepted in the spool file.) If no CPI value was specified when the printer was configured, the CPI value is set through the printers operator panel if the printer supports CPI.

| See also FONT, later in the description of this OCL statement.

FORMSNO specifies the forms number of the forms to be used. The forms number can be any combination of up to 4 characters except commas (,), apostrophes ('), and blanks. However, question marks (?), slashes (/), equal signs (=), greater than signs (>), plus signs (+), and hyphens (-) should be used with caution because they have special meanings within procedures.

If FORMSNO is not specified, the forms number set previously by a FORMS statement is used. If the forms number was not set during the session, the forms number specified for the display station during system configuration or assigned by the SET procedure or the $SETCF utility program is used.

ALIGN allows the operator who controls the printer to line up the forms in the printer before the printing of output for the job step begins. If the ALIGN parameter is specified, any alignment indicator used in the program is ignored. If ALIGN is not specified, the alignment indicator specified in the program (if any) is used. ALIGN is ignored if the output is printed on a 4224 or 3812 IPDS Printer because the 4224 and 3812 IPDS Printers format and print a page of data, not one line at a time.

YES specifies that the system is to print the first line of output and display a message. The operator can then line up the forms and then either:

- Select an option that causes a line to be printed again and the alignment message to be redisplayed
- Select an option that causes printing to continue with the rest of the output

NO specifies that the system is to print the output without allowing the forms to be lined up.

SPOOL specifies whether the printer output is to be spooled.

YES specifies that output is to be spooled.

NO specifies that output is not to be spooled and that the printer used in the DEVICE parameter is to be assigned to the job or job step. If the printer is not available, a message is displayed.

If spooling was selected during system configuration, SPOOL-YES is assumed; otherwise, the SPOOL parameter is ignored.

COPIES specifies the number of copies of spooled printer output to be printed for the job step. You can specify any number from 1 through 255. If COPIES is not specified, 1 is assumed. The COPIES parameter affects only spooled output.

CONTINUE specifies whether the printed output is to continue, that is, whether printer output from this job step and from following job steps is to be considered as a single print step. The CONTINUE and NAME parameters can only be specified together when the PRINTER statement is placed between the LOAD and RUN OCL statements.

Note: If the printer output has been spooled, the spool file entry will be processed as if it were created by one job step. If you specify different print file characteristics, for example, a different number of lines per page, in steps within the job, those values may be used only when the spool file entry is printed from beginning to end. If you use the RESTART PRT control command to start printing such an entry at a place other than its beginning, programming errors may occur, or the printer output may be different than if the entire spool file entry had been printed.

YES specifies that the printer output for a specific printer is to continue in following job steps. The DEVICE parameter specifies the printer whose output is to be continued.

Printing continues until another PRINTER statement containing CONTINUE-NO is processed for that device or until the job ends. This allows you to group the output of several job steps together, rather than possibly having each job step's output interspersed with the output from other jobs on the system.

Note: A page eject occurs between the output of each job step.

If the printer output is being spooled, the spool file entry associated with the printer output is left open. If the printer output is not being spooled, that is, the output is being sent directly to a printer, the printer remains allocated to the job.

If two or more printer files in the same job step refer to the same printer, only one of the printer files from the job step continues. If print spooling is being used, the other printer files are placed on the spool file as separate entries. The printer file that continues is determined by:

- If a PRINTER OCL statement is specified for same printer device, the system continues the print step represented by the first PRINTER OCL statement.
- If a PRINTER OCL statement is not specified for the same printer device, the system continues the first print step to be allocated by the program that uses the printer device.

When a print file is being continued, all parameters are ignored for that device except NAME, DEVICE, and CONTINUE.

If a job step is released (by using the RELEASE parameter on the ATTR OCL statement), or runs a multiple requester terminal (MRT) program, that step cannot use continued print files defined by other steps of the job. That step must have its own PRINTER OCL statement for that print file.

NO specifies that the print step is to no longer continue. When the print step is really closed depends on the following:

- If the PRINTER statement is not between a LOAD and RUN OCL statement pair, the continued print step is immediately considered complete.
- If the PRINTER statement is between a LOAD and RUN OCL statement pair, the continued print step is complete after the current job step ends (or after the program closes the printer file).

When the print step is complete, the spool entry is indicated as complete or the printer is released from the job.

PRIORITY specifies the priority of spooled output from the job step. The priority can be any decimal number from 5 through 0. The system prints spool file entries with higher priority numbers before jobs with lower priority numbers. For example, all printed output with a priority of 5 is printed before any other spool file entries. Spool file entries with the same priority are printed in the order that they were placed on the spool file. Priority 1 entries are the last entries printed by the system. If the PRIORITY parameter is not specified, PRIORITY-1 is assumed.

Priority 0 entries are placed on the spool file with a priority of 1 and are held. These entries are not printed until a RELEASE control command is entered to specifically cause them to be printed.

The actual order of the printed output may change because of the forms number being used. See the PRT parameter on the "START Control Command" on page 6-36 for how the order can be changed.

ACTIVITY specifies the expected activity for a spooled printer file. This allows you to control the size of the spool intercept buffer (which temporarily contains printer output) allocated to your program in order to decrease the processing time your program takes.

Spooled output is really written to the spool file when the spool intercept buffer allocated to your program becomes full. If the buffer is too small for the amount of activity, it will fill up quickly and will have to be written to the spool file very often, thus possibly increasing the processing time for your job and other jobs on the system. If the buffer is too large, you may needlessly increase the amount of main storage your program requires.

The ACTIVITY parameter is ignored if print spooling is not being performed.

NORMAL specifies that a spool intercept buffer of normal size is to be allocated to your program. (The buffer is 512 bytes for system list output; 1280 bytes for all other printed output.) If ACTIVITY is not specified, ACTIVITY-NORMAL is assumed.

HIGH specifies that a spool intercept buffer much larger than normal is to be allocated to your program. (The buffer is 2560 bytes.)

MEDIUM specifies that a spool intercept buffer larger than normal is to be allocated to your program. (The buffer is 1280 bytes.)

LOW specifies that a spool intercept buffer smaller than normal is to be allocated to your program. (The buffer is 256 bytes.)

DEFER specifies whether the system can begin printing spooled output before the print step is complete. The DEFER parameter affects only spooled output.

YES specifies that the system should not print spooled output from the job step until the print step is complete. If DEFER is not specified, DEFER-YES is assumed.

NO specifies that the system can begin printing spooled output from the print step before the step is complete.

If you specify DEFER-NO for long-printing output, you might reduce the total time needed to run the program and print the output.

HOLD specifies whether the printed output is to be held on the spool file after it has been printed. To hold printed output in the spool file without it being printed, see the PRIORITY parameter.

NO specifies that the printed output is not to be held and is to be removed from the spool file after being printed. If HOLD is not specified, HOLD-NO is assumed.

YES specifies that the printed output is to be held after all copies are printed; the number of copies held is set to one.

IGCCPI (ideographic characters per inch) specifies the horizontal print density to use for ideographic printed output. The values that can be specified are 5 or 6.7. If IGCCPI-6.7 is used, the output must be printed on a 5227 or 5553 ideographic printer. If IGCCPI-6.7 is attempted on any other printer, a error message is displayed. If IGCCPI is not specified, IGCCPI-5 is assumed.

SOSI (shift-out shift-in) specifies the amount of space to be left for the shift-out (hex 0E) and shift-in (hex 0F) characters when printing. If SOSI-SHIFT or SOSI-DROP is specified, the output must be printed on a 5227 or 5553 ideographic printer. If any other printer is used, an error message is displayed.

NORMAL specifies that a space be substituted for each shift-out and shift-in character for printing. If the SOSI keyword is not specified, NORMAL is assumed.

SHIFT specifies that no space be left for the shift-out (hex 0E) character and that two spaces be left for the shift-in (hex 0F) character for printing.

DROP specifies that no spaces be left for the shift-out or shift-in characters. During printing, all characters will be moved to the left to fill the spaces deleted for the shift-out and shift-in characters.

TYPE specifies to which printer the output should be printed.

ANY specifies that the printer output can be printed on any type of printer. If an ideographic character is encountered and the printer cannot print ideographic characters, the character position is blanked. If the TYPE parameter is not specified, TYPE-ANY is assumed.

IGC specifies that the output contains ideographic characters and should be printed on a printer that can print ideographic characters. If the printer cannot print ideographic characters and if the output is not spooled, ideographic characters will be blanked. If the printer cannot print ideographic characters and if the output is spooled, the operator can choose to hold the spooled output or to print it, with blanks substituted for ideographic characters.

IGC18 specifies that the output should be printed on a 5224 Model 12 or 5225 Models 11 or 12 Printer.

IGC24 specifies that the output should be printed on a 5227 or 5553 ideographic printer.

Note: If a system does not have the ideographic feature, the printing of ideographic characters is unpredictable.

EXTN specifies whether the extended character task is called to process any extended characters in the printer output. The EXTN parameter is for the ideographic version of the SSP and is ignored for nonideographic systems. If NAME-$SYSLIST is specified and the system list device is a printer, the EXTN parameter overrides the value specified on the SYSLIST OCL statement.

ON specifies that the extended character task is called to process any extended characters in the printer output. If the EXTN parameter is not specified, ON is assumed.

OFF specifies that the extended character task is not called; the system-defined default ideographic character is printed for any extended characters in the output.

JUSTIFY specifies the percentage that output from a 5219 Printer is to be right-justified. The system sets the right margin for justification based on the record length. For example, if the record length is 80, all text is justified to column 80. Justification can be controlled with a program by using skipping and spacing values. If skipping and spacing values are specified before printing, the printed line will be justified. If skipping and spacing values are specified after printing, the printed line will not be justified. If the output is printed on a printer other than a 5219 Printer, the JUSTIFY parameter is ignored.

100 specifies that the output is to be right-justified to the right margin.

50 specifies that the output is to be right-justified to a position halfway between the end of the text and the right margin (that is, ragged edged).

0 specifies that the output is not to be right-justified. If the JUSTIFY parameter is not specified, 0 is assumed.

FONT specifies the identification number of the printwheel to be supported on a 5219 Printer or the font to be printed on a 4224 Printer. Not all 5219 fonts are available on the 4224 Printer. The following table shows the valid font ID numbers and how they map to the 4224 Printer fonts.

5219 Font ID	Description	4224 Font ID	Description
05	Rhetoric	0B	Courier 10
0B	Courier 10	0B	Courier 10
0C	Prestige Pica	0B	Courier 10
0D	Artisan 10	0B	Courier 10
14	Pica	0B	Courier 10
50	Symbol Scribe	55	Courier 12
54	Script	55	Courier 12
55	Courier 12	55	Courier 12
56	Prestige Elite	55	Courier 12
57	Letter Gothic	57	Gothic 12
5B	Light Italic	55	Courier 12
9E	Modern	A0	Essay
9F	Boldface	A0	Essay
A0	Essay	A0	Essay
A2	Essay Italic	A0	Essay
DD	Prestige 15	DF	Courier 15
DE	Gothic 15	DE	Gothic 15
DF	Courier 15	DF	Courier 15
E1	Symbol Scribe	DF	Courier 15

Not all of the 4224 fonts will print at all quality levels. If TEXT-YES is specified, the Gothic fonts cannot be used. Instead, the Courier fonts will be used.

If TEXT-NO is specified or if the TEXT parameter is not specified, and the quality on the printer operator panel is set to 001 (DP quality), then only the Gothic fonts can be used. If any other font is specified, then a corresponding Gothic font will be used. Gothic 12 will be used instead of the Essay font. If the quality on the printer operator panel is set to 002 (text quality) or 003 (near letter quality), then the Gothic fonts cannot be used. The Courier fonts will be used only if TEXT-NO is specified or if the TEXT parameter is not specified.

The default value for the FONT parameter is specified during the CNFIGSSP procedure. If the output is printed on a printer other than a 5219 or 4224 Printer, the FONT parameter is ignored. Also, if a value is specified for both the CPI parameter and the FONT parameter, an error message is issued.

TEXT specifies whether the output is to be printed in draft or final quality. If the output is printed on a printer other than a 4214, 4224, 4234, or 5219 Printer, the TEXT parameter is ignored.

YES specifies that the output is to be printed in final quality, regardless of the switch setting on the 4214, 4224, 4234, or 5219 Printer.

NO specifies that the TEXT keyword is not used on the PRINTER OCL statement. The SSP will send a command to the printer, which will cause it to print based upon the setting of its quality switch if it has one.

EOFMSG specifies whether the end of forms message is to be issued to the console or subconsole operator.

NO specifies that the end of forms message is not to be issued.

YES specifies that the end of forms message is to be issued. If the EOFMSG parameter is not specified, YES is assumed.

ROTATE specifies that the output is to be rotated on the page or that the size of the printed output is to be reduced. This parameter is valid only if output is printed on a 3812 Printer.

0 specifies that printed output is to be rotated zero degrees on the page.

90 specifies that printed output is to be rotated 90 degrees on the page.

180 specifies that printed output is to be rotated 180 degrees on the page.

270 specifies that printed output is to be rotated 270 degrees on the page.

COR (computer output reduction) specifies that the size of the printed output is to be reduced. Output normally printed on 14-inch wide paper will print on 8-1/2 x 11 inch paper.

DRAWER specifies that the paper used by a printer is to be selected from the specified drawer. The 3812 SNA Control Stream (SCS) and Intelligent Printer Data Stream (IPDS) Printers have two paper drawers, and the 4214 and 5219 Printers have three paper drawers.

1 specifies that output is to be printed on paper from drawer 1. If the DRAWER parameter is not specified, 1 is assumed.

2 specifies that output is to be printed on paper from drawer 2.

3 specifies that output is to be printed on paper from drawer 3. Specify drawer 3 if you want to select envelopes on the 5219 Printer.

The valid combinations of these parameters for available printers are indicated in the following table.

Parameter	3262	3812 SCS	3812 IPDS	4214	4224	4234	4245	5219	5224 Model 1 Model 2	5224 Model 12	5225 Model 1 Model 4	5225 Model 11 Model 12	5256	5262	5227/ 5553
NAME	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DEVICE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LINES	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LPI			X	X	X	X	X		X	X	X	X			X
CPI			X	X	X	X			X		X				
FORMSNO	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ALIGN	X	X		X		X	X	X	X	X	X	X	X	X	X
SPOOL	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
COPIES	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
CONTINUE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
PRIORITY	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ACTIVITY	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
DEFER	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
HOLD	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
IGCCPI															X
SOSI															X
TYPE										X		X			X
EXTN										X		X			X
JUSTIFY								X							
FONT		X	X		X			X							
TEXT				X	X	X		X							
EOFMSG	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ROTATE		X	X												
DRAWER		X	X	X				X							

S9020517-2

Example 1

For the program named PROGRAM1, output to the file called PRINT1 is to go to the system printer and is to be spooled. Three copies are to be printed and the SSP can begin printing before the job step ends.

```
// LOAD PROGRAM1
// PRINTER NAME-PRINT1,COPIES-3,DEFER-NO,
//         DEVICE-SYSTEM
// RUN
```

Example 2

The output from the three programs is to be printed as a single report on the session printer.

```
// LOAD PROG1
// PRINTER NAME-PRINT1A,CONTINUE-YES
// RUN
// LOAD PROG2
// PRINTER NAME-PRINT1B
// RUN
// LOAD PROG3
// PRINTER NAME-PRINT1C
// RUN
```

PROMPT OCL Statement

The PROMPT OCL statement allows you to:

- Prompt for up to 64 procedure parameters by using one or more display formats. Up to 1024 characters can be prompted for.
- Define each parameter for the operator.
- Assign default parameters.
- Show the display format to be read on the first read operation to the display station in a program. This is also called read-under-format.
- Control various display format functions.

The PROMPT OCL statement provides a return code (that is, a value is returned to the procedure) to indicate which command key or function key was pressed. You can use the ?CD? substitution expression to determine the value of this code. See the "?CD? (Return Code) Expression" on page 3-16 for more information. The following chart shows the command and function keys and their return codes. These return codes will only be set if PDATA-YES is not specified.

Return Code	Key Pressed
0000	Enter/Rec Adv
2001 through 2024	Command keys 1 through 24
2090	Roll Up (Roll ↑)
2091	Roll Down (Roll ↓)
2092	Help
2093	Record Backspace (Home key pressed while the cursor is in the Home position)

See the manual *Creating Displays* for an explanation of how to design and enter display formats. The PROMPT OCL statement cannot be used to show a display format that contains more than 1024 characters of run time output data.

The following rules apply to the D specifications for input fields and input/output fields on a format displayed by the PROMPT statement:

- Fields must be defined in parameter number order on the D specification. Fields do not, however, have to appear in parameter number order on the display screen. (The position on the display screen is determined by the line number and horizontal position entries.)
- Normally, the output data entry (columns 23 and 24) should be an indicator number that corresponds to the parameter position in the procedure. If a parameter has already been assigned a value when the prompt format is displayed, the indicator for the parameter is turned on, and the assigned value is displayed. If the parameter has not already been assigned a value, the constant default value coded in columns 57 through 64 is displayed. If an indicator number greater than 11 is specified in the output data entry, the default value in columns 57 through 64 will always be displayed when the format is displayed by the PROMPT statement.
- Y must be specified in the input allowed entry (column 26).
- A default value can be coded in the Constant Data entry (columns 57 through 64).

Placement: The PROMPT OCL statement can be placed anywhere among the OCL statements. It cannot be entered at the keyboard, and is not allowed from the job queue or from a job started by the EVOKE OCL statement.

```
// PROMPT   MEMBER-display format load member,FORMAT-display format name

          [,LIBRARY-library name] [,START-{parameter number}]
                                          {1              }

          [,LENGTH-n        ] [,PDATA-{NO }] [,UPSI-{NO }]
                   'n,n...'           {YES}          {YES}
```

S9020321-0

MEMBER specifies the name of the display format load member that contains the display format.

FORMAT specifies the name of the display format that is to be displayed.

LIBRARY specifies the library to be searched for the display format load member. If a library name is specified, that library is searched and then the system library (#LIBRARY) is searched. If a library name is not entered, the current library is searched, and then #LIBRARY is searched.

START specifies the number of the first procedure parameter to be displayed and prompted for. Any previously defined parameters whose positions are less than the position specified are not affected.

parameter number can be any number from 1 through 64. If no parameter is specified, the first parameter is assumed.

LENGTH specifies the number of display format input and output positions to be used by each procedure parameter being displayed and prompted for.

n specifies the length of one parameter (quotation marks are not necessary) and the value applies to the parameter specified by START.

'n,n...' specifies the length of more than one parameter, starting with the parameter specified by START. Each **n** can have a value of 0 (zero) through 128. The total length cannot exceed 1024 characters.

If a value is not specified for a position (for example: ',,'), the parameter is to have a length of eight.

A value of 0 (zero) indicates that the parameter is not to be displayed or prompted for, and the input field for that parameter is to be skipped. If 0 is specified for the parameter's length, that parameter's value and length are not changed from what they were before the PROMPT statement was processed.

When the display format is shown, the parameters are displayed beginning in the leftmost position in the display format field. If the actual length of the parameter to be displayed is greater than the corresponding length value specified, the displayed parameter's rightmost characters are truncated (that is, cut off; only the leftmost number of characters specified are displayed).

After the operator types in the parameters and presses the Enter key or a valid command key, the length of each parameter is determined by counting the number of characters entered from left to right. All characters, including leading and embedded blanks, are counted up to the rightmost nonblank character. For example, if 'AB␣C' were entered, it would have a length of 4. If '␣␣XY␣Z␣' were entered, it would have a length of 6. The parameter in this case would be ␣␣XY␣Z.

PDATA specifies whether the input from the display format is to be used as parameters, or to be held as input for the program's first input request from the display station.

NO specifies the input from the display is to be used as parameters for a procedure, and is to be treated as if it had been entered with the procedure. If PDATA is not specified, NO is assumed.

YES specifies the input from the display format is to be used as program data, and is to be given to a program on its first input request from the display station.

UPSI specifies whether the current setting of the eight UPSI switches is to be placed into the display format indicator table.

NO specifies that the switches are not to be used. If UPSI is not specified, NO is assumed.

YES specifies that the current setting of the eight UPSI switches (1 through 8) are to be placed into the display format indicator table as indicators 91 through 98, respectively.

Example 1

The following PROMPT statement displays a format named DISPLAY1 from a format load member named PROMPTS.

```
// PROMPT MEMBER-PROMPTS,FORMAT-DISPLAY1
```

Example 2

See "Example 3: Procedure SCRNPRT" on page 2-16 for an example of a procedure using the PROMPT OCL statement.

Example 3

The following PROMPT statements show two prompt displays. The first display contains prompts for parameters 1 through 5; the second display contains prompts for parameters 6 through 10. Parameter 2 has a length of 6, parameter 9 has a length of 20, and all other parameters have lengths of 8. Figure 5-1 and Figure 5-2 show the display formats used with these statements.

```
// PROMPT MEMBER-SCREENS,FORMAT-DISPLAY1,  PROMPT Statement for
//        LENGTH-'8,6,8,8,8'                 Parameters 1-5
*
// PROMPT MEMBER-SCREENS,FORMAT-DISPLAY2,  PROMPT Statement for
//        START-6,LENGTH-',,,20,'            Parameters 6-10
*
PROCA ?1?,?2?,?3?,?4?,?5?,?6?,?7?,?8?,?9?,?10?
```

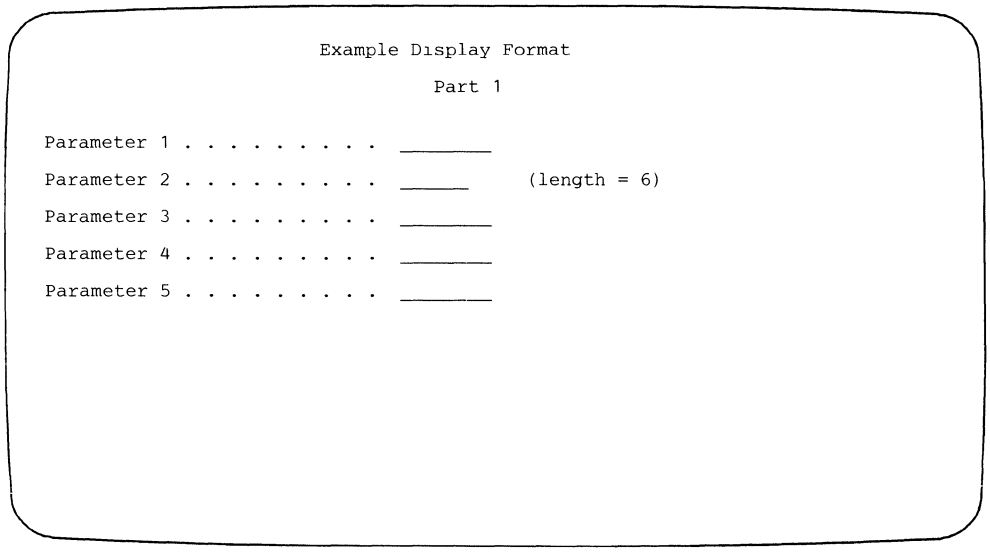


Figure 5-1. Example Display Format DISPLAY1

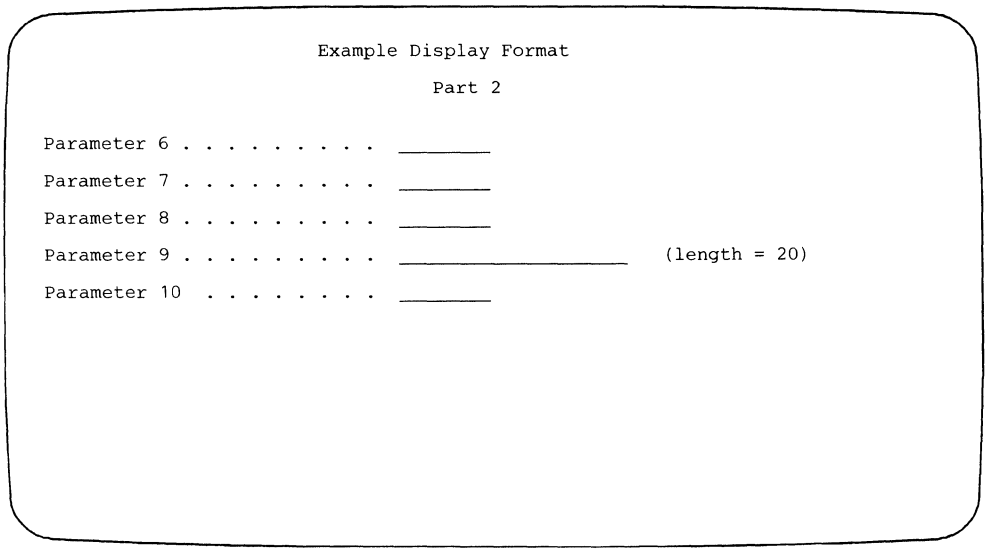


Figure 5-2. Example Display Format DISPLAY2

Example 4

The following statements demonstrate a potential problem with assigning procedure parameters before a PROMPT OCL statement is processed. In this example, the PROMPT statement shows a prompt display that contains two fields: Field 1 is defined in the D specification of the display format as an input field, and Field 2 is defined as an input/output field. Field 2 is also defined in the D specification to have data supplied by a program.

```
// EVALUATE P1=A
// PROMPT MEMBER-SCREENS,FORMAT-DISPLAY3,LENGTH-'1,2'
*
PROCB ?1?,?2?
```

Figure 5-3 shows how the display (named DISPLAY3) appears after the PROMPT statement is processed: the character 'A' has been placed in **Field 2** even though the EVALUATE PCE statement processed before the PROMPT statement assigned that value to the first parameter (P1). To have the field and parameter correspond, the operator must enter 'A' in Field 1. If the operator also entered the characters 'BC' in Field 2, the following values would be substituted in PROCB:

```
PROCB A,BC
```

Parameters assigned before PROMPT statements must correspond in order with the output or input/output fields defined in a display format to have data supplied by a program.

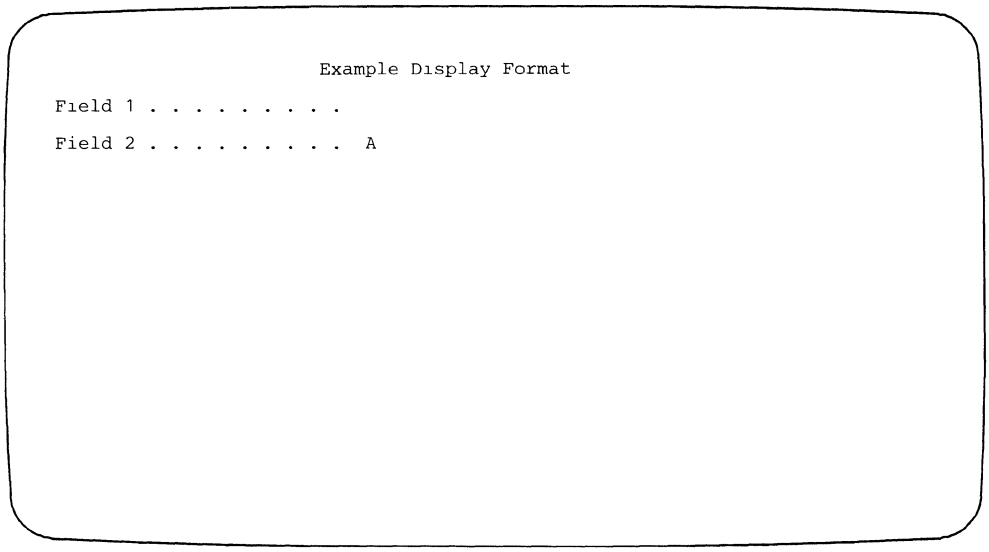


Figure 5-3. Example Display Format DISPLAY3

REGION OCL Statement

The REGION OCL statement specifies the region size for a job or job step. The session default region size that was set by the SET procedure or the $SETCF utility program is overridden.

The system ensures that an amount of main storage at least as large as the job region is available for use by the job steps in the job. If a job step releases the requesting display station or if the job step runs a multiple requester terminal procedure, the job step is, in effect, a new job. The SSP does not ensure that enough storage will be available for such job steps.

The SSP considers all user storage not occupied by nonswappable programs as available storage. (IBM-supplied, nonswappable programs include portions of MSRJE, portions or all SSP-ICF subsystem tasks, BSC support for RPG II, and data communications programs). If all programs in storage are swappable, the available storage is equal to user storage.

The job step region size is the maximum amount of storage that can be used by programs that request storage beyond their load member size. The job step region size is used to limit the size of a job step that uses a changing amount of main storage. Some system utilities will require more than the specified size.

Placement: The REGION OCL statement can appear anywhere among the OCL statements, except between a LOAD OCL statement and a RUN OCL statement.

When the REGION OCL statement specifies a job region size, it must come before the first LOAD statement in the job.

Any REGION OCL statement appearing after the first REGION OCL statement in a job or after the first LOAD OCL statement in a job applies only to the next job step. If more than one REGION OCL statement is specified for a job step, the region size specified on the last REGION OCL statement is used.

```
// REGION   SIZE-region size
```

S9020322-0

SIZE specifies the region size in K-bytes (one K-byte equals 1024 bytes). The region size can be any number from 1 through 64, but may not exceed the number of K-bytes of main storage available to the user. The region assigned by the SSP is rounded up to an even number of K-bytes.

Example

PROCA is a first-level procedure. The first two programs run as part of PROCA use 24K bytes of main storage. The third program uses only 14K bytes of main storage. PROCA contains the following statements:

```
// REGION SIZE-24
// LOAD PROG1
// RUN
*
// LOAD PROG2
// RUN
*
// REGION SIZE-14
// LOAD PROG3
// RUN
```

RESERVE OCL Statement

The RESERVE OCL statement specifies the size of a reserved area on disk that is set aside to contain the scratch and job files used by a job. By using a RESERVE statement, you can ensure that enough disk space will be available for the scratch and job files used by the job.

Once an area is reserved, the SSP creates all new scratch and job files for the job in the reserved area. If not enough space is available in the reserved area for a new scratch or job file, the SSP issues an error message and the operator can either choose to attempt to create the file elsewhere on disk or to cancel the job.

At the end of each job step, the SSP automatically compresses the reserved area if any job files were deleted during the step. At the end of the entire job, the SSP automatically frees the reserved area for use by other programs.

If a reserved area exists for the job, the SSP ignores the LOCATION parameter on FILE OCL statements for new scratch and job file.

A job step that runs a multiple requester terminal program or a job step that releases the requesting display station, in effect, starts another job. Therefore, scratch and job files created by those job steps are not created in the reserved area for the job.

Placement: If a RESERVE OCL statement is used, it cannot be placed between the LOAD and RUN OCL statements in a job. Only one RESERVE statement can be used in a job.

```
// RESERVE  BLOCKS-size
```

S9020323-0

BLOCKS specifies the size, in number of blocks, of the reserved area. If the specified number of blocks is not available in the user area on disk, the SSP issues an operator message and waits for the space to become available. If the space does not become available, you can use the Attn key to interrupt and then cancel the job.

Example

Reserve 50 blocks for scratch file FILEA and for job file FILEB.

```
// RESERVE BLOCKS-50
// LOAD PROGA
// FILE NAME-FILEA,RETAIN-S,BLOCKS-20
// FILE NAME-FILEB,RETAIN-J,BLOCKS-30
// RUN
// LOAD PROGB
// FILE NAME-FILEB,RETAIN-J
// RUN
```

RUN OCL Statement

The RUN OCL statement indicates the end of the OCL statements for a job step. After the system reads the RUN statement, it runs the program named in the LOAD statement.

Placement: A RUN OCL statement is needed for each program run on the system. It is the last statement within the set of OCL statements for each job step.

```
// RUN
```

S9020324-0

The RUN OCL statement has no parameters.

Example

To load and run a program called PASTDUE.

```
// LOAD PASTDUE
// RUN
```

SESSION OCL Statement

The SESSION OCL statement assigns an application program to a particular Interactive Communications feature (SSP-ICF) session on a particular subsystem.

Only the statement formats are shown here. See the manuals *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference*, *Interactive Communications Feature: Upline Subsystems Reference*, *Interactive Communications Feature: Finance Subsystem Reference*, and *Interactive Communications Feature: Base Subsystems Reference* for a description of the SESSION statements for the various SSP-ICF subsystems. See the manual *SSP-ICF Guide and Examples* for examples of how the SESSION OCL statement can be used.

Placement: The SESSION OCL statement must be placed between the LOAD and RUN OCL statements.

For the Intra subsystem:

```
// SESSION  LOCATION-name,SYMID-session id [,BATCH-{NO|YES}]
```

S9020325-0

For the BSCEL subsystem:

```
// SESSION  LOCATION-name,SYMID-session id [,PARTNER-{ATTR|NORM}]

         [,SWTYP-{MC|AA|MA}] [,PHONE-member name] [,REFRESH-{YES|NO}]

         [,RESTORE-{NO|YES}] [,RECL-record length] [,BLKL-block length]

         [,RECSEP-separator character] [,ITB-{YES|NO}] [,BLANK-{C|T|N}]

         [,TRANSP-{YES|NO}] [,LIBRARY-name]
```

S9020326-0

For the BSC CCP subsystem:

```
// SESSION  LOCATION-name,SYMID-session id [,SESNADDR-x]

         [,PHONE-member name] [,REFRESH-{YES|NO}] [,RESTORE-{NO|YES}] [,LIBRARY-name]
```

S9020327-0

For the BSC CICS subsystem:

```
// SESSION  LOCATION-name,SYMID-session id [,SESNADDR-x] [,SWTYP-{MC|AA|MA}]

         [,PHONE-member name] [,REFRESH-{YES|NO}] [,RESTORE-{NO|YES}] [,LIBRARY-name]
```

S9020328-0

For the BSC IMS subsystem:

```
// SESSION  LOCATION-name,SYMID-session id [,BATCH-{NO|YES}]

         [,MAXMSG-nnnn] [,PTERM-xxxx]
```

S9020329-0

For the Finance subsystem:

```
// SESSION  LOCATION-name,SYMID-session id,LWSID-logical work station id
```

S9020330-0

For the SNA Upline Facility subsystem:

```
// SESSION  LOCATION-name,SYMID-session id

          [,LWSID-logical work station id] [,APPLID-application id]

          [,HOSTNAME-{IMSRTR}] [,RECL-record length] [,FMHI-{NO }]
                     {IMS   }                        {YES}
                     {CICS  }

          [,MSGPROT-{YES}] [,BATCH-{NO }]
                    {NO }          {YES}
```

S9020331-0

For the BSC 3270 support:

```
// SESSION  LOCATION-name,SYMID-session id [,DATAID-cc,FLDLTH-length]

          [,DEVADDR-xx] [,HOSTNAME-{OTHER}]
                                   {CICS }
                                   {IMS  }
```

S9020332-0

For the Peer subsystem:

```
// SESSION  LOCATION-name,SYMID-session id
```

S9020333-0

For the APPC and APPN subsystems:

```
// SESSION  SYMID-session id,LOCATION-name[,GROUP-session group name]

          [,APPCNET-{NO }]
                    {YES}
```

S9020334-2

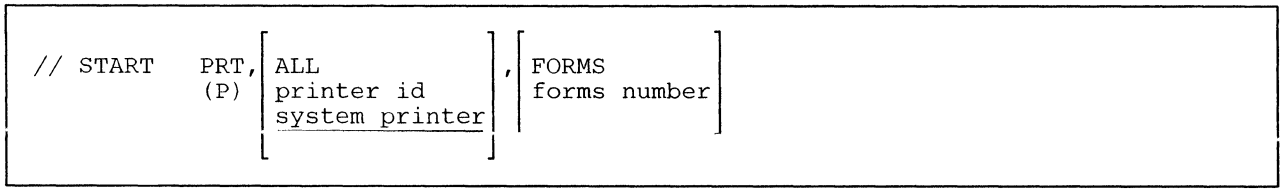
START OCL Statement

The START OCL statement causes the spool writer to be started. If you control one or more printers, you can use the START OCL statement to:

- Start the printing of all spool file entries for a specific printer or for all printers that you control
- Start the printing of all spool file entries with a specific forms number for a specific printer or for all printers that you control
- Start the printing of all spool file entries for a specific printer or for all printers that you control such that those entries using the same forms are printed together

Note: If you are operating the system console or the system service display station, you control all printers. You also control all printers if password security is active and you have system operator authority or higher. If you are operating a subconsole display station, you control those printers for which the display is a subconsole.

Placement: The START OCL statement can be placed anywhere among the OCL statements. You can evoke or submit the START OCL statement to the job queue but only if password security is active and you have system operator authority or higher.



S9020617-0

PRT or P specifies that a spool writer for a printer is to be started; that is, spool file entries can be printed when the printer is available. The PRT parameter can be used to start the printing of entries on the spool file after IPL if the automatic start function (also called the autowriter) was not selected during system configuration. The PRT parameter can also be used to start the spool writer(s) after a STOP PRT OCL statement or a STOP PRT control command is entered.

If no second parameter is entered, the system printer is assumed. If no third parameter is entered, the spool writer prints the available entries according to their position in the spool file.

ALL specifies that the spool writers for all printers you control are to be started.

printer id specifies the 2-character ID of the printer for which the spool writer is to be started. If no parameter is specified, the system printer is assumed. You must control the printer.

FORMS specifies that the spool writer is to print all available spool file entries that require the forms currently being used in the printer, regardless of the entry's position in the spool file. After all such entries are printed, the operator controlling the printer is prompted to change the forms, and all entries using the next group of forms are printed. This can reduce the number of operator messages to change the forms.

forms number specifies the 1- to 4-character forms number to be used by the spool writer. Only the spool file entries using the specified forms number are printed.

Example 1

This example shows how to start the spool writer for printer P1 in order to start printing all spool file entries with forms number 1324 (that is, they are to be printed on printout form 1324).

```
// START PRT,P1,1324
```

or:

```
// START P,P1,1324
```

Example 2

This example shows how to start the spool writer for the system printer. All spool file entries that require the forms currently being used by the system printer will be printed.

```
// START P,,FORMS
```

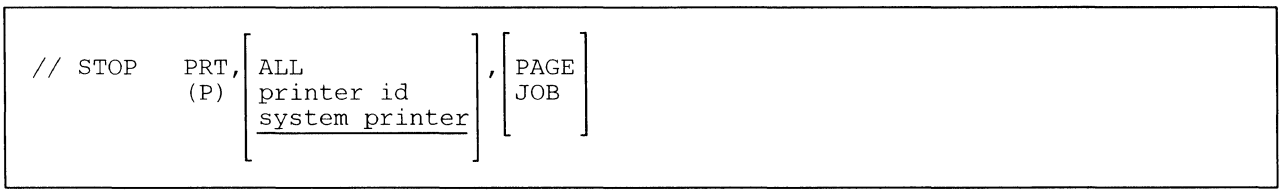
After all such entries are printed, the operator controlling the printer will be prompted to change the forms, and the next group of spool file entries will be printed.

STOP OCL Statement

If you control one or more printers, you can use the STOP OCL statement to stop the printing of all entries from the spool file for a specific printer or for all printers that you control. Printing can be stopped immediately, at the end of the currently printing page, or at the end of the currently printing spool file entry.

Note: If you are operating the system console or a system service display station, you control all printers. You also control all printers if password security is active and you have system operator authority or higher. If you are operating a subconsole display station, you control those printers for which the display is a subconsole.

Placement: The STOP OCL statement can be placed anywhere among the OCL statements. You can evoke or submit the STOP OCL statement to the job queue but only if password security is active and you have system operator authority or higher.



S9020618-0

PRT or P specifies that the printing of spool file entries is to stop. If a spool file entry is being printed and a third parameter is not specified, printing stops immediately. You can resume the printing by entering the START OCL statement, the START PRT control command, or the RESTART control command.

ALL specifies that printing is to stop for all printers that you control.

printer id specifies the 2-character ID of the printer for which the spool writer is to be stopped. If no printer ID is specified, the system printer is assumed. You must control the printer.

PAGE stops the spool writer when it has completed printing the current page.

JOB stops the spool writer when it has completed printing the current copy of the spool file entry.

Example

This example shows how to stop the spool writers for all printers that you control when the spool file entry currently printing on each of them is completed.

```
// STOP PRT,ALL,JOB
```

or:

```
// STOP P,ALL,JOB
```

SWITCH OCL Statement

The SWITCH OCL statement sets one or more of the user program status indicator (UPSI) switches for the display station to on (1, one) or off (0, zero). The switch setting remains in effect until one of the following occurs:

- Another SWITCH statement is processed
- The display station session is ended
- A program changes the setting of any of the indicators

All switches are set to off when a display station session begins, that is, when an operator signs on.

A job placed on the job queue uses a copy of the switches as they existed when the job was placed on the queue. A job that was EVOKED or a job RELEASED using ATTR OCL will also use a copy of the switches for the display station as they existed when the job was evoked or released. To determine the switch settings from a procedure, see the "SWITCH (Switches) Condition" on page 3-48.

If an SSP procedure changes the setting of a switch, the switch returns to its original setting when the SSP procedure ends. A set of switches also exists for each running multiple requester terminal (MRT) procedure.

Placement: The SWITCH OCL statement can appear anywhere among the OCL statements.

```
// SWITCH   switch settings
```

S9020335-0

switch settings specifies how the switches are to be set, and consists of 8 characters, one for each of the eight switches. The first, or leftmost, character gives the setting of switch 1; the second character gives the setting of switch 2; and so on.

The parameter must always contain 8 characters. For each indicator, one of the following characters must be used:

Character	Meaning
0 (zero)	Set the indicator off
1 (one)	Set the indicator on
X	Leave the indicator as it is

Example

This example:

```
// SWITCH 1X0110XX
```

causes the following results:

Indicator	Switch Setting	Result
1	1	Set on
2	X	Unaffected
3	0	Set off
4	1	Set on
5	1	Set on
6	0	Set off
7	X	Unaffected
8	X	Unaffected

SYSLIST OCL Statement

The SYSLIST OCL statement changes the printing or displaying of system list output. System list output is the output generated by all SSP utility programs except for the data communications utility programs and service aids.

The SYSLIST statement causes the system list output to be:

- Listed on the printer that was assigned to the display station during system configuration
- Listed on one of the other printers
- Displayed at the display station
- Not listed at all

The SYSLIST statement remains in effect until the display station session is ended or until it is changed by:

- Another SYSLIST OCL statement
- The SYSLIST procedure

SYSLIST

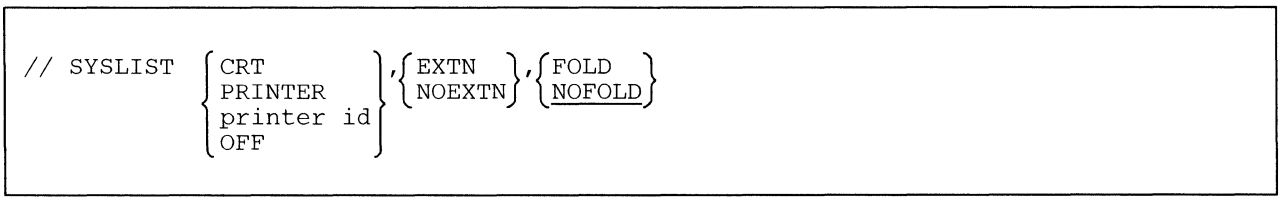
A PRINTER OCL statement with NAME-$SYSLIST specified further controls the system list output for one job step. In this case, the device specified on the PRINTER statement is used, not the device specified on the SYSLIST statement. See the "PRINTER OCL Statement" on page 5-83 for more information.

Notes:

1. *If the SYSLIST OCL statement is run during inquiry mode (the operator pressed the Attn key), the change made by the SYSLIST OCL statement is in effect only during inquiry mode.*

2. *The SYSLIST OCL statement does not change the system list device if debugging is turned on. For more information, refer to the "DEBUG OCL Statement" on page 5-28.*

Placement: The SYSLIST statement can appear anywhere among the OCL statements.



S9020336-0

CRT specifies that the system list output is to be displayed at the requesting display station. Depending on the model of the display station, up to 21 lines are displayed at a time. At least the first 80 columns of the output are displayed.

After each system list display has been shown, the operator can request that the next set of lines be displayed or that the display be ended. If the operator enters a blank, the number of lines shown on the next display will be the same as the number of lines shown on the previous display. If the operator enters 0, the display will be ended. If the operator enters any characters other than a blank or 0 through 18 (21 for the 3180 Model 2 display station), the next set of lines is displayed.

System output from multiple requester terminal programs is printed on the system printer. System list output from programs that release the requesting display station, and from programs that are run from the job queue is printed on the session printer.

PRINTER specifies that system list output is to be printed on the printer that was assigned to the display station during system configuration or assigned by the SET procedure or the $SETCF utility program.

printer id specifies that system list output is to be printed on the printer with the specified printer ID. You can use the STATUS WORKSTN command to determine the printer IDs. To print output on the system printer, specify the printer ID of the system printer.

OFF specifies that no system list output will be displayed or printed.

EXTN specifies that the extended characters should be printed or displayed in the system list output.

NOEXTN specifies that the extended characters should not be printed or displayed in the system output. The system-defined ideographic character is listed for any extended characters.

Note: The EXTN and NOEXTN parameters are for the ideographic version of the SSP and are ignored for nonideographic systems. When a session begins, extended character processing for system list output is assumed. During the session, the EXTN and NOEXTN parameters can be used to turn on and turn off extended character processing for system list output.

FOLD specifies that system list output displayed at the display station is not to be truncated if it exceeds the column width of the display station. Instead, data after column position 75 is continued on the next line of the display (for the 3180 Model 2 display station, data after column position 130 is continued on the next line).

NOFOLD specifies that system list output displayed at the display station is truncated if it exceeds the column width of the display station. If no parameter is specified, NOFOLD is assumed.

Example

The following is an example of assigning printer P2 to list system list output.

```
// SYSLIST P2
```

VARY OCL Statement

The VARY OCL statement allows you to change the status of the following from online to offline and from offline to online:

- Display stations
- Printers
- The system printer
- The diskette drive
- Communications controllers
- Communications lines
- Tape drives

Devices that are offline cannot be used by operators or programs. The VARY OCL statement cannot be used to take offline a device that is allocated to a program or signed on. If you have system operator authority or higher, you can enter the VARY OCL statement from any type of display station.

Placement: The VARY OCL statement can appear anywhere among the OCL statements except between the LOAD and the RUN OCL statements.

```
//  VARY {ON }[,{display id                    }]
         {OFF}  {printer id                    }
                {                              }
                {PRT                           }
                {(P)                           }
                {                              }
                {I1                            }
                {                              }
                {T1                            }
                {T2                            }
                {TC                            }
                {                              }
                {controller id                 }
                {,line number                  }
                {controller id,line number     }
```

S9020337-2

ON causes the specified device to be placed online.

OFF causes the specified device to be placed offline.

display id specifies the 2-character ID of a display station to be placed online or offline. You can use the STATUS WORKSTN command to determine the display station IDs.

A display station must be signed off to be placed offline. If the device being varied offline is a subconsole, any messages that were sent to the subconsole are sent to the system console.

printer id specifies the 2-character ID of a printer to be placed online or offline. You can use the STATUS WORKSTN command to determine the printer IDs.

The spool writers for a printer must be stopped or complete in order to place that printer offline. You can use the STATUS WRT command to determine whether a spool writer is stopped or complete.

PRT or P specifies that the system printer is to be placed online or offline.

I1 specifies that the diskette drive is to be placed online or offline.

controller id specifies the 3-character ID of a communications controller. All remote display stations and printers associated with the specified controller are placed online or offline.

line number specifies the communications line number. When this parameter is entered without the controller ID parameter, all controllers, display stations, and printers associated with this line are placed online or offline. When this parameter is entered with the controller ID parameter on the VARY ON OCL statement, the specified device is placed online on the specified communications line.

controller id,line number is valid only for switched communications lines. When these parameters are entered on the VARY ON OCL statement, the specified device is placed online on the specified communications line.

T1 specifies that tape drive 1 is to be placed online or offline.

T2 specifies that tape drive 2 is to be placed online or offline.

TC specifies that the tape cartridge drive is to be placed online or offline.

Example

To place all remote display stations and printers associated with controller C01 online, enter:

```
// VARY ON,C01
```

WAIT OCL Statement

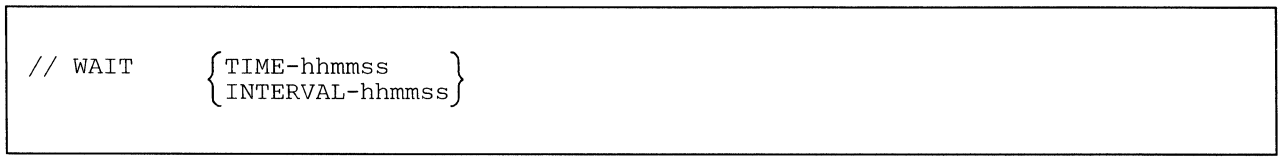
The WAIT OCL statement causes a job to wait until a certain time of day, or until a certain period of time has passed by. Once a WAIT OCL statement is processed, the job will not resume processing until the specified condition is met. The job may, of course, be ended by either of the following:

- The operator that started the job can press the Attn key and take an option to cancel the job.
- The system operator can use the CANCEL command to cancel the job.

Use caution with this statement. For example, if you place a job containing this statement on the job queue, you may cause that job to prevent other jobs on the queue from processing.

When the WAIT OCL statement is processed, any system resources (for example: disk files, the diskette drive, or a printer) allocated to a waiting job are treated as if they are owned by a never-ending program (NEP); see the "ATTR OCL Statement" on page 5-11 for more information about NEPs. This can prevent other programs from waiting for the resources owned by the waiting job. When the time of day is reached, all resources are returned to their previous states.

Placement: The WAIT statement can appear anywhere among the OCL statements. The WAIT statement can only be in a procedure; that is, it cannot be entered at the keyboard.



S9020338-0

TIME specifies the job is to stop and wait until the specified time of day occurs. **hhmmss** specifies the time of day when the job is to resume processing. A 6-digit number must be entered.

The actual resume time is dependent on system work load and processing time. For example, if the WAIT OCL statement says // WAIT TIME-080000, and the job is submitted at 075959, it is possible that while the job should start at 080000, it may wait until 080000 the next day due to processing time. Split second accuracy cannot be guaranteed on a busy system.

hh specifies the hour. The value must be from 00 through 24. The values 00 through 12 represent 12 midnight through 12 noon. The values 13 through 24 represent 1 p.m. through 12 midnight. Both 000000 and 240000 are considered to be midnight.

mm specifies the minute. The value must be from 00 through 59.

ss specifies the second. The value must be from 00 through 59.

240000 is the maximum value that can be specified. Therefore, if 24 is the specified hours, the minutes and seconds values must be 00.

INTERVAL specifies the amount of time a job is to wait before resuming processing. **hhmmss** specifies how long the job is to wait before it is to resume processing. A 6-digit number must be entered.

hh specifies the hours. The value must be from 00 through 24.

mm specifies the minute. The value must be from 00 through 59.

ss specifies the second. The value must be from 00 through 59.

240000 is the maximum value that can be specified. Therefore, if 24 is the specified hours, the minutes and seconds values must be 00.

Example 1

The program PROGA can only be run at 9 p.m. (time 210000). The procedure to run PROGA could contain the following statements.

```
// WAIT TIME-210000
// LOAD PROGA
// RUN
```

Example 2

The procedure is to wait 1 minute between job steps.

```
// LOAD PROG1
// RUN
// WAIT INTERVAL-000100
// LOAD PROG2
// RUN
```

Example 3

This example shows how to make a procedure wait until 4 p.m. If the time is already greater than 4 p.m., the WAIT statement is not processed. For example, if the procedure were run at 5 p.m., the procedure PROC1 would be run immediately; if the ?TIME? test were not performed, procedure PROC1 would not be run until 4 p.m. of the **next day**.

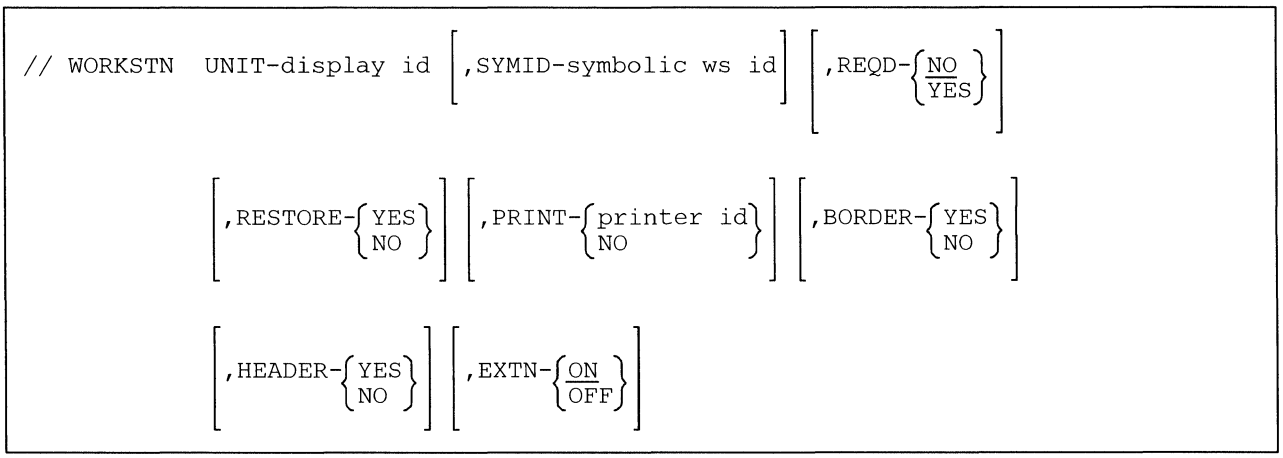
```
// IFF ?TIME?>160000  WAIT TIME-160000
PROC1
```

WORKSTN OCL Statement

The WORKSTN OCL statement supplies the system with information about a display station used by a program. A display station can be assigned for use by a program in one of the following ways:

- The requesting display station is automatically acquired, unless the program is placed on the job queue or evoked. No WORKSTN OCL statement is required for the requesting display station.
- A WORKSTN OCL statement specifying REQD-YES causes the system to acquire the specified display station.
- The program can do the acquire operation. If the program acquires a display station, a WORKSTN statement is optional. Even though the program acquires the display station, you might want to use a WORKSTN statement so that you can specify information about the display station.

Placement: The WORKSTN statement can appear anywhere among the OCL statements. All parameters, except the RESTORE-YES parameter, take effect immediately and remain in effect until the end of the current job step. The RESTORE-YES parameter does not take effect until the display station is used within the job step (either a PROMPT OCL statement using the display station ID must be encountered after the WORKSTN statement and before the RUN statement, or the program must do display station operations with the display station ID). The RESTORE-YES parameter remains in effect until the display station is released or until the job ends. If the WORKSTN statement appears after the last job step in a job, the WORKSTN statement parameters (including RESTORE-YES) will take effect.



S9020339-0

UNIT specifies the 2-character work station ID of the display station to be used by the program. You can use the STATUS WORKSTN command to determine the display station IDs.

SYMID specifies the symbolic work station ID used by the program. The symbolic work station ID is a 2-character ID (the first character must not be numeric) that the program uses to refer to the display station. The symbolic ID cannot be the same as the work station ID assigned to any other display station or printer on the system. The symbolic work station ID can be the same as the work station ID specified by the UNIT parameter. If SYMID is not specified, the symbolic work station ID is assumed to be the same as the work station ID specified in the UNIT parameter.

REQD specifies whether the SSP is to acquire the display station or whether the application program is to acquire the display station (if a display station needs to be acquired). If the display station is the requesting display station, it does not need to be acquired.

NO specifies that the application program is to acquire the display station. If the REQD parameter is not specified, REQD-NO is assumed.

YES indicates that the SSP is to acquire the display station.

RESTORE specifies:

- Whether the Command display (or the menu display if a menu is being used) should be restored when a Command display station is released by the program or when the job ends.
- Whether the Standby display should be displayed when a data display station is released by the program or when the job ends.

YES indicates that the last information displayed by the program is replaced by the Command or menu display on a command display station or by the Standby display on a data display station.

NO indicates that the last information displayed by the program remains on the display screen. The operator must press the Enter key to restore the Command, menu, or Standby display.

If RESTORE is not specified and the last display station operation was an input to the program, the Command, menu, or Standby display is restored. If RESTORE is not specified and the last display station operation was an output from the program, the command, menu, or standby display is not restored.

PRINT specifies the printer to be used when the operator presses the Print key. If the specified printer is already being used when the operator presses the Print key, an error message is displayed and the operator must press the Error Reset key; no information is printed. If spooling is active, Print key output will be spooled.

printer id specifies the 2-character printer ID of the printer to receive the Print key output.

NO specifies that the Print key is to be ignored for the display station.

If PRINT is not specified, system uses the printer assigned to the display station that submitted the job.

The PRINT parameter is in effect only while the display station is allocated to the program. If the operator presses the Print key while the display station is in command mode, the system uses the printer assigned to the display station.

BORDER specifies whether a border is to be printed around the display image when the Print key is pressed.

YES specifies that a border is to be printed.

NO specifies that a border is not to be printed.

HEADER specifies whether a heading is to be printed at the top of the page containing the display image when the Print key is pressed.

YES specifies that a heading is to be printed.

NO specifies that a heading is not to be printed.

EXTN specifies whether or not the extended character task is called to process any extended characters in the data stream. The EXTN parameter is for the ideographic version of the SSP and is ignored for nonideographic systems.

ON specifies that the extended character task is called to process any extended characters in the data stream to be displayed.

OFF specifies that the extended character task is not called; the system-defined default ideographic character is displayed for any extended characters in the data stream. If the EXTN parameter is not specified, ON is assumed.

Note: The EXTN parameter does not affect characters entered from the keyboard at the display station.

Example

Display station W3 is to be acquired for use by the program PROGA. The program uses the symbolic name A1. The Command, menu, or Standby display is to be restored when the display station is released. If the operator presses the Print key, displayed information is to be printed on printer P2.

```
// LOAD PROGA
// WORKSTN UNIT-W3,SYMID-A1,REQD-NO,RESTORE-YES,PRINT-P2
// RUN
```

/* (End Of Data) Statement

The /* statement indicates the end of data entered from the keyboard or the end of in-line source data.

Placement: A /* statement must be the last line of data entered from the keyboard.

```
/*
```

S9020340-0

The /* statement has no parameters and must be entered in columns 1 and 2.

Example

This example shows an end of data statement.

```
/*
```

Chapter 6. Control Commands

This chapter describes the System/36 control commands which are supplied as part of the SSP. Control commands are used to control the system, the printers, and the display stations. The commands can be entered from the keyboard or selected by a menu option. They cannot be coded as part of a procedure. The control commands are divided into three groups:

- Control commands anyone can enter.
- Control commands you can enter if you control one or more printers. The display station that controls one or more printers is a **subconsole**.
- Control commands the system operator can enter from the system console. If you are using a system service display station, you can enter all the system operator control commands except:

  ASSIGN
  CANCEL SESSION for the system console
  CONSOLE
  START SERVICE
  START SYSTEM
  STOP SERVICE
  STOP SYSTEM

  If the system service display station is also an alternative system console, the CONSOLE command can be used.

The description of each control command contains:

- What the command can do and who can use it.
- The syntax format of the command. For a description of the rules used to describe formats, see "Conventions Used for Describing Syntax Formats" on page 1-3.
- Descriptions of the control command's parameters.
- One or more examples of how to use the control command.

For information on operating the System/36, see the manual *Operating Your System* for your system unit.

When you are signed on to the System/36, you can use the system help support to perform a task. The system help support is made up of menus, prompt displays, and help text. The menus allow you to select a task you want to perform. When you select an item from a menu, either:

- Another menu is displayed (as you select options, each one gets more specific about the task you want to perform).
- A prompt display for a procedure or control command is shown.

The prompt displays allow you to run commands to perform the task. The help text explains the menus, the menu options, the procedures and commands, and the parameters for the procedures and commands.

For more information about the system help support, see the "HELP Procedure" on page 4-199.

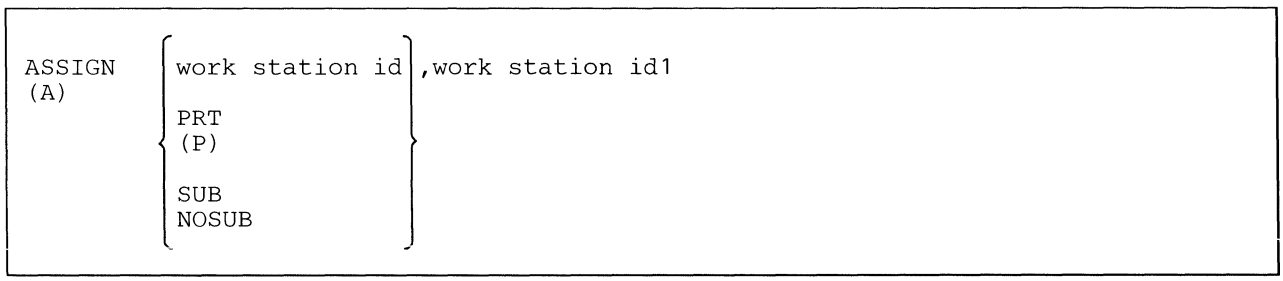
ASSIGN Control Command

If you are the system operator, you can use the ASSIGN control command to temporarily:

- Exchange the work station IDs of two display stations or two printers. The original work station IDs are restored when an IPL is performed.
- Assign a printer as the system printer.
- Activate or deactivate subconsole support for a subconsole display station.

The ASSIGN control command cannot be entered from a system service display station.

Note: The display stations or printers being exchanged must be offline. If subconsole support is being activated or deactivated, the subconsole being activated or deactivated must also be offline. See the "VARY Control Command" on page 6-53 for information about placing a device offline.



S9020341-0

work station id specifies the work station ID of a display station or printer. After the SSP processes the ASSIGN command, the display station or printer will have the work station ID specified by work station id1.

PRT or P specifies that the printer indicated by work station id1 is to be assigned as the system printer.

SUB specifies that the subconsole support for the subconsole display station specified by work station id1 is to be activated.

ASSIGN SUB is needed only if an ASSIGN NOSUB control command has been issued for that subconsole display station.

NOSUB specifies that subconsole support for the subconsole display station specified by work station id1 is to be deactivated. It will still function as a display station, but you will not be able to control printers.

work station id1 specifies:

- The 2-character ID of a display station or printer that will temporarily exchange IDs with the display station or printer identified by work station id.
- If SUB or NOSUB is entered, the display station ID that will have subconsole support either activated or deactivated.
- If PRT is entered, the ID of a printer that is to be assigned as the system printer.

Example 1

Display station W1 is not working; to exchange display station IDs with the display station identified as W3, the system operator enters:

```
VARY OFF,W1
VARY OFF,W3
ASSIGN W3,W1
VARY ON,W1
```

or:

```
V OFF,W1
V OFF,W3
A W3,W1
V ON,W1
```

Example 2

To assign printer P2 as the system printer (which is currently P1), the system operator enters:

```
VARY OFF,P1
VARY OFF,P2
ASSIGN PRT,P2
VARY ON,P2
VARY ON,P1
```

Example 3

To deactivate subconsole support for subconsole display station W3, the system operator enters:

```
VARY OFF,W3
ASSIGN NOSUB,W3
VARY ON,W3
```

Example 4

To activate subconsole support for subconsole display station W3, the system operator enters:

```
VARY OFF,W3
ASSIGN SUB,W3
VARY ON,W3
```

CANCEL Control Command

You can use the CANCEL control command to:

- Cancel one or all of your spool file entries
- Cancel all of your spool file entries with a specific forms number
- Cancel your jobs on the job queue

If you control one or more printers, you can also use the CANCEL control command to:

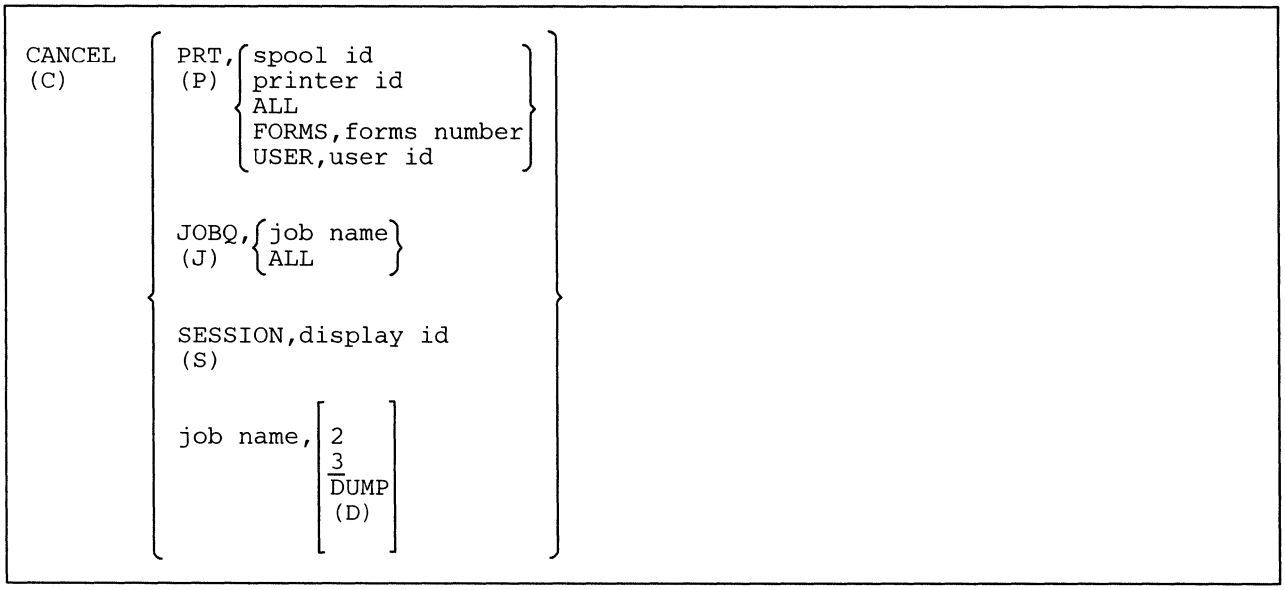
- Cancel one or all spool file entries for a specific printer that you control
- Cancel all spool file entries for all printers you control
- Cancel all spool file entries with a specific forms number for all printers that you control
- Cancel all spool file entries with a specific user ID for all printers that you control

Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are the system operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.

If you are the system operator or a system service display station operator you can also use the CANCEL control command, from any type of display station, to:

- Cancel the following on the spool file:
  - One entry
  - All entries
  - All entries for a specific printer
  - All entries with a specific forms number
  - All entries with a specific user ID
- Cancel any one or all jobs on the job queue
- Cancel the session of a display station (that is, sign another display station off the system)
- Cancel a currently running job

CANCEL



S9020342-1

PRT or P specifies that one or more entries are to be canceled from the spool file.

spool id specifies the 6-character ID of the spool file entry to be canceled. The STATUS PRT command can be used to determine the spool file IDs.

printer id specifies the 2-character ID of a printer for which all spool file entries are to be canceled. You must control the printer.

ALL specifies that all spool file entries for all the printers you control are to be canceled. If you do not control any printers, only your spool file entries are canceled.

FORMS specifies that all spool file entries with the specified forms number are to be canceled. If you do not control any printers, only your spool file entries with the specified forms number are canceled. If you control any printers, all entries with the specified forms number are canceled for all printers that you control.

forms number specifies the 1- to 4-character forms number of the printout form used for the spool file entries that are to be canceled. You can use the STATUS PRT control command to determine the forms number to be used for each spool file entry.

USER specifies that all spool file entries with the specified user ID are to be canceled. If you specify your own user ID, all of your spool file entries are canceled. If you specify a user ID other than your own, all spool file entries with the specified user ID are canceled for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of all spool file entries that are to be canceled. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

JOBQ or J specifies that one or more jobs are to be canceled from the job queue.

job name specifies the 8-character name of a job, on the job queue, to be canceled. The STATUS JOBQ command can be used to display the names of jobs on the job queue.

ALL specifies that all jobs on the job queue are to be canceled. **ALL can be entered only** at the system console or a system service display station.

SESSION or S specifies that the session for a display station is to be canceled. The specified display station is signed off the system. **This parameter can be entered only** at the system console or a system service display station. The system service display station cannot sign off the system console.

display id specifies the 2-character ID of the display station to be signed off. You can use the STATUS WORKSTN control command to determine the display station IDs.

job name specifies a currently running job to be canceled. The second parameter specifies how the job is to be ended. The STATUS USERS command can be used to display the names of jobs currently running. **This parameter can be entered only** at the system console or at a system service display station.

2 specifies that a controlled cancel is to be taken. Any files, either new or old, being used by the job are closed. Any remaining job steps are not processed.

3 specifies that an immediate cancel is to be taken. Any old files being used by the job are closed. New files created by the current job step are lost. Any remaining job steps are not processed. See the note below.

DUMP or D specifies that an immediate cancel with a task dump is to be taken. Any files being used by the job are not closed. The main storage assigned to the job is written to a system dump area on disk. Any remaining job steps are not processed. See the note below.

Note: If 3 or DUMP (D) are specified when a job is canceled, the following can occur:

- *Any new files created by the program that was running are lost.*
- *If the program that was running added records to a nonshared file, the added records are lost.*

Example 1

To cancel spool file entry SP0010, enter:

```
CANCEL PRT,SP0010
```

or:

```
C P,SP0010
```

Example 2

To cancel all spool file entries with user ID AA120199, the system operator enters:

```
CANCEL PRT,USER,AA120199
```

or:

```
C P,USER,AA120199
```

Example 3

To cancel job W3123412 from the job queue, enter:

```
CANCEL JOBQ,W3123412
```

Example 4

To cancel all jobs on the job queue, the system operator enters:

```
CANCEL JOBQ,ALL
```

Example 5

To sign display station W2 off the system, the system operator enters:

```
CANCEL SESSION,W2
```

or:

```
C S,W2
```

Example 6

To immediately cancel job W2001820, which is currently running, the system operator enters:

```
C W2001820
```

or:

```
C W2001820,3
```

CHANGE Control Command

You can use the CHANGE command to change:

- The number of copies to be printed for one or all of your spool file entries
- The number of copies to be printed for all of your spool file entries with a specific forms number
- The defer status for one of your spool file entries
- The forms number to be used for one or all of your spool file entries
- The forms number to be used for all of your spool file entries with a specific forms number
- The printer used for one or all of your spool file entries
- The printer used for all of your spool file entries with a specific forms number

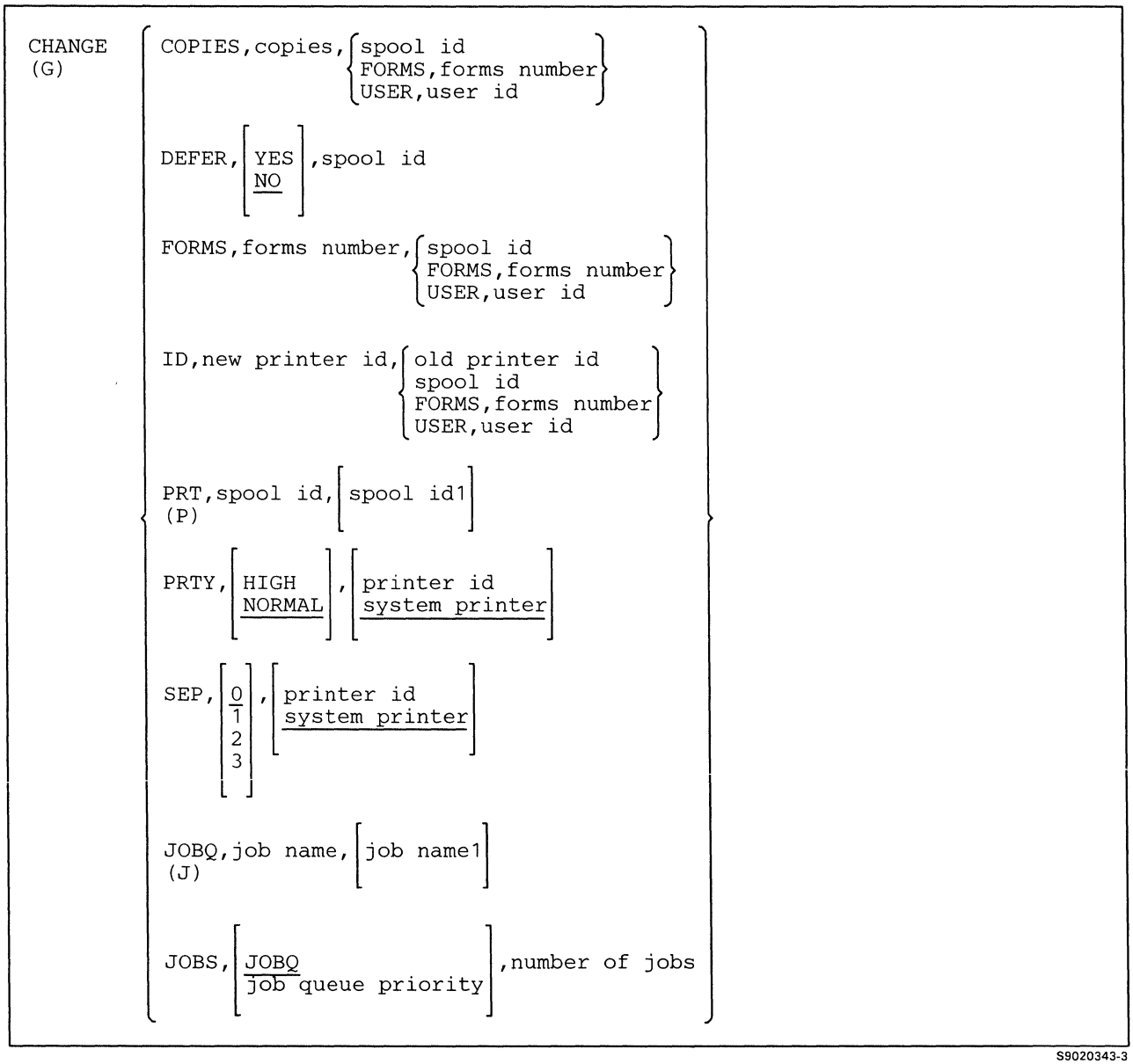
If you control one or more printers, you can also use the CHANGE command to change:

- The number of copies to be printed for a spool file entry for a printer you control
- The number of copies to be printed for all spool file entries with a specific forms number for all printers you control
- The number of copies to be printed for all spool file entries with a specific user ID for all printers you control
- The defer status of a spool file entry for a printer you control
- The forms number to be used for a spool file entry for a printer you control
- The forms number to be used for all spool file entries with a specific forms number for all printers you control
- The forms number to be used for all spool file entries with a specific user ID for all printers you control
- The printer used for a spool file entry for a printer you control
- The printer used for all spool file entries for a printer you control
- The printer used for all spool file entries with a specific forms number for all printers you control
- The printer used for all spool file entries with a specific user ID for all printers you control
- The position of a spool file entry for a printer you control
- The priority of the spool writer that prints entries for a printer you control
- The number of separator pages to be placed between the printed output for a printer you control

| *Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.*

If you are the system operator or a system service display station operator, you can also use the CHANGE command to change:

- The position of a job on the job queue
- The number of job queue jobs the system will keep active



COPIES specifies that the number of copies of printed output for one or more spool file entries is to be changed.

copies specifies the number of copies to be printed for the spool file entry or entries. You can specify any number from 1 through 255.

spool id specifies the 6-character ID of a spool file entry. The STATUS PRT command can be used to determine the spool file IDs.

FORMS specifies that the number of copies of all spool file entries with the specified forms number is to be changed. If you do not control any printers, only your spool file entries with the specified forms number are changed. If you do control any printers, all entries with the specified forms number are changed for all printers that you control.

forms number specifies the 1- to 4-character forms number of the spool file entries for which the number of copies is to be changed. You can use the STATUS PRT control command to determine which form is to be used for each spool file entry.

USER specifies that the number of copies of all spool file entries with the specified user ID is to be changed. If you specify your own user ID, all of your spool file entries are changed. If you specify a user ID other than your own, all spool file entries with the specified user ID are changed for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries for which the number of copies is to be changed. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

DEFER specifies that the defer status for a particular spool file entry is to be changed.

YES indicates that the spool file entry should not start printing until the program has completed the spool file entry.

NO indicates that the spool file entry can start printing before the program has completed the spool file entry. If no parameter is specified, NO is assumed.

spool id specifies the 6-character ID of a spool file entry. The spool file entry must not be completed yet by the program creating it. The STATUS PRT command can be used to determine the spool file IDs.

FORMS specifies that the forms number to be used for one or more spool file entries is to be changed. You can use the STATUS WRT control command to determine which forms are currently being used by the printer. The STATUS PRT control command identifies which forms are to be used for each spool file entry.

forms number specifies the new forms number of the printout form to be used; from 1 to 4 characters can be entered.

spool id specifies the 6-character ID of a spool file entry. The spool file entry must not be printing. The STATUS PRT command can be used to determine the spool file IDs.

FORMS specifies that the printout form for all spool file entries with the specified forms number is to be changed. If you do not control any printers, only your spool file entries with the specified forms number are changed. If you do control any printers, all entries with the specified forms number are changed for all printers that you control.

forms number specifies the current 1- to 4-character forms number of the spool file entries for which the printout form is to be changed.

USER specifies that the printout form for all spool file entries with the specified user ID is to be changed. If you specify your own user ID, all of your spool file entries are changed. If you specify a user ID other than your own, all spool file entries with the specified user ID are changed for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries for which the printout form is to be changed. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

ID specifies that the printer to be used for one or more spool file entries is to be changed.

Note: If you direct your printed output to another printer with different printer characteristics, this may result in printing or programming errors, or your output may not print properly. (DW/36 jobs will fail.)

new printer id specifies the 2-character ID of the new printer to be used.

old printer id specifies the 2-character ID of the current printer. All spool file entries to be printed using this printer will use the new printer. If a spool file entry is currently printing on the old printer, it will continue to print on that printer. You must control this printer.

spool id specifies the 6-character ID of a spool file entry. The STATUS PRT command can be used to determine the spool file IDs. If a spool ID is specified in the third parameter position, that spool file entry will use the new printer. The spool file entry must not be printing.

FORMS specifies that the new printer is to be used by all spool file entries with the specified forms number. If you do not control any printers, only your spool file entries with the specified forms number are changed. If you do control any printers, all entries with the specified forms number are changed for all printers that you control.

forms number specifies the 1- to 4-character forms number of the spool file entries for which the printer is to be changed. You can use the STATUS PRT control command to determine the forms number.

USER specifies that the new printer is to be used by all spool file entries with the specified user ID. If you specify your own user ID, all of your spool file entries are changed. If you specify a user ID other than your own, all spool file entries with the specified user ID are changed for all printers that you control. You must control one or more printers to specify a user ID other than your own.

user id specifies the 8-character user ID of the spool file entries for which the printer is to be changed. You can use the STATUS PRT or STATUSF PRT control commands to determine the user ID.

PRT or P specifies that the position of an entry on the spool file is to be changed.

spool id specifies the 6-character ID of the spool file entry to be moved. The STATUS PRT command can be used to determine the spool file IDs.

spool id1 specifies the 6-character ID of a spool file entry. The *spool id* is moved to the position following *spool id1*, and is given the same priority as spool id1. If spool id1 is not specified, the spool file entry being changed is moved to the front of the spool file and is given a priority of 5. For information about spool file entry priorities, see the *Concepts and Programmer's Guide.*

PRTY specifies that the processing priority for the spool writer for the specified printer is to be changed. For information about job processing priority, see the *Concepts and Programmer's Guide.*

HIGH indicates that the spool writer program is to have high processing priority. This can cause the output to be printed faster. Note that while the spool writer is running faster, other jobs on the system may be running slower.

NORMAL indicates that the spool writer program is to have normal processing priority. If no parameter is specified, NORMAL is assumed.

printer id specifies the 2-character ID of the printer whose spool writer priority is to be changed. You must control this printer. If no parameter is specified, the system printer is assumed.

SEP specifies that the number of separator pages to be printed between spool file entries is to be changed. Either 0, 1, 2, or 3 separator pages can be specified. If no parameter is specified, 0 (zero) is assumed. Separator pages are used to separate the output for different print files.

printer id specifies the 2-character ID of the printer whose number of separator pages is to be changed. You must control this printer. If no parameter is specified, the system printer is assumed.

JOBQ or J specifies that the position of a job on the job queue is to be changed.

Note: This parameter can be entered only at the system console or a system service display station.

job name specifies the 8-character name of the job to be changed. You can use the STATUS JOBQ control command to determine the job names.

job name1 specifies the 8-character name of a job on the job queue. The *job name* is moved to the position following *job name1*. The *job name* assumes the same job queue priority as *job name1*. If *job name1* is not specified, *job name* is moved to the front of the job queue and is given a job queue priority of 5. See the "JOBQ Control Command" on page 6-21 for more information on job queue priority.

JOBS specifies that the number of jobs to be run at once from the job queue or a job queue priority is to be changed.

JOBQ specifies that the number of jobs to be run at once from the job queue is to be changed.

job queue priority specifies that the number of jobs to be run at once from the given job queue priority is to be changed.

number of jobs specifies the number of jobs to be run from the job queue or specified job queue priority.

Notes:

1. *The values entered by this command remain in effect until the next CHANGE JOBS command or until the job queue file is rebuilt.*

2. *This parameter may only be entered at the system console or system service device.*

Example 1

To change spool file entry SP0011 so that five copies of its output are printed, the operator enters:

```
CHANGE COPIES,5,SP0011
```

or:

```
G COPIES,5,SP0011
```

Example 2

To change the forms number to 6987 for all spool file entries with current forms number 1324, the system operator enters:

```
CHANGE FORMS,6987,FORMS,1324
```

or:

```
G FORMS,6987,FORMS,1324
```

Example 3

To cause the spool writer for printer P2 to run at high priority, the operator controlling P2 enters:

```
CHANGE PRTY,HIGH,P2
```

or:

```
G PRTY,HIGH,P2
```

Example 4

To cause no separator pages to be printed between jobs at printer P3, the operator controlling P3 enters:

```
CHANGE SEP,0,P3
```

or:

```
G SEP,,P3
```

Example 5

To move job W3083001 on the job queue so that it follows job W1051050, the operator enters:

```
CHANGE JOBQ,W3083001,W1051050
```

or:

```
G J,W3083001,W1051050
```

CONSOLE Control Command

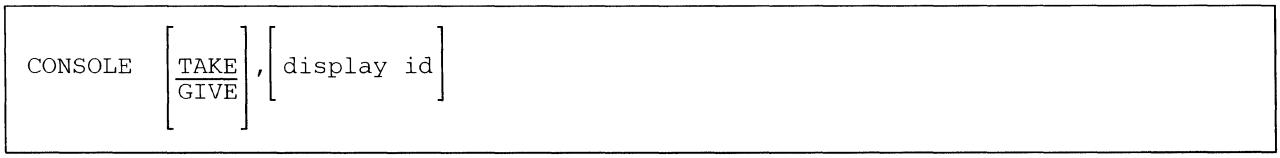
You can use the CONSOLE control command to transfer the system console function from the configured system console to an alternative system console. You may specify local and remote displays as alternative system consoles when you configure your system. See the manual *Changing Your System Configuration* for more information. Any display attached through the Distributed Host Command Facility (DHCF) support is automatically an alternate system console. For Display Station Pass-Through (DSPT) displays:

- Any System/38 attached display is automatically an alternate system console for a remote System/36.
- System/36 attached displays act as alternate system consoles for a remote System/36 if they are configured as the system console or an alternate system console for the local System/36.

If password security is active, the CONSOLE command must be entered by an operator assigned the security classification of system operator or higher.

The CONSOLE GIVE command must be entered at the system console, and cannot be entered from a display station designated as a system service device.

The CONSOLE TAKE command must be entered from a command display; that is, not the inquiry display or a console display.



S9020344-0

TAKE specifies that the system console function is to be taken by an alternative system console. TAKE is specified by the operator of the alternative system console that is to become the system console. If no parameter is specified, TAKE is assumed.

CONSOLE TAKE can be entered for any of the following conditions:

- The current system console is not working.
- The current system console is turned off.
- The current system console is at a remote location that is no longer communicating.
- A CONSOLE GIVE command was entered.
- The ***IPL in progress*** message is displayed on an alternative system console (only the word CONSOLE can be entered).

The current system console operator can also enter CONSOLE TAKE. If no display ID is entered, the effect of the CONSOLE GIVE command is ignored. That is, no operator can take the system console. If a display ID is entered, the operator at the specified display station cannot take the system console. That is, the CONSOLE GIVE command for that display station is ignored.

display id specifies the 2-character ID of a display station that should not be allowed to take the system console.

Entering:

```
CONSOLE  ,display id
```

is not allowed.

GIVE specifies that the system console function is to be made available to an alternative system console. A CONSOLE TAKE command must be entered for the transfer to be complete.

display id specifies the 2-character ID of the alternative system console to become the system console. If display id is not entered, any alternative system console can become the system console. You can use the STATUS WORKSTN control command to determine which display stations are alternative consoles.

The current system console remains the system console until a CONSOLE TAKE command is entered. The current system console must be in command mode for the CONSOLE TAKE command to work.

Example 1

An error occurs at the system console which prevents it from being used. To assign display station W3 (an alternative system console) as the system console, the system operator signs on at display station W3 and enters:

```
CONSOLE
```

Example 2

To switch the system console from W1 to W3, the system console operator enters the following on the W1 (the current system console):

```
CONSOLE GIVE,W3
```

The system console operator then signs on to W3 and enters the following to make W3 the current system console:

```
CONSOLE TAKE
```

Example 3

This example shows a CONSOLE GIVE control command allowing display station W3 to take the system console. Some time later, but before the operator at W3 has entered a CONSOLE TAKE control command, the system operator enters a CONSOLE TAKE control command which causes the system to ignore the earlier CONSOLE GIVE command.

```
CONSOLE GIVE,W3
  .
  .
  .
CONSOLE TAKE,W3
```

Note: If the current system console is a display attached to a Distributed Host Command Facility (DHCF) or a Display Station Pass-through (DSPT) location and communications to that location are interrupted, the following steps may be necessary before the CONSOLE command can be used to transfer the system console function:

- *Use the STATUS SUBSYS (D I) command to determine the name of the SSP-ICF subsystem being used to communicate with the DHCF or DSPT location.*
- *Use the DISABLE procedure to terminate that SSP-ICF subsystem.*
- *After approximately 30 seconds, you should be able to use the CONSOLE command to transfer the system console function.*

HOLD Control Command

You can use the HOLD control command to:

- Prevent one of your entries on the spool file from being printed
- Prevent your jobs on the job queue from being processed

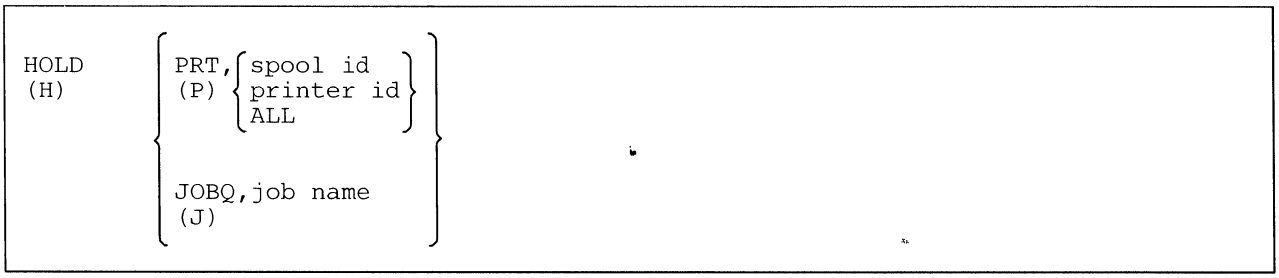
If you control one or more printers, you can also use the HOLD control command to:

- Prevent a spool file entry from being printed for a printer you control
- Prevent all spool file entries from being printed for a printer you control
- Prevent all spool file entries from being printed for all printers you control

Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.

If you are the system operator or a system service display station operator, you can also use the HOLD control command to prevent any job on the job queue from being processed.

If a HOLD PRT command specifies a spool file entry that is being printed, the printing of that entry is interrupted and the printing of the next entry on the spool file begins. The held entry can be printed when a RELEASE control command releases that spool file entry.



S9020345-0

PRT or P specifies that one or more entries on the spool file should be held.

spool id specifies the 6-character ID of the spool file entry to be held. The STATUS PRT command can be used to determine the spool file IDs.

printer id specifies the 2-character ID of a printer. All entries on the spool file for the specified printer are to be held. Only the entries currently on the spool file are held; entries placed on the spool file after the HOLD PRT command are not held. You must control the printer.

ALL specifies that all entries on the spool file for all the printers you control are to be held. Only the entries currently on the spool file are held; any entries that are placed on the spool file after the HOLD command is processed are not held.

JOBQ or J specifies that a job on the job queue is to be held.

job name specifies the 8-character name of a job, on the job queue, to be held. The STATUS JOBQ command can be used to display the names of jobs on the job queue.

Example 1

To hold job W3121536 on the job queue, the system operator enters:

```
HOLD JOBQ,W3121536
```

or:

```
H J,W3121536
```

Example 2

To hold entry SP0012 on the spool file, the operator enters:

```
HOLD PRT,SP0012
```

or:

```
H P,SP0012
```

Example 3

To hold the spool file entries for printer P3, the operator controlling P3 enters:

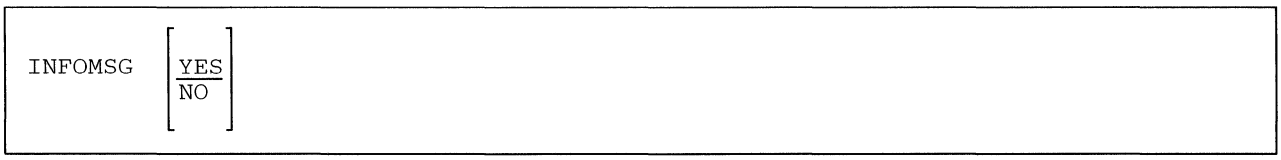
```
H P.P3
```

INFOMSG Control Command

You can use the INFOMSG control command to specify whether you want informational messages to be displayed. Informational messages are displayed by the // * statement.

If you are the system operator, you do not have to respond to informational messages issued by the SSP (if you entered INFOMSG NO in console mode at the system console).

Messages displayed with the // ** statement are always displayed at the system console.



S9020346-0

YES specifies that any informational messages are displayed. Any informational messages displayed on the system console while in console mode must be replied to using the REPLY command. If a parameter is not specified, YES is assumed.

NO specifies that any informational messages issued by the // * statement are not displayed. Also, when entered on the system console or subconsole while in console mode, any informational messages displayed do not have to be replied to.

Example

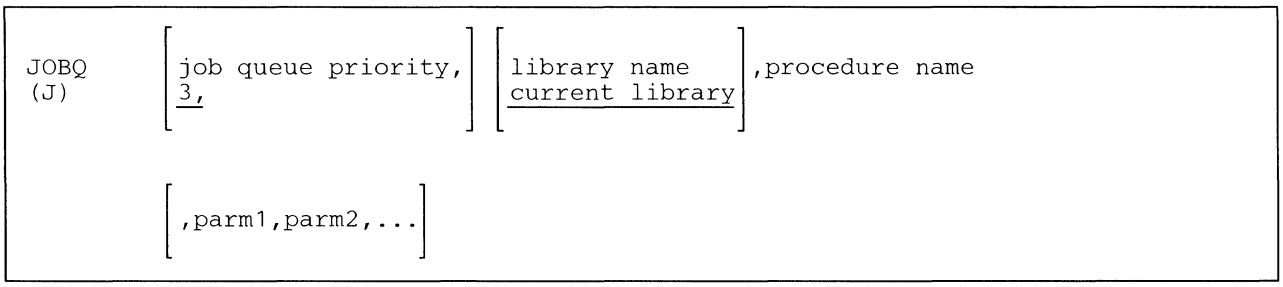
To prevent any informational messages sent to the display station from being displayed, the operator enters:

```
INFOMSG NO
```

JOBQ Control Command

You can use the JOBQ control command to place a job on the job queue.

The JOBQ control command must be entered from the command display. The total number of characters entered at the keyboard for the JOBQ command cannot exceed 120.



S9020347-1

job queue priority specifies job queue priority; that is, the job's order of processing from the job queue. The job queue priority can be any number from 0 through 5. When choosing the next job to run, the system considers jobs with higher priority numbers before jobs with lower priority numbers. For example, all jobs with a job queue priority of 5 are considered before any other jobs in the job queue. Jobs with the same job queue priority are considered in the order they were placed in the job queue. Jobs with a job queue priority of 0 are the last jobs considered by the system. Job queue priority 0 is usually stopped, that is, any jobs placed on the job queue with a priority of 0 will not be considered until the system operator starts priority 0.

If no parameter is specified, a job queue priority of 3 is assigned to the job, and the comma shown in this parameter should not be specified.

library name specifies the library for the job. The system searches the specified library and then the system library for the procedures, load members, message members, and display formats used in this job. If no parameter is specified, the current library is assumed.

procedure name specifies the procedure that defines the job to be placed on the job queue.

parm1,parm2,. . . specifies any parameters required by the procedure. From 1 through 64 parameters can be specified. See the "INCLUDE OCL Statement" on page 5-63 for more information about how parameters can be specified.

Example 1

To place the PAYROLL procedure, which is located in library PAYLIB, on the job queue with a priority of 3, the operator enters:

```
JOBQ PAYLIB,PAYROLL,FILEA,FILEB
```

Example 2

To place the PAYROLL procedure, which is located in library PAYLIB, on the job queue with a job queue priority of 5, the operator enters:

```
JOBQ 5,PAYLIB,PAYROLL,FILEA,FILEB
```

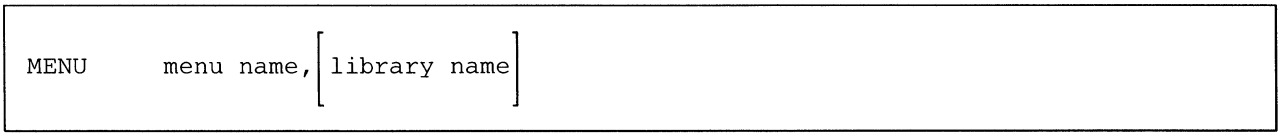
MENU Control Command

You can use the MENU control command to display a user menu. (To display a help menu, see the "HELP Procedure" on page 4-199.) Menus allow you to select procedures or control commands by a number, rather than having to enter that procedure or control command. If you try to display an ideographic menu at a nonideographic display station, an error message is displayed.

For sample user menus and for information about creating user menus, see the manual *Creating Displays*.

You can remove a user menu from your command display by entering a 0 (zero) as a menu option.

The MENU command must be entered from a command display.



S9020348-0

menu name specifies the user menu to be displayed.

library name specifies the library containing the menu. If a library name is not specified, the system searches the current library, if one exists, and then searches the system library (#LIBRARY) for the menu. The library specified becomes the session library.

Example 1

To display menu M12, which is in the current library, the operator enters:

```
MENU M12
```

Example 2

To display menu PAYROL, which is in the library MYLIB, the operator enters:

```
MENU PAYROL,MYLIB
```

MODE Control Command

You can use the MODE control command to put the display station into a mode of operation called **standby mode**, in which it waits to be acquired and used by a program. If the MODE command is entered when the display station is in standby mode, the display station returns to its normal operating mode.

While a display station is in standby mode, only the MODE, MSG, and OFF commands can be entered.

The MODE command must be entered from a command or standby display.

```
MODE
```

S9020349-0

The MODE command has no parameters.

Example

A display station is in normal operating mode. To change the mode of display station operation so that the display station can be acquired by a program, the operator enters:

```
MODE
```

MSG Control Command

You can use the MSG control command to:

- Display messages that were sent to your display station.
- Display a system console message that was also sent to your display station (for your information). Such a message is called a **dual routed message**.
- Send a message to the system console, to another display station, or to another operator.
- Send a message to all display stations.
- Send messages to or receive messages from personal computers that are attached to the System/36 via the IBM Token-Ring Network.
- Send a message to a user on another system.

Note: The message receiver of a personal computer must be active if the personal computer is to receive a message.

Messages received by a display station or operator are placed in the system message file (#MESSAGE). When an initial program load (IPL) is done, messages that were sent to a specific display station are removed from the message file. Messages that were sent to a specific operator and are more than seven days old are also removed from the message file when an IPL is done.

The MSGFILE procedure can be used to:

- Define the size and location of the message file.
- List the display stations and users that have messages in the message file.
- Remove messages from the message file.

For more information about the MSGFILE procedure, see the "MSGFILE Procedure" on page 4-299.

If an operator is signed on to the display station when a message is sent, the audible alarm sounds and the Message Waiting light comes on. Up to 25 messages can be waiting to be displayed for any display station or operator. If an attempt is made to send a message when 25 messages are waiting to be displayed, the messages are not placed in the message file and the sender is notified that the display station or the operator cannot receive messages at this time.

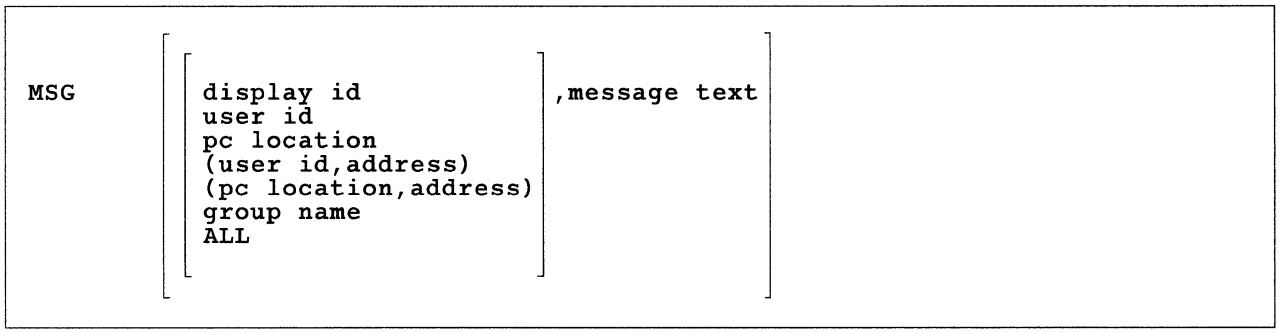
If an operator is signed on to more than one display station when a message is sent to a specific operator, the audible alarm sounds and the Message Waiting light comes on at all of the display stations that the operator is signed on to. The message is displayed at the first display station that requests that the message be displayed. After the message is displayed, the Message Waiting light is reset at all of the display stations that the operator is signed on to.

If an operator is not signed on when a message is sent and an entry in the user identification file exists for that operator, that message is placed in the message file. The message is displayed when the operator signs on. If the operator does not have an entry in the user identification file, the sender is notified that the message cannot be sent to that operator.

Messages sent to a display station are displayed when:

- An operator signs on the display station.
- An operator enters the OFF command (except when OFF DROP is specified on a switched communications line).
- The operator enters the MSG command with no parameters.
- A job being run from the display station ends.
- The operator switches from command mode to console or subconsole mode.

| **Network**, as used in the following discussion, can consist of System/36s and PCs attached to System/36s.



S9020350-3

If no parameters are specified, the messages in the message file for the display station are displayed. If the first parameter is not entered but the second one is, the message is sent to the system console.

display id specifies the 2-character ID of the display station to which the message is to be sent. The STATUS WORKSTN command can be used to determine the display station IDs.

user id specifies the 1- to 8-character user ID that identifies the operator to whom the message is to be sent. The STATUS WORKSTN command can be used to determine the user IDs of operators who are signed on.

| If you specify the user ID of the system operator, the message is sent to the command display of the system
| console. (The message can be viewed from the console display.)

| *Note: It is not recommended to have a user ID that includes special characters such as greater than (>) or less*
| *than (<) signs and parentheses. These characters may be used as control characters by the system.*

| **pc location** specifies the 1- to 8-character personal computer location name in the IBM Token-Ring Network to
| which the message is to be sent. You can use the STATUS SUBSYS control command to determine the
| personal computer locations that are currently active.

(user id,address) specifies the 1- to 8-character user ID and the 1- to 8-character address that together identify a user in the network to whom the message is to be sent. The user ID and address must already exist in the directory.

Notes:

1. *This parameter is valid only if DSNX-ND is installed.*

2. *It is not recommended to have a user ID that includes special characters such as greater than (>) or less than (<) signs and parentheses. These characters may be used as control characters by the system.*

(pc location,address) specifies the 1- to 8-character personal computer location name in the IBM Token-Ring Network and the 1- to 8-character address that together identify the personal computer location name in the network to which the message is to be sent. The PC location name and address must already exist in the directory.

Note: This parameter is valid only if DSNX-ND is installed.

group name specifies the name of the distribution list given in a System/36 directory that identifies a group of users in the network to which the message is sent.

Note: This parameter is valid only if DSNX-ND is installed.

ALL specifies that the message is to be sent to all active display stations, except the display station itself.

message text specifies the message to be sent. Up to 75 alphameric or special characters can be entered. The message text can contain IGC characters; however, the message will not be sent to a nonideographic display station. If you try to send an ideographic message to a nonideographic display station, an error message is displayed.

Example 1

A display station operator is preparing to submit a job that requires diskette BFILE. To inform the system operator that diskette BFILE is required, the display station operator enters:

```
MSG ,PLEASE PUT DISKETTE BFILE INTO SLOT 1
```

Example 2

To notify the operator at display station W4 that no more jobs can be entered, the system operator enters:

```
MSG W4,PLEASE DO NOT SUBMIT ANY MORE JOBS
```

Example 3

To send a message to all operators on the system:

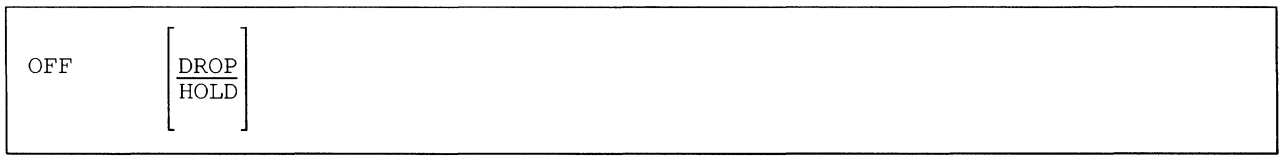
```
MSG ALL,PLEASE END YOUR JOBS, I NEED A DEDICATED SYSTEM
```

OFF Control Command

You can use the OFF control command to sign your display station off the system. The OFF command is not processed unless all jobs using the display station have ended.

Messages in the message file for the display station are displayed at sign off unless you specify OFF DROP at a remote display station on a switched communications line. If the messages are not displayed, they remain in the message file and can be displayed later.

The OFF command must be entered from a command display.



S9020351-0

DROP and HOLD are used only for remote display stations on switched or X.25 communications lines and display stations attached through Display Station Pass-Through. These parameters are ignored for other display stations.

DROP specifies that the communications session for this display is no longer needed.

If the display is communicating using display station pass-through, the pass-through session will terminate and the display will only communicate with its local system.

If the display is communicating using remote work station support (RWS), the communications line will be dropped if all work stations on that line have completed their activity after all work stations have entered OFF or OFF DROP. For X.25, the difference is that the virtual circuit to the remote work station controller will be disconnected instead of dropping the communications line. This support for X.25 must be configured using CNFIGSSP. If the display is the system console, an alternate console may use the CONSOLE command to transfer the system console function once the communications line has dropped.

HOLD specifies that the communications session for the display is to be held. A SIGN ON display will appear. If the display is communicating using display station pass-through, the SIGN ON display will be for the remote system.

Example

To sign your display station off the system, enter:

```
OFF
```

POWER Control Command

If you are the system operator, you can use the POWER control command at any display station to power off the system.

If any jobs are running, a message is displayed at the system console and you can either cause the POWER command to be ignored (the jobs can continue running), retried (to see if jobs are still running), or proceed with powering off the system.

```
POWER     OFF
```

S9020352-0

OFF specifies that the system is to be powered off.

Example

To power off the system, the system operator would enter:

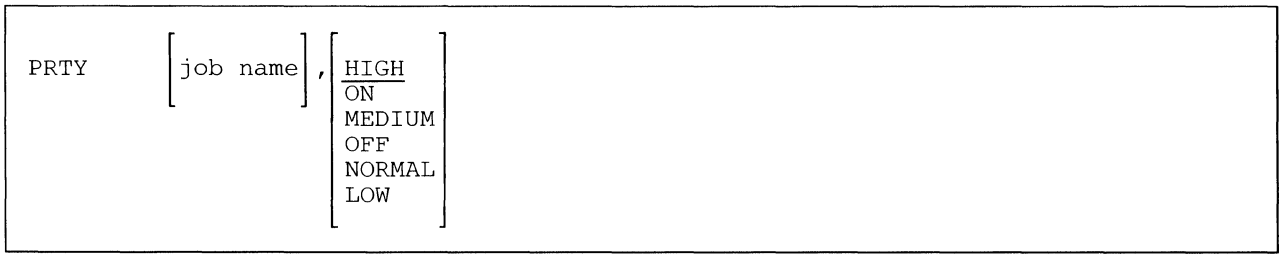
```
POWER OFF
```

PRTY Control Command

You can use the PRTY control command to change the processing priority of the next job run from the display station or of the next job placed on the job queue.

If you are the system operator or a system service device operator, you can also use the PRTY command to change the processing priority of a currently running job or of a job on the job queue.

NORMAL priority jobs can have their processing priority automatically changed by the system. HIGH, MEDIUM, and LOW priority jobs do not automatically change priority. See the manual *Learning about Your Computer* for more information about job priority.



S9020353-0

job name specifies the 8-character name of the job whose processing priority is to be changed. The STATUS USERS or STATUS JOBQ command can be used to determine the job name.

Note: This parameter can be entered **only** *at the system console or a system service display station.*

HIGH or ON specifies that system resources are assigned to the job before they are assigned to a lower priority (MEDIUM, NORMAL, OFF, or LOW) job. If a second parameter is not specified, HIGH is assumed.

MEDIUM specifies that system resources are assigned to the job before they are assigned to a lower priority (NORMAL, OFF, or LOW) job.

OFF or NORMAL specifies that system resources are assigned to the job before they are assigned to a LOW priority job.

LOW specifies that system resources are assigned to the job after all other higher priority jobs have been assigned system resources.

Example 1

To assign high priority to job W2010112, the system operator enters:

```
PRTY W2010112
```

Example 2

To assign high priority to the procedure PROCA, an operator enters:

```
PRTY
PROCA
```

RELEASE Control Command

You can use the RELEASE control command to:

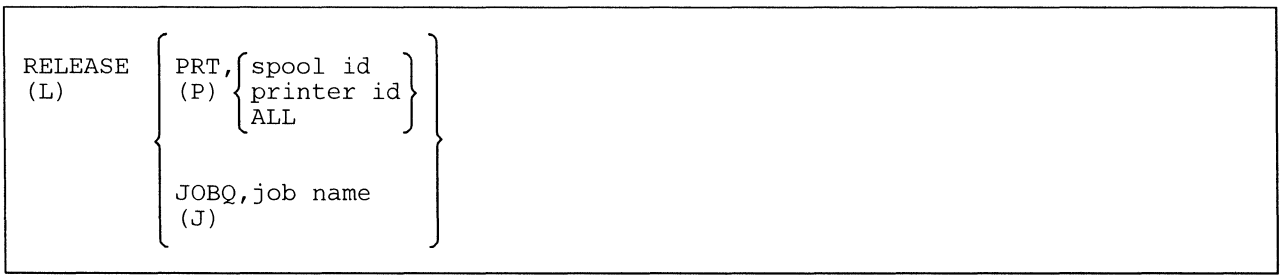
- Release one of your entries on the spool file for printing
- Release your jobs on the job queue for processing

If you control one or more printers, you can also use the RELEASE control command to:

- Release one or all entries on the spool file for a specific printer that you control
- Release all held entries on the spool file for all printers you control

Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.

If you are the system operator, or a system service display station operator, you can also use the RELEASE control command to release any one or all held entries on the spool file.



S9020354-0

PRT or P specifies that one or more spool file entries are to be released.

spool id specifies the 6-character ID of the spool file entry to be released. The STATUS PRT command can be used to determine the spool file IDs.

Note: Some spool file entries are automatically held by the system during an IPL. The entries are held because the IPL has determined that the entries were not closed before the IPL. This situation most often occurs when there is an unscheduled IPL, such as a power failure or a system problem, that requires an IPL to recover.

These entries may be released with the RELEASE control command; however, because the entries have not been properly closed, the system cannot determine the end of these entries, and problems can occur when the released entries are printed.

For example, in some cases, programming errors will be reported when the output is printed. In other cases, the output may be followed by unrelated information in the spool file. If your spool file contains or has contained sensitive information, you should use appropriate caution when you print incomplete spool file entries.

printer id specifies the 2-character ID of a printer. All the spool file entries to be printed on the specified printer are to be released. Only the entries currently held on the spool file are released; any held entries that are placed on the spool file after the RELEASE command is processed are not released. You must control the printer.

ALL specifies that all spool file entries for all printers you control are to be released. Only the entries currently held on the spool file are released; any held entries that are placed on the spool file after the RELEASE command is processed are not released.

JOBQ or J specifies that a job on the job queue is to be released.

job name specifies the 8-character name of a job, on the job queue, to be released. The STATUS JOBQ command can be used to display the names of jobs on the job queue.

Example 1

To release job W3121536 on the job queue, an operator enters:

```
RELEASE JOBQ,W3121536
```

or:

```
L J,W3121536
```

Example 2

To release entry SP0013 on the spool file, an operator enters:

```
RELEASE PRT,SP0013
```

Example 3

To release all the entries on the spool file, the system operator enters:

```
L P,ALL
```

Example 4

To release the spool file entries for printer P3, the operator controlling P3 enters:

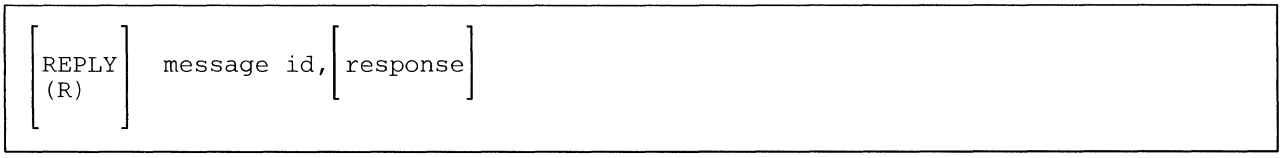
```
L P,P3
```

REPLY Control Command

You can use the REPLY control command to:

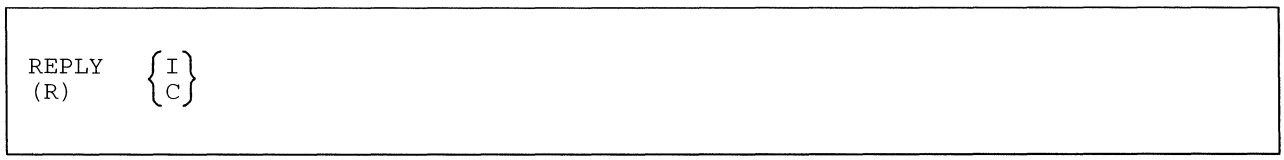
- Respond to an individual message on the Console display
- Respond to all informational messages on the Console display
- Compress the console display so that only the messages that still need a response are displayed
- Respond to an individual message at a subconsole that is displayed with the STATUS MESSAGE control command

To reply to a message or to display help information about a message:



S9020355-0

To reply to all informational messages or to clear the display:



S9020356-0

message id specifies the 2-character message ID that identifies the message being responded to.

response specifies the response to the message. For example, if an SSP message is being responded to, the response might be 0, 1, 2, 3, or D.

I specifies that all informational messages are to be responded to. REPLY or R is required when I is specified.

C clears the display of all messages that have been responded to. REPLY or R is required when C is specified.

Example 1

The following message appears on the Console or subconsole display:

```
02 SYS-1405 Options (012 )
   Do you want spool separator pages on printer P1...
```

To select option 0, the system operator enters:

```
REPLY 02,0
```

or:

```
R 2,0
```

or:

```
2,0
```

To get additional text for the message, the operator enters:

```
R 2
```

or:

```
2
```

Example 2

To clear the system console or subconsole display of all informational messages and messages that were responded to, the operator enters:

```
R I
R C
```

The R I command responds to all informational messages, and the R C command clears the display, leaving only those messages not yet responded to (if any).

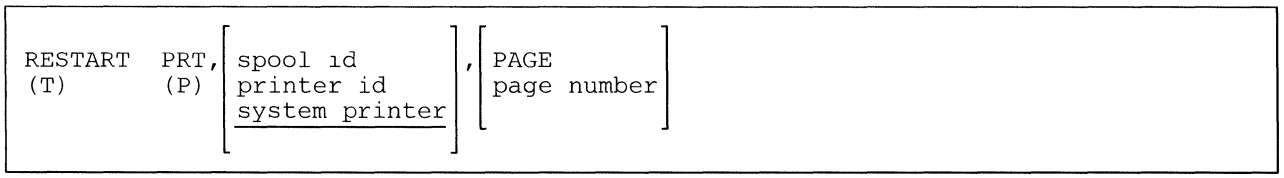
RESTART Control Command

If you control one or more printers, you can use the RESTART control command to restart the printing of an entry from the spool file for a printer you control. You can restart the printing at the beginning of the specified entry, from the top of a specified page of that entry, or at the top of the last page that was partially printed by the spool writer.

Notes:

1. *If you restart printing a spool file entry at a place other than its beginning, the system assumes that all of the data in the spool file entry has the same print file characteristics, for example, the same number of lines per page. If the spool file entry consists of data with varying print file characteristics and you restart printing the spool file entry at a place other than its beginning, programming errors may occur, or the printed output may be different than if the entire spool file entry had been printed.*

2. *If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.*



S9020357-0

PRT or P specifies that the printing of a spool file entry is to be restarted.

spool id specifies the 6-character ID of the spool file entry to be restarted. The STATUS PRT command can be used to determine the spool file IDs.

printer id specifies the 2-character ID of the printer to be restarted. If no printer ID is specified, the system printer is assumed. You can use the STATUS WRT command to display the printer IDs.

PAGE specifies that the printing is to be restarted at the top of the last page that was partially printed by the spool file writer. If the output was not partially printed, the printing begins at the top of the first page.

page number specifies the number of the page where printing is to restart. If the page number is not specified, printing restarts at the beginning of the printed output. The maximum page number is 65535. If the spool file entry exceeds 65535 pages, this parameter should not be used.

Example 1

To restart printing of the current spool file entry at the top of page 6 on the system printer, the operator controlling the system printer enters:

```
RESTART PRT,,6
```

or:

```
T P,,6
```

Example 2

To restart printing of spool file entry SP0012 at the top of the last page that was printed:

```
RESTART PRT,SP0012,PAGE
```

or:

```
T P,SP0012,PAGE
```

START Control Command

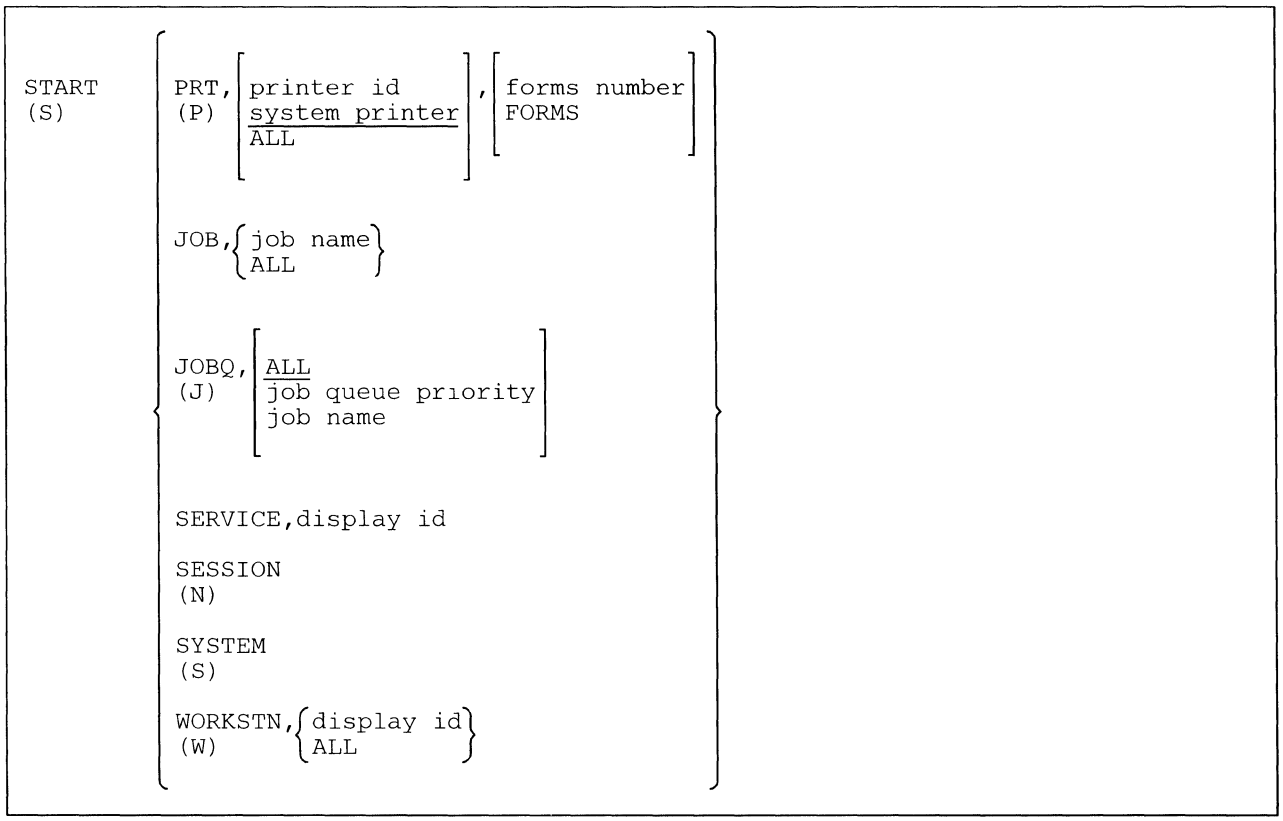
If you control one or more printers, you can use the START control command to:

- Start the printing of all spool file entries for a specific printer or for all printers you control
- Start the printing of all spool file entries with a specified forms number for a specific printer or for all printers you control
- Start the printing of all spool file entries for a specific printer or for all printers you control such that those entries using the same forms are printed together

Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.

If you are the system operator or a system service device operator, you can use the START control command to:

- Resume the running of a job, or all jobs, that were stopped by a STOP JOB command.
- Start running jobs from the job queue.
- Allow a display station to be used as a system service display station. (This can only be done from the system console.)
- Resume the SSP-ICF activity that was stopped by a STOP SESSION command.
- Resume the system activity that was stopped by a STOP SYSTEM command. (This can only be done from the system console.)
- Allow jobs to be started from all display stations or from a specified display station that was stopped by a STOP WORKSTN command.



S9020358-0

PRT or P specifies that a spool writer for a printer is to be started; that is, spool file entries can be printed when the printer is available. The PRT parameter can be used to start the printing of entries on the spool file after IPL if the automatic start function (also called the autowriter) was not selected during system configuration. The PRT parameter can also be used to start the spool writer(s) after the STOP PRT command is entered.

If no second parameter is entered, the system printer is assumed. If no third parameter is entered, the spool writer prints the available entries according to their position in the spool file.

printer id specifies the 2-character ID of the printer for which the spool writer is to be started. If no parameter is specified, the system printer is assumed. You can use the STATUS WRT control command to determine whether the spool writer for a particular printer is started or stopped.

ALL specifies that the spool writers for all printers you control are to be started.

forms number specifies the forms number to be used by the spool writers; from 1 through 4 characters can be entered. Only entries with the specified forms numbers are printed.

FORMS specifies that the spool writer is to print all available spool file entries that require the forms currently being used in the printer, regardless of the entry's position in the spool file. After all such entries are printed, the operator controlling the printer is prompted to change the forms, and all entries using the next group of forms are printed. This can reduce the number of operator messages to change the forms.

JOB specifies that a job, or all jobs, that were stopped by a STOP JOB command are to be started. **This can be entered only** from the system console or a system service display station.

job name specifies the 8-character job name of the job to be started. The STATUS USERS command can be used to determine the job names.

ALL specifies that all jobs that were stopped by a STOP JOB command are to resume processing.

JOBQ or J specifies that all jobs, or those jobs with the specified job queue priority, on the job queue can be processed. When the job queue becomes empty, running of jobs begins automatically when one or more jobs are placed on the job queue. **This can be entered only** from the system console or a system service display station.

ALL specifies that all jobs on the job queue can be run. If no parameter is specified, ALL is assumed.

job queue priority specifies that only those jobs on the job queue with the specified job queue priority can be started. You can enter any number from 0 through 5.

job name specifies that a specific job on the job queue should be run immediately. It will not affect the processing of the other jobs currently running from the job queue.

SERVICE specifies that the display station specified by **display id** is to become a system service display station.

Note: This parameter can be entered only at the system console.

Also, if password security is active, SERVICE can only be specified by an operator that has service aid authorization.

You must be able to enter commands from the display station specified, that is, the display station must not be a data display station.

Procedures that can be run at any display station and by any operator can also be run at a system service display station. Procedures that can be run by a system operator at any display station can be run at a system service display station. Procedures that can only be run at a system console cannot be run at a system service display station. Some procedures can be run at any display station by any operator when password security is not active. These procedures can always be run at a system service display station whether or not password security is active.

You can run all the control commands at a system service display station except:

ASSIGN
CANCEL SESSION for the system console
CONSOLE
START SERVICE
START SYSTEM
STOP SERVICE
STOP SYSTEM

If the system service display station is also an alternative system console, the CONSOLE command can be used. If the system service display station is a remote work station, you will need to establish a communications link before you can use the system service display station. See the manual *Operating Your System* for your system unit for more information about establishing a communications link.

SESSION or N resumes the SSP-ICF activity that was stopped by a STOP SESSION command.

Note: This parameter can be entered only at the system console or a system service display station.

SYSTEM or S resumes the system activity that was stopped by a STOP SYSTEM control command.

Notes:

1. *If the system is in the process of sorting indexed files because of a stop system sort request and a start system command is entered, the system will finish sorting the current file and will not sort any more files.*

2. *This parameter can be entered only at the system console.*

WORKSTN or W specifies that jobs can be started and commands can be entered from all display stations (if ALL is entered in the second parameter position), or from a specified display station for which a STOP WORKSTN control command was entered.

Note: This parameter can be entered only at the system console or a system service display station.

display id specifies the 2-character ID of the display station to start. You can use the STATUS WORKSTN command to determine the display IDs.

ALL allows the initiation of jobs and the entry of commands stopped by a STOP WORKSTN control command.

Example 1

The initiation of jobs from the job queue was stopped by a STOP JOBQ control command. To start running jobs from the job queue, the system operator enters:

```
START JOBQ
```

Example 2

Initiation of all jobs, including jobs from the job queue and spool file, was stopped by a STOP SYSTEM control command. To allow the initiation of jobs on the system, the system operator enters:

```
S S
```

Example 3

To start printer P3, the operator controlling P3 enters:

```
S P,P3
```

STATUS Control Command

You can use the STATUS control command to display the status of:

- Yours or another display station session.
- Display station communications line parameters.
- System communications line parameters.
- Activity for a communications line.
- Jobs you have placed on the job queue.
- Entries on the spool file. You are shown all the information about your entries or about entries for printers that you control. For other entries shown, the spool ID, procedure name, job name, user ID, and printer file names are not displayed.

  Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if system security is active and you have system operator authority or higher.

- Active SSP-ICF sessions.
- Enabled SSP-ICF subsystems.
- Enabled APPC subsystems.
- MSRJE (multiple session remote job entry).
- Display stations, printers, Display Station Pass-Through (DSPT) and Distributed Host Command Facility (DHCF) devices, the diskette drive, and tape drives.
- Spool writers.
- Active tasks in the system and their task block addresses.
- A selected job or all jobs running on the system.
- A system task.
- Messages that are not replied to for all of the subconsoles (display stations that control a printer) or for a specific subconsole.

If you control one or more printers, you can also use the STATUS control command to display the status of entries on the spool file for a specified printer. You are shown all the information about the entries.

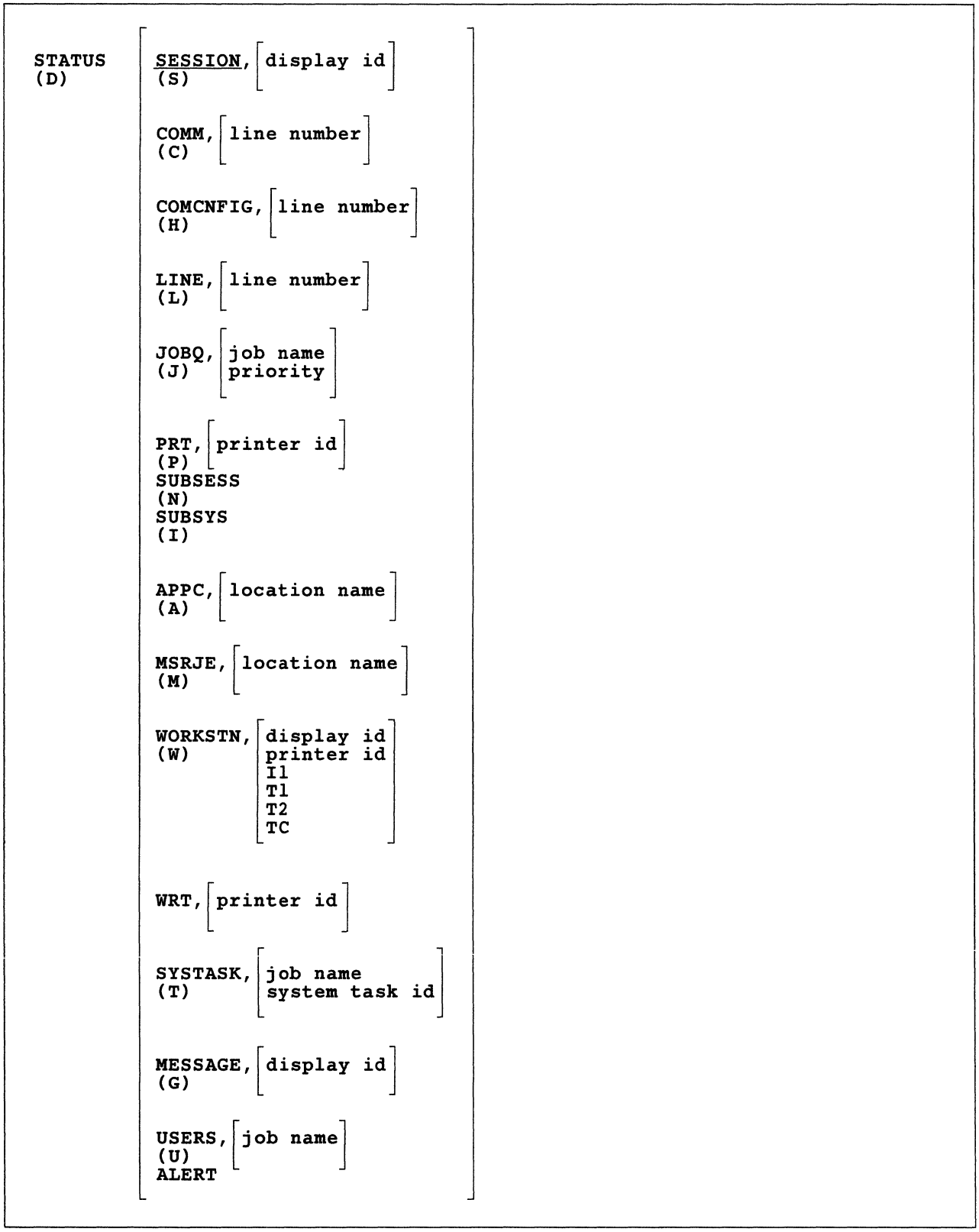
If you are the system operator or a system service display station operator, you can also use the STATUS control command to display the status of:

- All jobs on the job queue.
- All entries on the spool file. All information is displayed.

Each display for the STATUS control command contains two parts: the status part and the menu part. For a detailed description of the information appearing on the display for the STATUS control command, press command key 8. Each time status information is displayed for the STATUS control command, the operator can:

- End the status display
- Display the next page of status information (if the last page is being displayed, the status display resets to the first page of status information)
- Display the previous page
- Enter a command or menu option
- Reset the status display to the first page of status information
- Request an updated status display

To return to the menu part of the display, press the Help key.


STATUS
(D)
SESSION,
(S)
display id
COMM,
(C)
line number
COMCNFIG,
(H)
line number
LINE,
(L)
line number
JOBQ,
(J)
job name
priority
PRT,
(P)
printer id
SUBSESS
(N)
SUBSYS
(I)
APPC,
(A)
location name
MSRJE,
(M)
location name
WORKSTN,
(W)
display id
printer id
I1
T1
T2
TC
WRT,
printer id
SYSTASK,
(T)
job name
system task id
MESSAGE,
(G)
display id
USERS,
(U)
job name
ALERT

S9020359-4

SESSION or S displays information about the display station session. If no first parameter is specified, SESSION is assumed. If no display ID is specified, the status of your display station is shown.

display id specifies the 2-character ID of the display station for which status information is to be displayed.

COMM or C displays the status of the display stations communications line parameters.

COMCNFIG or H displays the status of the system communications line parameters.

LINE or L displays the activity for a communications line.

line number specifies the communications line for which status information is to be displayed. If no parameter is specified, the status of line one is displayed.

JOBQ or J displays entries on the job queue. If no job name is specified, all jobs on the queue are displayed.

job name specifies the 8-character name of a job, on the job queue, for which status information is displayed.

priority specifies that you want to display all jobs on the job queue with the specified priority. Priorities are 0 through 5.

PRT or P displays information about spool file entries. If a printer ID is specified, only the spool entries for that printer are displayed. If no printer ID is specified, the spool entries for all the printers are displayed.

printer id specifies the 2-character ID of the printer for which spool file information is to be displayed.

SUBSESS or N displays information about active SSP-ICF sessions.

SUBSYS or I displays information about enabled SSP-ICF subsystems.

APPC or A displays the status of one or all currently-enabled APPC or APPN remote locations. If no remote location name is specified, status information about all APPC remote locations is displayed.

location name specifies the 8-character name of a single APPC remote location for which status information is to be displayed.

MSRJE or M displays the status of one or all currently enabled MSRJE locations. If no location name is specified, status information about all MSRJE locations is displayed.

location name specifies the 8-character name of a single MSRJE location for which status information is to be displayed.

WORKSTN or W displays status information about:

- The local and remote display stations and printers
- The diskette drive
- The tape drives
- DHCF and DSPT devices
- Printer pass-through devices

If no second parameter is specified, the status of all devices is shown.

display id specifies the 2-character ID of the display station for which status information is to be displayed.

printer id specifies the 2-character ID of the printer for which status information is to be displayed.

I1 specifies status information for the diskette drive is to be displayed.

T1 specifies that status information for tape drive 1 is to be displayed.

T2 specifies that status information for tape drive 2 is to be displayed.

TC specifies that status information for the tape cartridge drive is to be displayed.

WRT displays the status of the spool file writers. If no printer ID is specified, the status of all the spool file writers is displayed.

printer id specifies the 2-character ID of the printer whose spool file writer status is to be displayed.

SYSTASK or T displays the active tasks in the system and their task block addresses. If no second parameter is specified, the status of all active tasks is displayed.

job name specifies the 8-character name of the job for which status information is displayed.

system task id specifies the 4-character ID that identifies the system task. For example, the task ID of the command processor is 0009.

MESSAGE or G displays the messages that are not replied to for one or more subconsoles. If no display ID is specified, the status of all active subconsole messages is displayed.

display id specifies the 2-character ID of the subconsole display station for which message information is to be displayed.

USERS or U displays the status of a selected job or all jobs running on the system. If no job name is specified, the status of all jobs is displayed.

job name specifies the 8-character name of the job for which status information is displayed.

ALERT displays the status of active and enabled alert locations.

Example

To display the status of all jobs on the spool file:

```
STATUS PRT
```

or:

```
D P
```

STATUSF Control Command

You can use the STATUSF control command to display the status of:

- Jobs you have placed on the job queue.
- Entries on the spool file. You are shown all the information about your entries or about entries for printers you control. For other entries shown, the spool ID, procedure name, work station ID, and user ID are not displayed.

  Note: If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.

- Display stations, printers, Display Station Pass-Through (DSPT) and Distributed Host Command Facility (DHCF) devices, the diskette drive, and tape drives.
- A selected job or all jobs running on the system.

If you control one or more printers, you can also use the STATUSF control command to display the status of entries on the spool file for a specified printer. You are shown all the information about the entries.

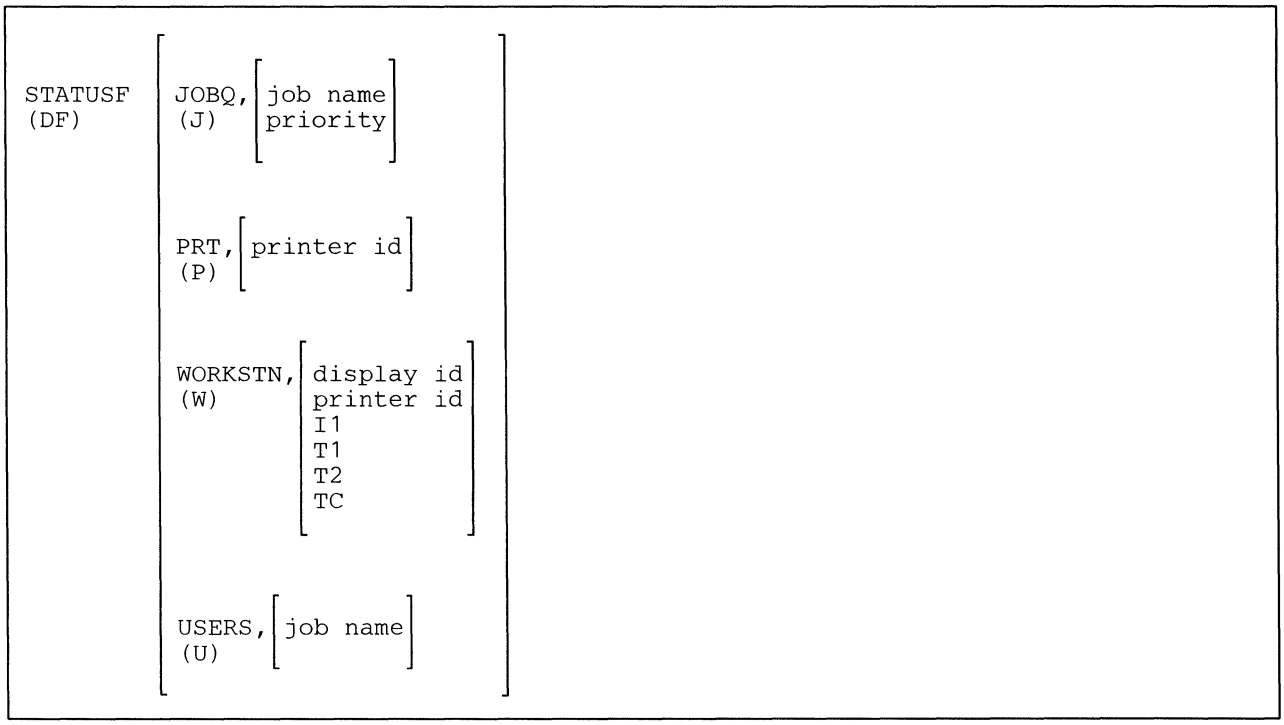
If you are the system operator or a system service display station operator, you can also use the STATUSF control command to display the status of:

- All jobs on the job queue.
- All entries on the spool file. All information is displayed.

Each display for the STATUSF control command contains a full display of status information. For a detailed description of the information appearing on the display for the STATUSF control command, press the Help key. Each time status information is displayed for the STATUSF control command, the operator can:

- End the status display.
- Display the next page of status information (if the last page is being displayed, the status display resets to the first page of status information).
- Display the previous page.
- Enter a command.
- Reset the status display to the first page of status information.
- Request an updated status display.

STATUSF



S9020360-2

JOBQ or J displays entries on the job queue. If no job name is specified, all jobs on the job queue are displayed.

job name specifies the 8-character name of a job, on the job queue, for which status information is displayed.

priority specifies that you want to display all jobs on the job queue with the specified priority. Priorities are 0 through 5.

PRT or P displays information about spool file entries. If a printer ID is specified, only the spool entries for that printer are displayed. If no printer ID is specified, the spool entries for all the printers are displayed.

printer id specifies the 2-character ID of the printer for which spool file information is to be displayed.

WORKSTN or W displays status information about:

- The local and remote display stations and printers
- The diskette drive
- The tape drives
- DHCF and DSPT devices

If no second parameter is specified, the status of all devices is shown.

display id specifies the 2-character ID of the display station for which status information is to be displayed.

printer id specifies the 2-character ID of the printer for which status information is to be displayed.

I1 specifies that status information for the diskette drive is to be displayed.

T1 specifies that status information for tape drive 1 is to be displayed.

T2 specifies that status information for tape drive 2 is to be displayed.

TC specifies that status information for the tape cartridge drive is to be displayed.

USERS or U displays the status of a selected job or all jobs running on the system. If no job name is specified, the status of all jobs is displayed.

job name specifies the 8-character name of the job for which status information is displayed.

Example

To display the status of all jobs on the spool file:

```
STATUSF PRT
```

or:

```
DF P
```

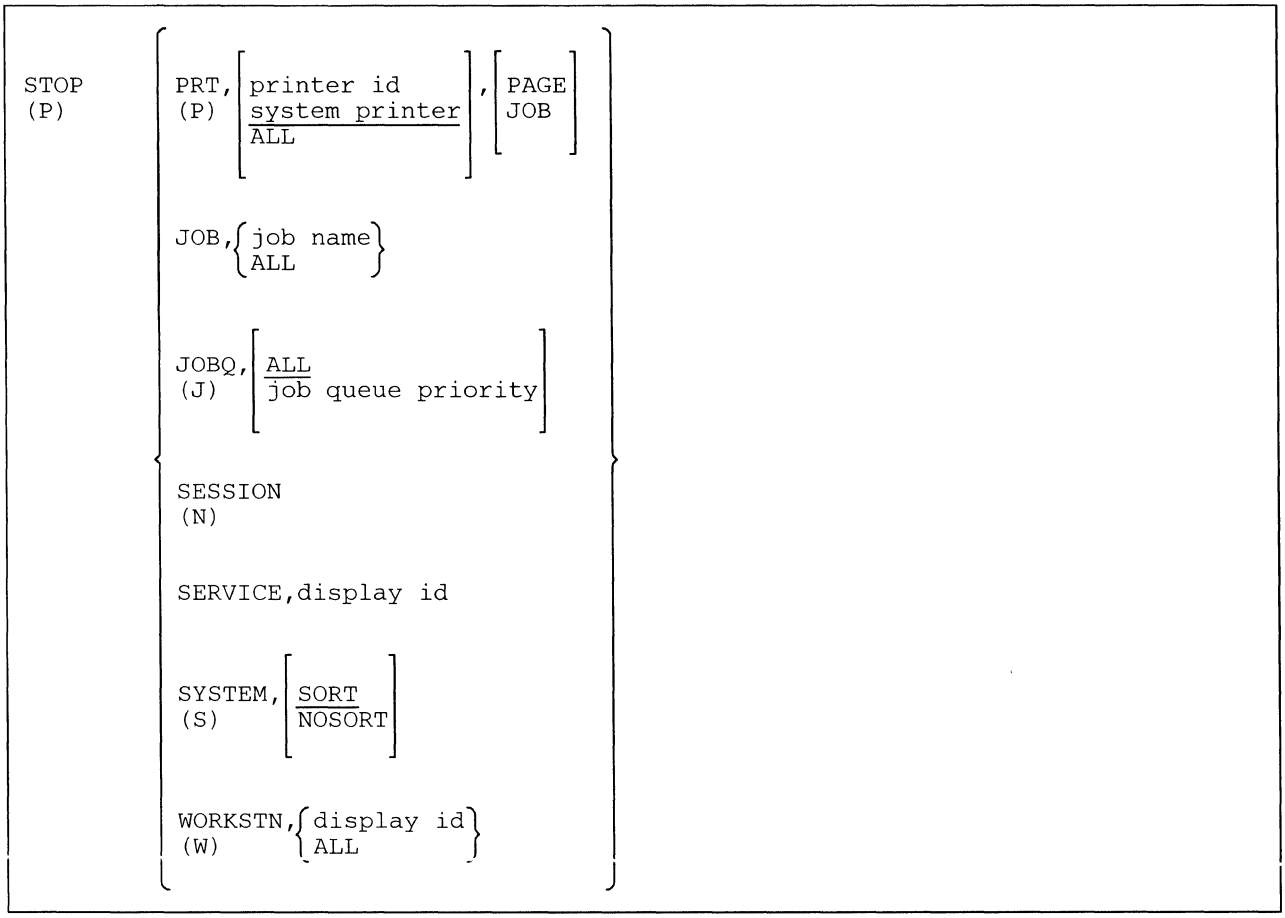
STOP Control Command

If you control one or more printers, you can use the STOP control command to stop the printing of all entries from the spool file for a specified printer or for all printers that you control. Printing can be stopped immediately, at the end of the currently printing page, or at the end of the currently printing spool file entry.

| *Note:* *If you are a subconsole operator, you control the printers for which the subconsole display station is designated as the subconsole. If you are a system console operator or a system service display station operator, you control all printers. You also control all printers if password security is active and you have system operator authority or higher.*

If you are the system operator or a system service display station operator, you can also use the STOP control command to:

- Stop the processing of a specified job or all jobs.
- Stop the running of jobs from the job queue.
- Stop a display station from being used as a system service display station. (This can only be done from the system console.)
- Prevent the starting of jobs from incoming SSP-ICF sessions
- Begin an orderly shutdown of the system with or without the sorting of index keys. (This can only be done from the system console.)
- Prevent the starting of jobs from all display stations other than the system console, or prevent the starting of jobs from a specified display station.



S9020361-0

PRT or P stops the printing of entries on the spool file. If an entry is being printed and a third parameter is not specified, printing stops immediately. You can resume the printing by entering the START PRT command or the RESTART command.

printer id specifies the 2-character ID of the printer for which the spool writer is to be stopped. If no printer ID is specified, the system printer is assumed.

ALL specifies that printing is to stop for all printers you control.

PAGE stops the spool writer when it has completed printing the current page.

JOB stops the spool writer when it has completed printing the current copy of the spool file entry.

JOB stops the running of all jobs or stops the running of a specified job. **This parameter can be entered only** from the system console or a system service display station. The STOP JOB command can be used to:

- Stop a job that appears to be holding control of the system
- Stop all jobs in order to run an important job or a job that can be run only when no other jobs are running

job name specifies the 8-character name of the job to be stopped. The STATUS USERS command can be used to determine the job names.

ALL specifies that all running jobs are to be stopped.

JOBQ or J prevents the starting of all jobs or jobs with the specified job queue priority from the job queue. The job that is already running continues until completed.

Note: This parameter can be entered only from the system console or a system service display station.

ALL specifies that all jobs on the job queue are to be prevented from starting.

job queue priority specifies a job queue priority; a number from 0 through 5. All jobs with the specified job queue priority are to be prevented from starting.

SESSION or N prevents the starting of jobs from incoming SSP-ICF sessions.

Note: This parameter can be entered only from the system console or a system service display station.

SERVICE stops a display station from being used as the system service display station.

Note: This parameter can be entered only from the system console.

display id specifies the 2-character ID of a display station that is to be stopped from being used as a system service display station.

If the system service display station is a remote work station, you may need to end the communications link to that remote display station. See the manual *Operating Your System* for your system unit for more information about ending a communications link.

SYSTEM or S causes an orderly shutdown of system activities to begin. **This parameter can be entered only** from the system console.

SORT specifies that index keys are to be sorted as part of the system shutdown. Only index files with a large overflow area for the keys will be sorted by the system. If you want to force a specific index file to be sorted, refer to the "KEYSORT Procedure" on page 4-252. If SYSTEM is specified as the first parameter and a second parameter is not specified, SORT is assumed.

NOSORT specifies that index keys are not to be sorted.

STOP SYSTEM causes the following to occur:

For each display station input operation in a program, a condition code that indicates that a STOP SYSTEM control command was entered is returned to the program along with the data. Programs should check for this condition, close all files, and go to end of job as soon as possible.

After the system is stopped:

- Display station operators cannot start jobs or enter the JOBQ command.
- At the system console, the system operator can start new jobs and can enter all commands except for STOP WORKSTN.

WORKSTN or W prevents the starting of jobs and the processing of certain control commands from all display stations except the system console. All jobs currently running continue to run. If a display station ID is entered in the second parameter position, only the starting of jobs from the specified display station is prevented.

Note: This parameter can be entered only from the system console or a system service display station.

display id specifies the 2-character ID of the display station from which no new jobs can be started.

ALL specifies that no new jobs can be started. Also, the JOBQ control command is not allowed.

Example 1

The system operator is preparing to shut down the system. To stop the initiation of jobs from the display stations, the job queue, and the spool file, sort the index keys the system operator enters:

```
STOP SYSTEM
```

Example 2

To stop the printing of printer P3, the operator who controls printer P3 would enter:

```
STOP PRT,P3
```

or:

```
P P,P3
```

TIME Control Command

You can use the TIME control command to display the time of day and the system date. The time is based upon the time specified by the system operator during IPL and is displayed in the following format:

```
hh:mm:ss
```

hh specifies the hours.

mm specifies the minutes.

ss specifies the seconds.

The system date is displayed in the system date format:

```
mmddyy, ddmmyy, or yymmdd
```

mm specifies the month.

dd specifies the day.

yy specifies the year.

When the time advances from 23:59:59 to 00:00:00, the system date advances.

```
TIME
```

S9020362-0

The TIME command has no parameters.

Example

To display the time of day and the system date, an operator enters:

```
TIME
```

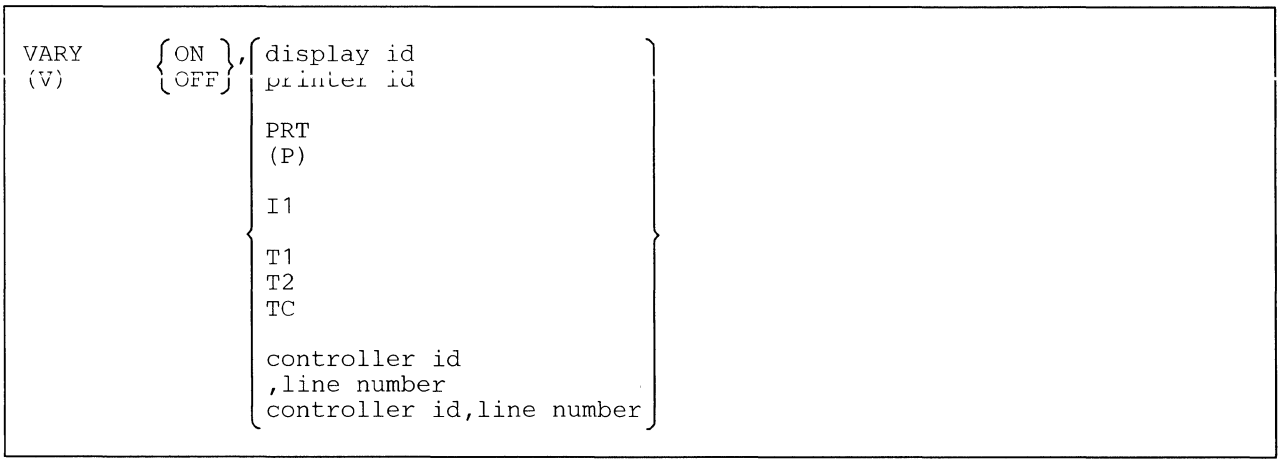
VARY Control Command

If you have system operator authority or higher, you can enter the VARY control command at any type of display station to change the status of the following from online to offline and from offline to online.

- Display stations
- Printers
- The system printer
- The diskette drive
- Communications controllers
- Communications lines
- Tape drives

Devices that are offline cannot be used by operators or programs. The VARY command cannot be used to take offline a device that is allocated to a program or signed on.

You can use the STATUS WORKSTN command to determine the current status of devices on the system.



S9020363-2

ON causes the specified device to be placed online.

OFF causes the specified device to be placed offline.

display id specifies the 2-character ID of a display station to be placed online or offline. You can use the STATUS WORKSTN command to determine the display station IDs.

A display station must be signed off to be placed offline. If the device being varied offline is a subconsole, any messages that were sent to the subconsole are sent to the system console.

printer id specifies the 2-character ID of a printer to be placed online or offline. You can use the STATUS WORKSTN command to determine the printer IDs.

The spool writers for a printer must be stopped or complete in order to place that printer offline. You can use the STATUS WRT command to determine whether a spool writer is stopped or complete.

PRT or P specifies that the system printer is to be placed online or offline.

I1 specifies that the diskette drive is to be placed online or offline.

controller id specifies the 3-character ID of a communications controller. All remote display stations and printers associated with the specified controller are placed online or offline.

line number specifies the communications line number. When this parameter is entered without the controller ID parameter, all controllers, display stations, and printers associated with this line are placed online or offline. When this parameter is entered with the controller ID parameter on the VARY ON command, the specified device is placed online on the specified line.

controller id,line number is valid only for switched communications lines. When these parameters are entered on the VARY ON command, the specified device is placed online on the specified line.

T1 specifies that tape drive 1 is to be placed online or offline.

T2 specifies that tape drive 2 is to be placed online or offline.

TC specifies that the tape cartridge drive is to be placed online or offline.

Example

To place all remote display stations and printers associated with controller C01 online, the system operator enters:

```
VARY ON,C01
```

Appendix A. SSP Utility Programs

This appendix shows the SSP utility programs you can use instead of the SSP procedures. The utility programs are in alphabetical order by utility, with a description of the functions that can be performed using that utility, and the OCL and utility control statements that can be used to perform that function.

Making Your Own Procedures Using SSP Utility Programs

This section describes why you may want to make your own procedures from the SSP utility programs. By making your own procedures based on the statements shown, you can specify OCL statements to change how the procedures run. For example:

- You can specify different job or job step dates, using the DATE OCL statement. See the "DATE OCL Statement" on page 5-25 for more information about changing the date.
- You can specify different disk file retention types, using the FILE OCL statement. See the "FILE OCL Statement (for Disk Files)" on page 5-32 for more information about disk file retention types.
- You can specify special printer directions, using the PRINTER OCL statement. See the "PRINTER OCL Statement" on page 5-83 for more information about specifying printer information.

This section shows an example based on the statements you can use instead of the supplied procedure.

Example

This example shows how you can create your own version of the LISTLIBR procedure to cause output to be printed with a priority of 0. This priority causes output to be held on the spool file until it is specifically released for printing. You could, for example, run this sample procedure and then run the COPYPRT procedure to display the output.

The name of the procedure is LSAMPLE. If you want to test it, you will have to enter it into a library procedure member yourself. See the "List Library Members and Information (LISTLIBR Procedure)" on page A-75 and the "PRINTER OCL Statement" on page 5-83 for information about the statements used in this procedure.

```
* LSAMPLE procedure
// LOAD    $MAINT
// PRINTER NAME-$SYSLIST,SPOOL-YES,PRIORITY-0
// RUN
// COPY    TO-PRINT,NAME-?1R?,LIBRARY-?2'S'?,FROM-?3'?CLIB?'?
// END
// RETURN
*
* Sample procedure (based on LISTLIBR procedure)
*
* Parameter      Entry          Meaning
* ------------   -----------    ------------------------------
* Parameter 1:   member name    Name of member to list
*                DIR            List library directory
*                ALL            List all members
* ------------   -----------    ------------------------------
* Parameter 2:   S              List source members (default)
*                P              List procedure members
*                O              List load members
*                R              List subroutine members
*                ALL            List all member types
* ------------   ------------   ------------------------------
* Parameter 3:   library name   Default is the current library
```

The statements in the LSAMPLE procedure indicate the following:

* An asterisk (*) in the first column indicates that the statement is a comment statement. Any information following the * is not processed.

LOAD indicates the $MAINT utility program is to be loaded.

PRINTER indicates how the system list output printed by $MAINT is to be changed. The NAME-$SYSLIST parameter indicates that system list output is to be modified by the statement; the SPOOL-YES parameter indicates the system list output is to be spooled. The PRIORITY-0 parameter causes the output to be held on the spool file; that is, the output is not printed until it is released (by using the RELEASE command, for example).

RUN indicates that the $MAINT program should begin running. The $MAINT program then reads the COPY and END statements.

COPY indicates that the specified library member, or directory (if DIR is specified), is to be listed.

END indicates the end of the $MAINT utility control statements.

RETURN: The procedure ends. The following comment statements are not processed by the system.

To run the LSAMPLE procedure, first use SEU to enter it into a library procedure member named LSAMPLE. You do not have to enter the comment (*) statements. After you have entered the procedure, you can run it. For example, to list a library source member named TEST from a library named MYLIB, you would enter:

```
LSAMPLE TEST,,MYLIB
```

To list the directory of the library MYLIB, you would enter:

```
LSAMPLE DIR,ALL,MYLIB
```

$ARSP Utility

The $ARSP utility program allows you to:

- Change the automatic response values and the severity levels in a message load member (RESPONSE procedure).
- Change the alert indicators for messages in a message load member (SETALERT procedure).

```
// LOAD $ARSP
// RUN

// RESPONSE SOURCE-source member name [,LIBRARY-{library name}] [,ALERTS-{YES}]
                                      {#LIBRARY  }             {NO }

// END
```

S9020364-1

Changing Automatic Response Values (RESPONSE Procedure)

See the "RESPONSE Procedure" on page 4-374 for more information.

Example

To apply the automatic responses contained in the library source member named AUTORESP (which is stored in a library named MYLIB):

```
// LOAD $ARSP
// RUN
// RESPONSE SOURCE-AUTORESP,LIBRARY-MYLIB
// END
```

Changing Alert Indicators (SETALERT Procedure)

See the "SETALERT Procedure" on page 4-457 for more information.

Example

To set the alert indicators for messages specified in the library source member PROGALRT (which is stored in a library named MYLIB):

```
// LOAD $ARSP
// RUN
// RESPONSE SOURCE-PROGALRT,LIBRARY-MYLIB,ALERTS-YES
// END
```

$BICR Utility

The $BICR utility program allows you to:

- List a basic data exchange or I exchange file on diskette (LISTFILE procedure).
- Converts a basic data exchange or I exchange diskette file to a sequential or an indexed disk file (TRANSFER procedure).
- Adds a diskette file that is in basic data exchange or I exchange format to an existing sequential disk file (TRANSFER procedure).
- Converts a disk file or adds records from a disk file to a basic data exchange or I exchange diskette file (TRANSFER procedure).

Listing Diskette Files (LISTFILE Procedure)

See the "LISTFILE Procedure" on page 4-267 for more information.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-input file name,UNIT-I1
// RUN

// DISPLAY [FROM-first record [,TO-last record]]

// END
```

S9020365-0

Only some of the parameters are shown for the FILE OCL statement. For information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Disk Files)" on page 5-32.

The **DISPLAY** statement causes the file to be listed on the system list device. You can use the STATUS SESSION command to determine your system list device.

FROM-first record specifies the first record to be displayed. If FROM is not specified, the file is listed beginning with the first record.

TO-last record specifies the last record to be displayed. If TO is not specified, the file is listed ending with the last record. The maximum value that can be specified is 8000000.

Example

To list a basic data exchange diskette file named BASICDAT.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-BASICDAT,UNIT-I1
// RUN
// DISPLAY
// END
```

Copying Files (TRANSFER Procedure)

See the "TRANSFER Procedure" on page 4-542 for more information.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
                                              {I1}
// FILE NAME-COPYO,LABEL-output file name,UNIT-{F1}
                                               {I1}
// RUN
// TRANSFER [ADD-{NO }] [,KEYLEN-key length,KEYLOC-key location]
                {YES}
            [,FORMAT-{EXCHANGE}]
                     {IFORMAT }
// END
```

S9020366-0

Only some of the parameters are shown for the FILE OCL statement. For information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Disk Files)" on page 5-32.

Example 1

To create a disk sequential file named FILE2 from a diskette basic data exchange or I exchange file named FILE2.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE2,UNIT-I1
// FILE NAME-COPYO,LABEL-FILE2,UNIT-F1
// RUN
// TRANSFER
// END
```

Example 2

To create an indexed disk file named FILE2 from a basic data exchange file named FILE2. The key is to be in positions 1 through 5.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE2,UNIT-I1
// FILE NAME-COPYO,LABEL-FILE2,UNIT-F1
// RUN
// TRANSFER ADD-NO,KEYLEN-5,KEYLOC-1
// END
```

Example 3

To add a basic data exchange or I-exchange diskette file named FILE1 to an existing disk file named FILE1.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE1,UNIT-I1
// FILE NAME-COPYO,LABEL-FILE1,UNIT-F1
// RUN
// TRANSFER ADD-YES
// END
```

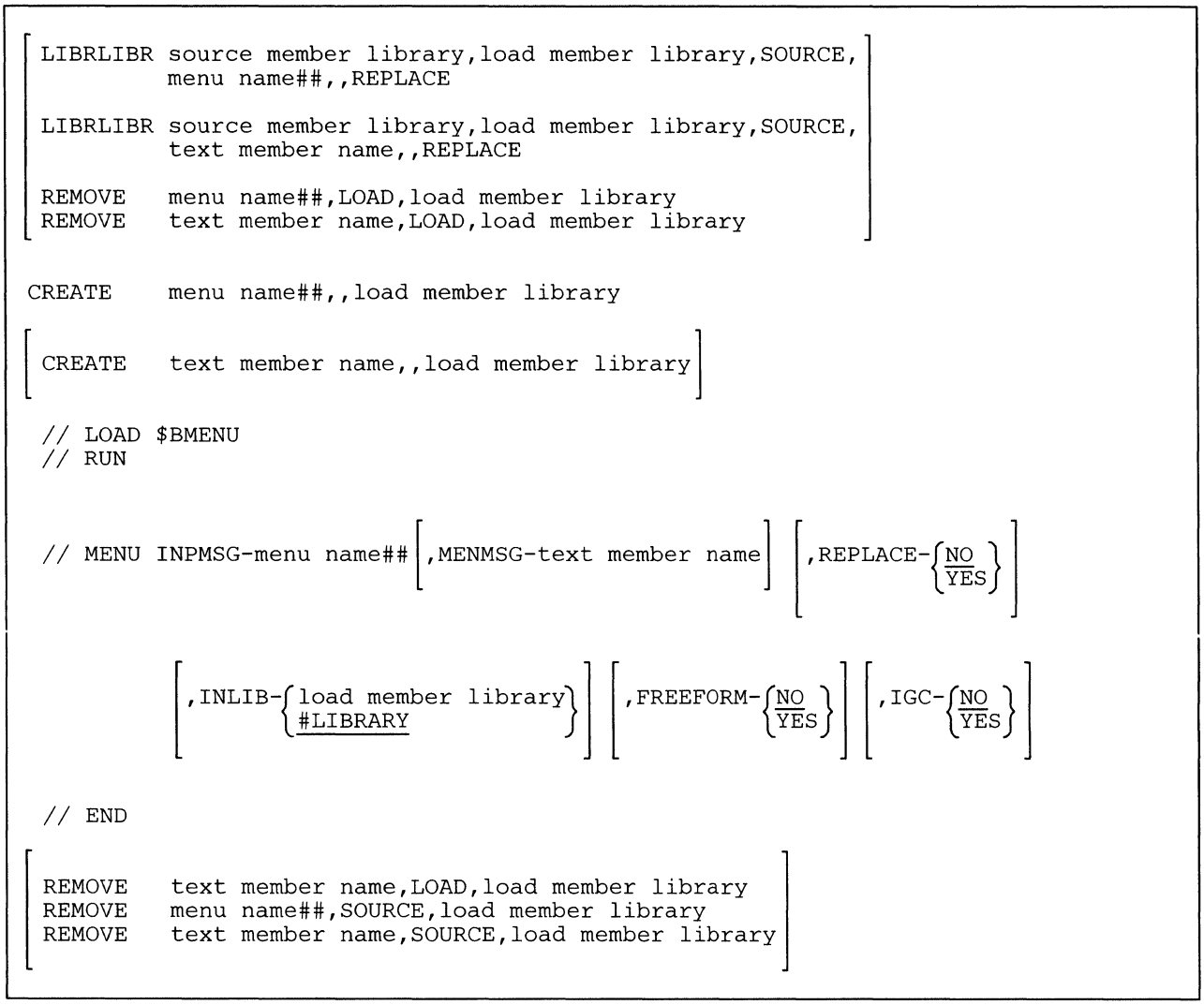
Example 4

To create an I-exchange diskette file named FILE3 on a diskette from a disk file named FILE3. The file is to be saved for 30 days. The volume ID of the diskette is VOL002.

```
// LOAD $BICR
// FILE NAME-COPYIN,LABEL-FILE3,UNIT-F1
// FILE NAME-COPYO,LABEL-FILE3,UNIT-I1,PACK-VOL002,RETAIN-30
// RUN
// TRANSFER FORMAT-IFORMAT
// END
```

$BMENU Utility

The $BMENU utility creates the library members required to display a menu. See the "BLDMENU Procedure" on page 4-66 for more information.



S9020367-0

Example

You have created a command source member named MENU## and a display text source member named MENUDT; both members are stored in a library named MYLIB. To build a menu named MENU, using the BLDMENU procedure, you could enter the following. The library members created by BLDMENU are to be placed in a library called MYLIB, and the display text load member created by BLDMENU is to be deleted from MYLIB before BLDMENU ends.

```
CREATE MENU##,,MYLIB
CREATE MENUDT,,MYLIB
// LOAD $BMENU
// RUN
// MENU INPMSG-MENU##,MENMSG-MENUDT,INLIB-MYLIB,
//      REPLACE-NO,FREEFORM-NO
// END
REMOVE MENUDT,LOAD,MYLIB
```

$BUILD Utility

The $BUILD utility displays data on the disk after a disk error occurs. You can then correct the displayed data. See the "BUILD Procedure" on page 4-70 for more information.

```
// SWITCH XXXXXXX0
// LOAD $BUILD
// PRINTER NAME-$SYSLIST,SPOOL-NO
// RUN

[COMPRESS]
```

S9020368-0

Example

To check a file after a disk file input/output error, you would enter the following:

```
// SWITCH XXXXXXX0
// LOAD $BUILD
// PRINTER NAME-$SYSLIST,SPOOL-NO
// RUN
```

$COPY Utility

The $COPY utility program allows you to:

- Copy disk files (COPYDATA procedure)
- List disk, diskette, or tape files (LISTDATA and LISTFILE procedures)
- Restore disk files from diskette or tape (RESTORE procedure)
- Restore the network resource directory from diskette or tape (RESTNRD procedure)
- Save disk files on diskette or tape (SAVE procedure)
- Save the network resource directory on diskette or tape (SAVENRD procedure)

Also shown in this section are $COPY utility control statements that are supported for compatibility with the IBM System/34.

Copying Disk Files (COPYDATA Procedure)

See the "COPYDATA Procedure" on page 4-111 for more information.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-copy from file name,UNIT-F1
// FILE NAME-COPYO,LABEL-copy to file name,UNIT-F1
// RUN

// COPYFILE [OUTPUT-{SAME}] [,{INCLUDE}-{EQ}] [,POSITION-position]
                    {SEQUENTL}          {OMIT}  {NE}
                    {(S)     }                  {LT}
                    {INDEXED }                  {LE}
                    {(I)     }                  {GT}
                    {DIRECT  }                  {GE}
                    {(D)     }

          [,CHAR-{'characters'}] [,REORG-{NO}] [,RECL-record length]
                 {Xdddd...dd }           {YES}

          [,LIMIT-maximum records]

[// SELECT {RECORD},FROM-starting value[,TO-ending value]]
          {KEY   }
          {PKY   }

[// KEY   POSITION-key position,LENGTH-key length]

// END
```

S9020369-0

Only some of the parameters for the FILE OCL statement are shown. For more information on the FILE statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32.

Note: If DISP-OLD is specified in the FILE OCL statement containing NAME-COPYO, the COPYO file will not be deleted from disk if you select option 2 or 3 in response to an error message or while in Inquiry mode during the $COPY job step or job. In these cases, the COPYO file may contain invalid records.

Differences from the Procedure Parameters

For the **COPYFILE** statement:

CHAR-'characters' allows you to specify up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, o'clock would be entered: 'o''clock'.

If the system contains the ideographic version of the SSP, the character string may contain a mixture of ideographic and nonideographic characters. An ideographic character string must be surrounded by a shift-out (0E) character and shift-in (0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each ideographic character occupies two characters in the character string while each shift-out and shift-in character occupies one character.

Note: If two nonideographic characters happen to have an EBCDIC value equivalent to the 2 bytes of an ideographic character specified in the comparison string and they lie in the position specified, that record may be included or omitted (depending on what was specified earlier, the INCLUDE or OMIT keyword). Care should be taken when dealing with files whose records are not all in the same format.

Xdddd...dd can also specify the comparison characters. This form specifies hexadecimal data. Up to 15 bytes can be specified, where 2 digits (dd) represent the hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must precede the hexadecimal digits.

The **SELECT** statement allows you to further define the records to be copied.

RECORD specifies that a portion of the file is to be copied. When RECORD is specified, the FROM parameter and the TO parameter (if TO is used) must specify relative record numbers.

FROM-starting value specifies the relative record number of the first record to be copied. For example, if the first record to be copied is the fifth record in the file, FROM-5 should be specified. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same relative record number.

The FROM value must be less than or equal to the TO value.

TO-ending value specifies the relative number of the last record to be copied. For example, if the last record to be copied is the fifteenth record in the file, TO-15 should be specified. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same relative record number. If an ending value is not specified, the file is copied through the last record in the file.

The TO value must be greater than or equal to the FROM value.

KEY or PKY specifies that a specified portion of an indexed file is to be copied. PKY must be specified when the indexed file contains packed keys. If regular keys are used, up to 120 characters can be specified. If packed keys are used, up to 239 characters can be specified.

FROM-starting value specifies the key (or the beginning characters of the key) of the first record to be copied. The characters must be enclosed in apostrophes ('). If none of the keys in the file begin with the specified characters, the record with the next higher key is the first record to be copied or displayed. For example, if FROM-'NAME1' is specified, the first key beginning with NAME1 or a larger value is the key of the first record to be copied.

The FROM value must be less than or equal to the TO value. If only one record is to be copied, the FROM and TO parameters must specify the same key.

TO-ending value specifies the key (or the beginning characters of the key) of the last record to be copied. The ending value must be enclosed in apostrophes (').

The TO value must be greater than or equal to the FROM value.

If none of the keys in the file begin with the specified characters, the record with the next lower key is the last record to be copied. For example, if TO-'34' is specified, the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record to be copied.

If the TO parameter is not specified, $COPY uses the last key in the index as the TO key. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same key.

Example 1

This example shows how to copy FILE1 and create an exact copy of it, FILE2.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE2
// RUN
// COPYFILE
// END
```

Example 2

This example shows how to copy a file named FILE1, which is an indexed file, and create a new file named FILE3. The new file is to be an indexed file and the key is to be in positions 5 through 24.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE3
// RUN
// COPYFILE OUTPUT-INDEXED
// KEY POSITION-5,LENGTH-20
// END
```

Example 3

This example shows how to use the SELECT statement to copy only those records that have the relative record numbers 2, 3, and 4.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE7
// RUN
// COPYFILE
// SELECT RECORD,FROM-2,TO-4
// END
```

Listing $COPY Files (LISTDATA/LISTFILE Procedures)

See the "LISTDATA Procedure" on page 4-261 for more information.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
                                               {I1}
                                               {T1}
                                               {T2}
                                               {TC}

// RUN

// COPYFILE  OUTPUT-{CHAR   } [,{INCLUDE}-{EQ}] [,POSITION-position]
                    {HEX    }  {OMIT   } {NE}
                    {PARTHEX}           {LT}
                    {CRT    }           {LE}
                                        {GT}
                                        {GE}

          [,CHAR-{'characters'}] [,REORG-{NO }] [,RECL-record length]
                 {Xdddd...dd }           {YES}

          [,LIMIT-maximum records] [,IGC-{YES}]
                                         {NO }

[// SELECT {RECORD},FROM-starting value[,TO-ending value]]
          {KEY   }
          {PKY   }

// END
```

S9020370-2

For more information about the FILE statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

If T1, T2, or TC is specified for the UNIT parameter in the FILE statement, then RECFM-FB, RECL-256, and BLKL-24576 must also be specified.

For the **COPYFILE** statement:

CHAR-'characters' allows you to specify up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, o'clock would be entered: 'o''clock'.

If the system contains the ideographic version of the SSP, the character string may contain a mixture of ideographic and nonideographic characters. An ideographic character string must be surrounded by a shift-out (0E) character and shift-in (0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each ideographic character occupies 2 characters in the character string while each shift-out and shift-in character occupies one character.

Note: If two nonideographic characters happen to have an EBCDIC value equivalent to the 2 bytes of an ideographic character specified in the comparison string and they lie in the position specified, that record may be included or omitted (depending on what was specified earlier, the INCLUDE or OMIT keyword). Care should be taken when dealing with files whose records are not all in the same format.

CHAR-Xdddd...dd specifies hexadecimal data. Up to 15 bytes can be specified, where 2 digits (dd) represent the hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must come before the hexadecimal digits.

For the **SELECT** utility control statement:

RECORD specifies that a portion of the file is to be listed. When RECORD is specified, the FROM parameter and the TO parameter (if TO is used) must specify relative record numbers.

FROM-starting value specifies the relative record number of the first record to be listed. For example, if the first record to be copied is the fifth record in the file, FROM-5 should be specified. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same relative record number.

TO-ending value specifies the relative number of the last record to be listed. For example, if the last record to be copied is the fifteenth record in the file, TO-15 should be specified. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same relative record number. The TO parameter cannot be specified unless the FROM parameter is also specified. If an ending value is not specified, the file is listed through the last record in the file.

KEY or PKY specifies that a specified portion of an indexed file is to be listed. PKY must be specified if the indexed file contains packed keys. If packed keys are used, up to 239 numeric characters can be specified. If regular keys are used, up to 120 characters can be specified.

FROM-starting value specifies the key (or the beginning characters of the key) of the first record to be listed. The starting value must be enclosed in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record to be listed. For example, if FROM-'15' is specified, the first key beginning with 15 or more is the key of the first record to be listed. If only one record is to be listed, the FROM and TO parameters must specify the same key.

TO-ending value specifies the key (or the beginning characters of the key) of the last record to be listed. The ending value must be enclosed in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record to be listed. For example, if TO-'34' is specified, the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record to be listed. The TO parameter cannot be specified unless the FROM parameter is also specified.

If the TO parameter is not specified, $COPY uses the last key in the index as the TO key. If only one record is to be listed, the FROM and TO parameters must specify the same key.

IGC is used for the ideographic version of the SSP and is ignored for nonideographic systems. IGC-YES specifies that the file might contain ideographic characters, and if possible, $COPY should display those characters. When OUTPUT-CHAR, OUTPUT-HEX, or OUTPUT-PARTHEX is specified along with IGC-YES, the hexadecimal representation of ideographic characters are not displayed, even if the characters are not displayable.

Note: When a file is displayed on a system with the ideographic version of the SSP and IGC-YES is specified or defaulted to, all occurrences of bytes with a hexadecimal value of OE will be interpreted as shift-out characters and all occurrences of bytes with a hexadecimal value of OF will be interpreted as shift-in characters. An attempt will be made to interpret all pairs of bytes in between such OE and OF bytes as ideographic characters. To avoid misinterpretations, care should be exercised when displaying the contents of a file that includes bytes with hexadecimal values of OE or OF, which are not intended to be shift-out or shift-in characters.

Whenever the SYSLIST device is capable of displaying ideographic characters and the system supports the ideographic feature, the records displayed by $COPY will be displayed in ideographic format.

Note: When nonideographic characters are printed, 100 characters are printed on each line. The characters are printed in print positions 1 through 100. When a mixture of ideographic and nonideographic characters are printed, the number of characters printed on a line is variable. Each ideographic character represents 2 bytes of information and requires 2 print positions on the line. If an ideographic character begins in print position 100, position 100 is left blank and the character is printed in positions 0 and 1 on the next line. (Position 0 is staggered 1 position to the left of the normal starting position, which is position 1.) To determine the position of a character within the record, you can use the following algorithm:

```
position in    = 100 x   number of       +    print position of
record                   previous lines       the character
```

For example, if an ideographic character begins in position 0 of the third line of printout, the character begins in byte 200 of the record (100 x 2 + 0 = 200).

IGC-NO specifies that the file contents should be displayed just as they would be for a nonideographic system. Any ideographic data in the file is treated as 1-byte alphameric characters.

Example 1

Print only the printable characters in all the records of the file named FILE1.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-CHAR
// END
```

Example 2

Print only records 5 through 10 in the disk file named FILE1. Both the characters in the records and their hexadecimal representations are to be printed.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-HEX
// SELECT RECORD,FROM-5,TO-10
// END
```

Example 3

Display all records on a tape file called FILE1. The tape is mounted on tape drive 2.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-T2,RECFM-FB,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-CRT
// END
```

Example 4

Display the contents of the tape cartridge file named DATAFILE on the system list device.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-TC,LABEL-DATAFILE
// RUN
// COPYFILE OUTPTX-PRINT
// END
```

Restoring Files (RESTORE Procedure)

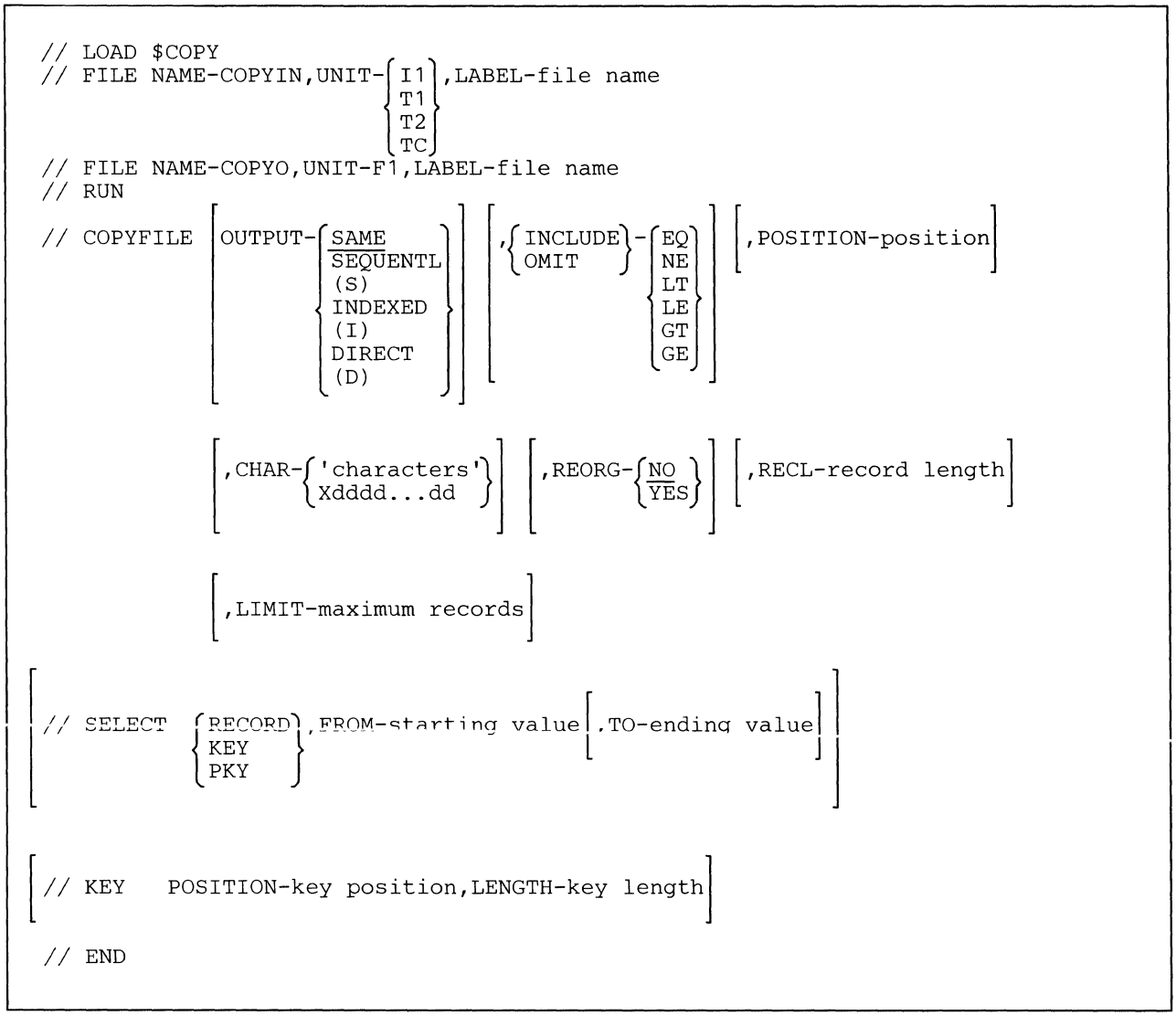
See the "RESTORE Procedure" on page 4-392 for more information.

For restoring all previously saved files of a set:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-{I1|T1|T2|TC},LABEL-set name
// FILE NAME-COPYO,UNIT-F1
// RUN
// COPYALL TO-F1[,STRTLABL-starting file name] [,STRTDATE-starting file date]


// END
```

S9020371-1

For restoring a single previously saved file:



S9020372-1

Only a few parameters for the FILE statement are shown. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

If T1, T2, or TC is specified for the UNIT parameter in the FILE statement, then RECFM-FB, RECL-256, and BLKL-24576 must be specified.

If T1, T2, or TC is specified, no parameters may be specified with the COPYFILE statement. The SELECT and KEY statements are also invalid.

For the **COPYFILE** statement:

CHAR-'characters' specifies up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, o'clock would be entered: 'o''clock'.

If the system contains the ideographic version of the SSP, the character string may contain a mixture of ideographic and nonideographic characters. An ideographic character string must be surrounded by a shift-out (0E) character and shift-in (0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each ideographic character occupies 2 characters in the character string while each shift-out and shift-in character occupies one character.

Note: If two nonideographic characters happen to have an EBCDIC value equivalent to the 2 bytes of an ideographic character specified in the comparison string and it lies in the position specified, that record may be included or omitted (depending on what was specified earlier, the INCLUDE or OMIT keyword). Care should be taken when dealing with files whose records are not all in the same format.

CHAR-Xdddd...dd specifies hexadecimal data. Up to 15 bytes can be specified, where 2 digits (dd) represent the hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must come before the hexadecimal digits.

LIMIT-maximum records specifies the total number of records to be copied into the restored file. Any number greater than 0 can be entered. Note that the SSP cannot create a file that contains more than 16711408 records.

The **SELECT** statement allows you to further select which records you want to copy.

RECORD specifies that a portion of the file is to be processed. When RECORD is specified, the FROM parameter and the TO parameter (if TO is used) must specify relative record numbers.

FROM-starting value specifies the relative record number of the first record to be processed. For example, if the first record to be copied is the fifth record in the file, FROM-5 should be specified. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same relative record number.

TO-ending value specifies the relative number of the last record to be processed. For example, if the last record to be copied is the fifteenth record in the file, TO-15 should be specified. If no TO parameter is specified, $COPY uses the last record in the file as the TO record. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same relative record number.

KEY or PKY specifies that a portion of an indexed file is to be copied. PKY must be specified if the indexed file contains packed keys. If packed keys are used, up to 239 digits can be specified. If regular keys are used, up to 120 characters can be specified.

When SELECT KEY or PKY is used to select records from an indexed file, all records whose keys are within the limits specified will be copied to the output file, but the output records will be in key sequence only if the file had been saved in key sequence. For information on ordering the records of an indexed file in key sequence, see the "SAVE Procedure" on page 4-416.

FROM-starting value specifies the key (or the beginning characters of the key) of the first record to be processed. The starting value must be enclosed in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record to be processed. For example, if FROM-'15' is specified, the first key beginning with 15 or more is the key of the first record to be processed.

The FROM value must be less than or equal to the TO value. If only one record is to be copied or displayed, the FROM and TO parameters must specify the same key.

TO-ending value specifies the key (or the beginning characters of the key) of the last record to be processed. The starting value must be enclosed in apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record to be processed. For example, if TO-'34' is specified, the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record to be processed.

The TO value must be greater than or equal to the FROM value. If the TO parameter is not specified, $COPY uses the last key in the index as the TO key. If only one record is to be processed, the FROM and TO parameters must specify the same key.

The **KEY** statement allows you to specify the position and length of the key for indexed files. If either the key position or key length is entered, the other must also be entered.

POSITION-key position specifies the starting position of the key for the restored file. The key position must be specified if the restored file is to be an indexed file, but the saved file was not organized as an indexed file. The key position can be any number from 1 through 4096. If a value is not specified, the key position for the indexed file that was saved is assumed.

LENGTH-key length specifies the length of the key for the restored file. The key length must be specified if the restored file is to be an indexed file, but the saved file was not organized as an indexed file. The key length can be any number from 1 through 120. If a value is not specified, the key length for the indexed file that was saved is assumed.

Example 1

This example shows how to restore all the files that were saved on diskette using the SAVE ALL procedure. The diskettes are in magazine slots M1 and M2, and the drive is to automatically advance to M2 after processing the diskettes in M1.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-I1,LABEL-#SAVE,LOCATION-M1
// FILE NAME-COPYO,UNIT-F1
// RUN
// COPYALL TO-F1
// END
```

Example 2

This example shows how to restore a diskette file named FILE1. The key is to be changed from positions 1 through 4 to positions 5 through 24, and only those records that contain the phrase NEW anywhere in the record are to be copied.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-I1,LABEL-FILE1,LOCATION-S1,AUTO-NO
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE1
// RUN
// COPYFILE OUTPUT-I,INCLUDE-EQ,CHAR-'NEW'
// KEY POSITION-5,LENGTH-20
// END
```

Example 3

This example shows how to restore all the files that were saved on tape, using the SAVE ALL procedure. The tape is located on tape drive 1 and is to be rewound after the files are restored.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-T1,LABEL-#SAVE,RECFM-FB,RECL-256,
//      BLKL-24576,END-REWIND
// FILE NAME-COPYO,UNIT-F1
// RUN
// COPYALL TO-F1
// END
```

Example 4

This example shows how to restore a file named DATAFILE from tape cartridge to disk. The disk file name will be SAVEDATA.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-TC,LABEL-DATAFILE
// FILE NAME-COPYO,UNIT-F1,LABEL-SAVEDATA
// RUN
// COPYFILE
// END
```

Restoring the Network Resource Directory (RESTNRD Procedure)

The $COPY utility program allows you to restore the network resource directory from diskette or tape to disk. See the "$SINCT Utility" on page A-110 for information about how to restore the network resource directory as a system file.

See the "RESTNRD Procedure" on page 4-389 for more information.

```
// MEMBER USER1-#SI#M1,USER2-#SI#M2
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-{I1}
                         {T1}
                         {T2}
                         {TC}
// FILE NAME-COPYO,LABEL-#NRD.FLE,UNIT-F1,RETAIN-J
// RUN
// COPYFILE OUTPUT-SAME,SYSFILE-NRD
// END
```

S9020488-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Saving Files (SAVE Procedure)

See the "SAVE Procedure" on page 4-416 for more information.

To save a file on diskette, tape, or tape cartridge:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-file name
// FILE NAME-COPYO,UNIT-{I1|T1|T2|TC},LABEL-file name
// RUN
// COPYFILE [OUTPUT-{SAME|SEQUENTL|(S)|INDEXED|(I)|DIRECT|(D)}] [,REORG-{NO|YES}] [,{INCLUDE|OMIT}-{EQ|NE|LT|LE|GT|GE}]

         [,POSITION-position] [,CHAR-{'characters'|Xdddd...dd}]

         [,RECL-record length] [,LIMIT-record limit] [,COMPRESS-{NO|YES}]

[// SELECT {RECORD|KEY|PKY},FROM-starting value[,TO-ending value]]

[// KEY POSITION-key position,LENGTH-key length]

// END
```

S9020373-1

To save all files, file groups, or all files except file groups on diskette, tape, or tape cartridge:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1
// FILE NAME-COPYO,UNIT-{I1},LABEL-set name
                        {T1}
                        {T2}
                        {TC}
// RUN

// COPYALL TO-{I1} [,GROUP-{ALL       }] [,COMPRESS-{NO }]
              {T1}      {file group}      {YES}
              {T2}
              {TC}

// END
```

S9020374-1

To add a disk file to an existing diskette file:

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-file name
// FILE NAME-COPYO,UNIT-I1,LABEL-file name,PACK-volume id
// RUN
// COPYADD
// END
```

S9020375-0

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statements, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

If T1, T2, or TC is specified for the UNIT parameter in the FILE statement, then RECFM-FB, RECL-256, and BLKL-24576 must also be specified.

If T1, T2, or TC is specified, no parameters may be specified with the COPYFILE statement. The SELECT and KEY statements are also invalid.

For the **COPYFILE** statement:

OUTPUT-SAME specifies that the restored file is to have the same organization as the file that was saved. If a parameter is not specified, SAME is assumed.

OUTPUT-SEQUENTL specifies that the restored file is to be organized as a sequential file.

OUTPUT-INDEXED specifies that the restored file is to be organized as an indexed file.

OUTPUT-DIRECT specifies that the restored file is to be organized as a direct file.

CHAR-'characters' allows you to specify up to 30 characters of data. The data must begin and end with apostrophes ('). You can also specify apostrophes in the character string by entering two apostrophes. For example, o'clock would be entered: 'o''clock'.

If the system contains the ideographic version of the SSP, the character string may contain a mixture of ideographic and nonideographic characters. An ideographic character string must be surrounded by a shift-out (0E) character and shift-in (0F) character. If there is a shift-out character in the first position of the total character string, that shift-out character is not included for comparison. Likewise, if there is a shift-in character in the last position of the string, that shift-in character is not included for comparison. All other shift-out and shift-in characters are part of the comparison string. Each ideographic character occupies 2 characters in the character string while each shift-out and shift-in character occupies one character.

Note: If two nonideographic characters happen to have an EBCDIC value equivalent to the 2 bytes of an ideographic character specified in the comparison string and they lie in the position specified, that record may be included or omitted (depending on what was specified earlier, the INCLUDE or OMIT parameter). Care should be taken when dealing with files whose records are not all in the same format.

CHAR-Xdddd...dd specifies the comparison characters in hexadecimal form. Up to 15 bytes can be specified, where 2 digits (dd) represent the hexadecimal byte. The hexadecimal digits are 0 through 9 and A through F. The X must precede the hexadecimal digits.

RECL specifies the record length of the new file, and can be any number from 1 through 4096. If this parameter is not entered, the record length of the file to be saved is used.

If the record length of the file being saved is less than the entered record length, the additional record positions in the saved file are filled with blanks. If the record length of the file being saved is greater than the entered record length, the extra positions are truncated. If the new file is an indexed file and the key field would be truncated, an error message is displayed.

LIMIT specifies the total number of records to be copied into the new file. Any number greater than 0 can be entered. Note that the SSP cannot create a file that contains more than 16777200 records.

The **SELECT** utility control statement allows you to further select which records are processed.

RECORD specifies that a portion of the file is to be saved. When RECORD is specified, the FROM parameter and the TO parameter (if TO is used) must specify relative record numbers.

FROM-starting value specifies the relative record number of the first record to be saved. For example, if the first record to be saved is the fifth record in the file, FROM-5 should be specified. If only one record is to be saved, the FROM and TO parameters must specify the same relative record number.

TO-last record specifies the relative number of the last record to be saved. For example, if the last record to be saved is the fifteenth record in the file, TO-15 should be specified. If only one record is to be saved, the FROM and TO parameters must specify the same relative record number. If the TO parameter not is specified, $COPY uses the last record in the file as the TO record.

KEY or PKY specifies that a specified portion of an indexed file is to be saved. PKY must be specified if the indexed file contains packed keys. If packed keys are used, up to 239 numeric characters can be specified. If regular keys are used, up to 120 characters can be specified.

FROM-starting value specifies the key (or the beginning characters of the key) of the first record to be saved. The starting value must be enclosed by apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record to be saved. For example, if FROM-'15' is specified, the first key beginning with 15 or more is the key of the first record to be saved.

The FROM value must be less than or equal to the TO value. If only one record is to be saved, the FROM and TO parameters must specify the same key.

TO-ending value specifies the key (or the beginning characters of the key) of the last record to be saved. The ending value must be enclosed by apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record to be saved. For example, if TO-'34' is specified, the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record to be saved.

The TO value must be greater than or equal to the FROM value. If only one record is to be saved, the FROM and TO parameters must specify the same key. If the TO parameter is not specified, $COPY uses the last key in the index as the TO key.

The **KEY** statement allows you to specify the position and length of the key for newly created indexed files. If either the key position or key length is entered, the other must also be entered.

POSITION-key position specifies the starting position of the key for the saved file. The key position can be any number from 1 through 4096. If a value is not specified, the current key position for the indexed file is assumed.

LENGTH-key length specifies the length of the key for the saved file. The key length can be any number from 1 through 120. If a value is not specified, the current key length for the indexed file is assumed.

For the **COPYALL** statement:

GROUP specifies whether members of a file group are to be saved. If no GROUP parameter is specified, only files that are not members of file groups are saved.

ALL specifies that all files and all members of file groups are to be saved.

group name specifies that all the members of the specified file group are to be saved. Those files that are not part of the file group are not saved.

For the **COPYADD** statement:

COPYADD specifies that a file is to be added to an existing diskette file.

Example 1

This example shows how to save all disk files on diskette for a period of 7 days. The diskettes have volume IDs of VOL001, and are located in magazines M1 and M2, starting in position M1.01. The files are compressed.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1
// FILE NAME-COPYO,UNIT-I1,LABEL-#SAVE,PACK-VOL001,RETAIN-7,
//      LOCATION-M1.01
// RUN
// COPYALL TO-I1,GROUP-ALL,COMPRESS-YES
// END
```

Example 2

Save a file named FILE1 and add this file to an existing diskette file named FILE2. The volume ID of the diskette is VOL001.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-I1,LABEL-FILE2,PACK-VOL001
// RUN
// COPYADD
// END
```

Example 3

Save all files belonging to file group A1 to diskette. The name to be associated with the set of saved files is SAVEA1. The volume ID of the diskette is VOL002, and the file will be saved for 39 days.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1
// FILE NAME-COPYO,UNIT-I1,LABEL-SAVEA1,PACK-VOL002,RETAIN-39
// RUN
// COPYALL TO-I1,GROUP-A1
// END
```

Example 4

This example shows how to save specific records from a single file named FILE3 on a diskette. The only records to be saved contain the word SAVE in positions 10 through 13 of the record.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE3
// FILE NAME-COPYO,UNIT-I1,LABEL-FILE3,PACK-VOL001
// RUN
// COPYFILE INCLUDE-EQ,POSITION-10,CHAR-'SAVE'
// END
```

Example 5

This example shows how to save only records 5 through 10 of a disk file named FILE4 to diskette. The volume ID of the diskette is VOL001.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE4
// FILE NAME-COPYO,UNIT-I1,LABEL-FILE4,PACK-VOL001
// RUN
// COPYFILE
// SELECT RECORD,FROM-5,TO-10
// END
```

Example 6

This example shows how to save a disk file named FILE4 to tape. The volume ID of the tape is VOL001 and it is to be saved to tape drive 2. After the save, the tape position is to be left at the end of the file.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-FILE4,UNIT-F1
// FILE NAME-COPYO,LABEL-FILE4,UNIT-T2,RECFM-FB,
//      RECL-256,BLKL-24576,VOLID-VOL001,END-LEAVE
// RUN
// COPYFILE
// END
```

Example 7

This example shows how to save a disk file named DATAFILE to tape cartridge. The name on the tape has been changed to SAVEDATA.

```
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-DATAFILE
// FILE NAME-COPYO,UNIT-TC,LABEL-SAVEDATA,RECL-256,BLKL-24576,RECFM-FB,
//      VOLID-TEST1
// RUN
// COPYFILE
// END
```

Saving the Network Resource Directory (SAVENRD Procedure)

The $COPY utility program allows you to save the network resource directory on diskette or tape.

See the "SAVENRD Procedure" on page 4-434 for more information.

```
// MEMBER USER1-#SI#M1,USER2-#SI#M2
// LOAD $COPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-#NRD.FLE
// FILE NAME-COPYO,UNIT-{I1|T1|T2|TC}
// RUN
// COPYFILE OUTPUT-SAME,SYSFILE-NRD
// END
```

S9020489-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

IBM System/34 Compatible Statements

These statements are supported for compatibility with the IBM System/34 for copying files.

To copy a file:

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
                                              {I1}
// FILE NAME-COPYO,LABEL-output file name,UNIT-{F1}
                                               {I1}
// RUN
// COPYFILE  OUTPUT-DISK [,DELETE-{'position,c'   }] [,REORG-{NO }]
                                  {'position,cc'  }          {YES}
                                  {'position,Xdd' }
                                  {'position,Xdddd'}
                                  {SYSDEL         }
[// SELECT  {RECORD},FROM-starting value [,TO-ending value]]
           {KEY   }
           {PKY   }
[// KEY POSITION-key position,LENGTH-key length]
// END
```

S9020376-0

To list a file:

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
                                              {I1}
// RUN

// COPYFILE {OUTPUT-PRINT} [,DELETE-{'position,c'    }] [,REORG-{NO }]
            {OUTPTX-PRINT}          {'position,cc'   }          {YES}
                                    {'position,Xdd'  }
                                    {'position,Xdddd'}
                                    {SYSDEL          }

[
// SELECT  {RECORD},FROM-starting value[,TO-ending value]
           {KEY   }
           {PKY   }
]

// END
```

S9020377-0

Only some of the parameters are shown for the FILE OCL statement. For information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Disk Files)" on page 5-32.

For the **COPYFILE** statement:

OUTPUT-DISK specifies that the output is to be written on disk or diskette. The COPYIN and COPYO FILE OCL statements indicate the file to be copied (COPYIN) and the new file (COPYO).

OUTPUT-PRINT or OUTPTX-PRINT specifies that all or part of a file is to be listed on the system list device for the display station. This parameter is supported for compatibility with the IBM System/34.

DELETE specifies records that should be deleted. The DELETE parameter is optional and is supported for compatibility with the IBM System/34.

'position,c' specifies that any record that has the specified character in the specified position is to be deleted. For example, DELETE-'50,S' deletes all records that have an S in position 50.

'position,cc' specifies that any record that has the 2 specified characters in the specified beginning position is to be deleted. For example, DELETE-'50,ST' deletes all records that have ST in positions 50 and 51.

'position,Xdd' specifies that any record that has the specified character in the specified position is to be deleted. Xdd is the hexadecimal value of the character. For example, DELETE-'50,X31' deletes all records that have hex 31 position 50.

'position,Xdddd' specifies that any record that has the 2 specified characters in the specified beginning position is to be deleted. Xdddd are the hexadecimal values of the 2 characters. For example, DELETE-'50,X3132' deletes all records that have hex 3132 positions 50 and 51.

'position,ideographic-constant' specifies that any record that has the specified ideographic character in the specified position is to be deleted. The ideographic constant portion of the parameter must be bracketed by the shift-out (hex '0E') and shift-in (hex '0F') characters.

CAUTION

If 2 nonideographic characters are in the specified position, which happens to have an EBCDIC value equivalent to the 2 bytes of the ideographic character, that record will be deleted. Care should be taken when dealing with files whose records are not all in the same format.

SYSDEL specifies that any system-deleted records are to be deleted from the file.

REORG-YES specifies that:

- For an indexed file, records are copied sequentially by key. Also, deleted records should be deleted or not listed.
- For sequential files, system-deleted records should be deleted.

REORG-NO specifies that records are copied in the order they appear in the file. If no REORG parameter is specified, REORG-NO is assumed.

The **SELECT** utility control statement allows you to select which records are processed.

RECORD specifies that a portion of the file is to be copied. When RECORD is specified, the FROM parameter and the TO parameter (if TO is used) must specify relative record numbers.

FROM-starting value specifies the relative record number of the first record to be copied. For example, if the first record to be copied is the fifth record in the file, FROM-5 should be specified. If only one record is to be copied, the FROM and TO parameters must specify the same relative record number.

TO-last record specifies the relative number of the last record to be copied. For example, if the last record to be copied is the fifteenth record in the file, TO-15 should be specified. If only one record is to be copied, the FROM and TO parameters must specify the same relative record numbers. If the TO parameter is not specified, $COPY uses the last record in the file as the TO record.

KEY or PKY specifies that a specified portion of an indexed file is to be copied. PKY must be specified if the indexed file contains packed keys. If packed keys are used, up to 239 numeric characters can be specified. If regular keys are used, up to 120 characters can be specified.

FROM-starting value specifies the key (or the beginning characters of the key) of the first record to be copied. The starting value must be enclosed by apostrophes ('). If none of the keys in the file begins with the specified characters, the record with the next higher key is the first record to be saved. For example, if FROM-'15' is specified, the first key beginning with 15 or larger is the key of the first record to be saved.

The FROM value must be less than or equal to the TO value. If only one record is to be saved, the FROM and TO parameters must specify the same key.

TO-ending value specifies the key (or the beginning characters of the key) of the last record to be saved. The ending value must be enclosed by apostrophes ('). If packed keys are used, up to 239 numeric characters can be specified. If none of the keys in the file begins with the specified characters, the record with the next lower key is the last record to be saved. For example, if TO-'34' is specified, the last key beginning with 34 (or if no keys begin with 34, the last key that begins with a value smaller than 34) is the key of the last record to be copied.

The TO value must be greater than or equal to the FROM value. If only one record is to be copied, the FROM and TO parameters must specify the same key. If the TO parameter is not specified, $COPY uses the last key in the index as the TO key.

The **KEY** statement allows you to specify the position and length of the key for indexed files. If either the key position or key length is entered, the other must also be entered.

POSITION-key position specifies the starting position of the key for the new file. The key position must be specified if the new file is to be an indexed file but the old file was not organized as an indexed file. The key position can be any number from 1 through 4096. If a value is not specified, the key position for the old indexed file is assumed.

LENGTH-key length specifies the length of the key for the new file. The key length must be specified if the new file is to be an indexed file, but the old file was not organized as an indexed file. The key length can be any number from 1 through 120. If a value is not specified, the key length for the old indexed file is assumed.

Example 1

To save a disk file named FILEA on diskette.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-FILEA,UNIT-F1
// FILE NAME-COPYO,LABEL-FILEA,PACK-VOL001
// RUN
// COPYFILE OUTPUT-DISK
// END
```

Example 2

To list a disk file named FILEA.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-FILEA,UNIT-F1
// RUN
// COPYFILE OUTPUT-PRINT
// END
```

$CPPE Utility

The $CPPE utility program allows you to display a specified error message (ERR procedure).

See the "ERR Procedure" on page 4-176 for more information.

```
// LOAD $CPPE
// RUN
// ERR MIC-message id code,CONTROL-options,ALPHA-USER
// END
```

S9020378-0

Differences from the Procedure Parameters

ALPHA-USER specifies that the current level-one-user-message member is to be used to retrieve the message.

Example

This example shows the $CPPE utility displaying a message with a MIC number of 0014. The options to be allowed are 0, 1, and 3.

```
// LOAD $CPPE
// RUN
// ERR MIC-0014,CONTROL-013,ALPHA-USER
// END
```

$CZUT Utility

The $CZUT utility program allows you to specify which system messages should be considered alert messages.

See the "ALERT Procedure" on page 4-7 for more information.

```
// MEMBER PROGRAM1-#CZ#AU1
// LOAD $CZUT
// RUN
```

S9020379-0

$DCOPY Utility

The $DCOPY utility program allows you to make any number of copies of a single diagnostic (microcode) diskette.

See the "COPYDIAG Procedure" on page 4-120 for more information.

```
// LOAD $DCOPY
// RUN
```

S9020526-0

Example

To copy a diagnostic diskette:

```
// LOAD $DCOPY
// RUN
```

$DDST Utility

The $DDST utility program allows you to sort the index keys for a disk file.

See the "KEYSORT Procedure" on page 4-252 for more information.

```
// LOAD $DDST
// RUN

// KEYSORT LABEL-file name [,DATE-{mmddyy}] [,RETAIN-{J}]
                           [      {ddmmyy}]  [       {T}]
                           [      {yymmdd}]  [        ]

            [,CHECKDUP-{NO }]
            [          {YES}]

// END
```

S9020380-0

Example

Sort the index keys in a resident (T) file named PAYRCD.

```
// LOAD $DDST
// RUN
// KEYSORT LABEL-PAYRCD,RETAIN-T
// END
```

$DELET Utility

The $DELET utility program allows you to remove files, libraries, or folders from disk or diskette.

See the "DELETE Procedure" on page 4-144 for more information.

To delete one or more files from diskette:

```
// LOAD $DELET
// RUN

// {SCRATCH} LABEL-{file name},UNIT-I1[,DATE-{mmddyy}][,LOCATION-{S1 }]
   {REMOVE }      {ALL      }      [      {ddmmyy}]  [        {S2 }]
                                   [      {yymmdd}]  [        {S3 }]
                                                     [        {M1.nn}]
                                                     [        {M2.nn}]

                [,ENDLOC-{S1 }][,PACK-volume id][,DATA-{NO }]
                         {S2 }                       {YES}
                         {S3 }
                         {M1.nn}
                         {M2.nn}

// END
```

S9020381-0

To delete a file, library, folder, or a group of files from disk:

```
// LOAD $DELET
// RUN

// {SCRATCH} LABEL-{file name   },UNIT-F1[,DATE-{mmddyy}][,USERLIBS-{NO }]
   {REMOVE }      {library name}             {ddmmyy}               {YES}
                  {folder name }             {yymmdd}
                  {ALL         }

              [,FOLDER-{NO }][,GROUP-{file group}][,DATA-{NO }]
                       {YES}       {ALL       }       {YES}

// END
```

S9020382-0

The SCRATCH and REMOVE utility control statements can be repeated to delete several files, libraries, or folders with only one load and run of the $DELET utility program.

Differences from the Procedure Parameters

DATA specifies whether the data in the file, library, or folder is to be erased. The DATA parameter is allowed only if the REMOVE utility control statement is used. DATA-YES does the same function as the ERASE parameter in the DELETE procedure.

USERLIBS specifies whether one or more library is to be deleted. YES specifies that libraries are to be deleted. Files will also be deleted if FOLDER-YES or LABEL-ALL is specified. NO specifies that libraries are not to be deleted.

FOLDER specifies whether one or more folder is to be deleted. YES specifies that folders are to be deleted. Files will also be deleted if USERLIBS-YES or LABEL-ALL is specified. NO specifies that folders are not to be deleted.

GROUP-ALL specifies that all files on disk are to be deleted if UNIT-F1 and LABEL-ALL were specified. If LABEL-ALL, UNIT-F1, USERLIBS-YES, FOLDER-YES, and GROUP-ALL are specified, all files, libraries, and folders on disk are deleted. If LABEL-ALL and UNIT-F1 are specified, and GROUP-ALL is not, all files that are not part of a file group are to be deleted.

If password security is active, ALL can be specified only by an operator who has a security classification of system operator or higher. If password security is not active, ALL can be specified only at the system console.

Example 1

To delete the file named PAYROLL from the diskette in slot S1:

```
// LOAD $DELET
// RUN
// REMOVE LABEL-PAYROLL,UNIT-I1
// END
```

Example 2

To delete three files and two libraries with one load and run of the $DELET utility program:

```
// LOAD $DELET
// RUN
// REMOVE LABEL-FILE1,UNIT-F1
// REMOVE LABEL-FILE2,UNIT-F1
// REMOVE LABEL-FILE3,UNIT-F1
// REMOVE LABEL-LIBRARY1,UNIT-F1,USERLIBS-YES
// REMOVE LABEL-LIBRARY2,UNIT-F1,USERLIBS-YES
// END
```

Example 3

To delete all files, libraries, and folders from disk:

```
// LOAD $DELET
// RUN
// REMOVE LABEL-ALL,UNIT-F1,USERLIBS-YES,FOLDER-YES,GROUP-ALL
// END
```

System libraries and files will not be deleted; however, any program product libraries will be deleted.

$DPGP Utility

The $DPGP utility program allows you to print a graphics file on an intelligent printer data stream (IPDS) printer.

See the "PRTGRAPH Procedure" on page 4-359 for more information.

```
// LOAD $DPGP
// LOCAL OFFSET-1,DATA-'               ',AREA-SYSTEM
// LOCAL OFFSET-1,DATA-'file name',AREA-SYSTEM
// LOCAL OFFSET-10,DATA-'width',AREA-SYSTEM
// RUN
```

S9020566-0

Example

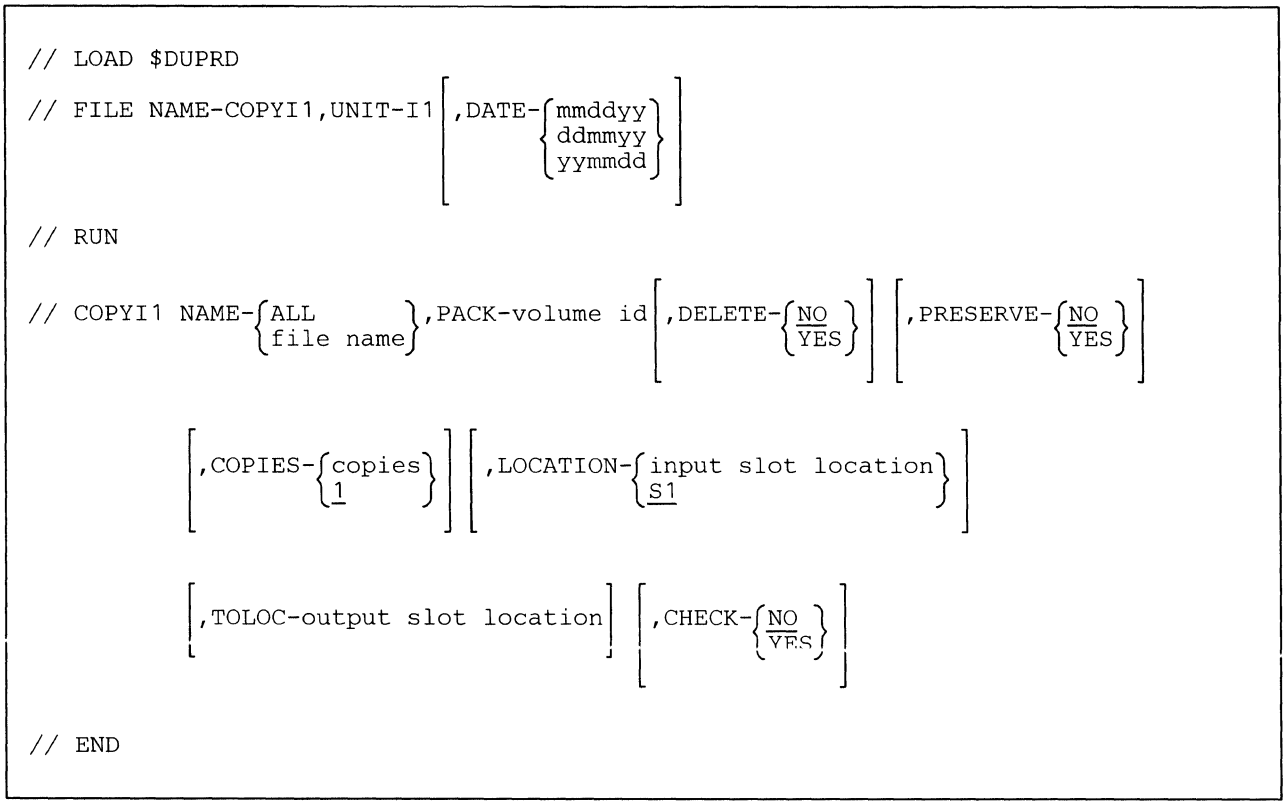
To print a graphics file named FILE1, with a width of 13, on an IPDS printer:

```
// LOAD $DPGP
// LOCAL OFFSET-1,DATA-'               ',AREA-SYSTEM
// LOCAL OFFSET-1,DATA-'FILE1',AREA-SYSTEM
// LOCAL OFFSET-10,DATA-'13',AREA-SYSTEM
// RUN
```

$DUPRD Utility

The $DUPRD utility program allows you to copy files or libraries on diskette(s) to another diskette(s).

See the "COPYI1 Procedure" on page 4-121 for more information.



S9020383-0

Example

Copy a diskette file named PAYROLL onto a diskette with a volume ID of VOL001:

```
// LOAD $DUPRD
// FILE NAME-COPYI1,UNIT-I1
// RUN
// COPYI1 NAME-PAYROLL,PACK-VOL001,LOCATION-S1
// END
```

$FBLD Utility

The $FBLD utility program allows you to:

- Create a new, empty disk file (BLDFILE procedure)
- Create an alternative index for a physical file (BLDINDEX procedure)

Creating Files (BLDFILE Procedure)

See the "BLDFILE Procedure" on page 4-56 for more information.

```
// LOAD $FBLD
// RUN
// FILE LABEL-file name,ATTRIB-{SEQUENTL}  ,{BLOCKS-size  }  ,RECL-record length
                              {(S)       }  {B-size       }
                              {INDEXED   }  {RECORDS-size }
                              {(I)       }  {R-size       }
                              {DIRECT    }
                              {(D)       }

      [,LOCATION-{A1          }]  [,RETAIN-{T}]  [,DFILE-{NO }]
                 {A2          }              {S}         {YES}
                 {A3          }              {J}
                 {A4          }
                 {block number}

      [,POSITION-key position,LENGTH-key length]

      [,DUPKEY-{NO }]  [,EXTEND-{extend value}]
               {YES}            {0           }

// END
```

S9020384-0

The FILE utility control statement can be repeated to build several files with only one load and run of the $FBLD utility program.

Example

Create a resident, delete-capable file that is 13 blocks long. The file is INVOICE, the record length is 50 bytes, the preferred file placement is on the first disk, and each record contains a 4-byte key beginning at position 9 in the record.

```
// LOAD $FBLD
// RUN
// FILE LABEL-INVOICE,ATTRIB-I,BLOCKS-13,RECL-50,
//      LOCATION-A1,RETAIN-T,DFILE-YES,
//      POSITION-9,LENGTH-4
// END
```

Creating Alternative Indexes (BLDINDEX Procedure)

See the "BLDINDEX Procedure" on page 4-59 for more information.

```
// LOAD $FBLD
// RUN
// FILE LABEL-alternative index file name,ATTRIB-{ALTINDEX},
                                                 {(X)      }

        PLABEL-physical file name [,PDATE-{mmddyy}] ,{POSITION-key position},
                                          {ddmmyy}   {POSITIN1-key position}
                                          {yymmdd}

        {LENGTH-key length } ,[POSITIN2-key position,LENGTH2-key length]
        {LENGTH1-key length}

        [,POSITIN3-key position,LENGTH3-key length]

        [,LOCATION-{A1             }] [,DUPKEY-{YES}]
                   {A2             }          {NO }
                   {A3             }
                   {A4             }
                   {block location }]

// END
```

S9020385-1

Example

This example shows how to create an alternative index file from a physical indexed file named CUSTOMER. The CUSTOMER file is keyed on a customer number in positions 1 through 4 of the record. The alternative index file is to be keyed on the customer's name, which is in positions 5 through 20 of the record (a key length of 16), is to be named CUSTNAME, and is to have a preferred location of A2.

```
// LOAD $FBLD
// RUN
// FILE LABEL-CUSTNAME,ATTRIB-ALTINDEX,PLABEL-CUSTOMER,
//      POSITION-5,LENGTH-16,DUPKEY-YES,LOCATION-A2
// END
```

$FREE Utility

The $FREE utility program allows you to gather all free space within the user area on disk.

See the "COMPRESS Procedure" on page 4-105 for more information.

```
// LOAD $FREE
// RUN
[
[                                                          ]
// COMPRESS [DISK-{ALL}]  [,FREE-{LOW }]
[            {A1 }     ]  [      {HIGH}]
[            {A2 }     ]
[            {A3 }     ]
[            {A4 }     ]
[                      ]
[                                                          ]
]
// END
```

S9020386-0

Example

To compress the free space on disk:

```
// LOAD $FREE
// RUN
// END
```

$HELP Utility

See the "HELP Procedure" on page 4-199 for more information. You cannot run $HELP by entering LOAD and RUN OCL statements from the keyboard.

$HIST Utility

The $HIST utility program allows you to display, print, copy, or erase entries in the history file.

See the "HISTORY Procedure" on page 4-209 for more information.

```
// LOAD $HIST
// RUN

// DISPLAY {CRT     }[,[USER      ][,[ALLWS       ][,[ALLENTS        ],
           {LIST    }               [ALL       ]    [display id ]    [EONLY     ]
           {COPYSYS }               [user id   ]    [           ]    [procedure ]
           {COPYPRT }               [          ]    [           ]    [name      ]
           {ERASE   }

           [ALLDAYS ][,[from time][,[to time][,[file name     ][,[NOERASE]
           [TODAY   ]  [000000   ]  [235959 ]  [HF display id]  [ERASE  ]
           [date    ]

// END
```

S9020387-0

Example

To list the history file entries associated with the running of a procedure named TEST:

```
// LOAD $HIST
// RUN
// DISPLAY LIST,,,TEST
// END
```

$IDSET Utility

The $IDSET utility program allows you to display, update, or delete a list of remote IDs to be used by the SSP-ICF BSCEL subsystem on a switched communications line.

See the "DEFINEID Procedure" on page 4-139 for more information.

```
// LOAD $IDSET
// RUN
// DEFINEID  MODE-{DISPLAY}
                  {UPDATE }
                  {DELETE }

// END
```

S9020388-0

Example

To create or update the remote ID list:

```
// LOAD $IDSET
// RUN
// DEFINEID MODE-UPDATE
// END
```

$IEDS Utility

The $IEDS utility program allows you to disable an SSP-ICF subsystem, MSRJE, 3270 device emulation, or PC Support/36.

See the "DISABLE Procedure" on page 4-157 for more information.

```
// LOAD $IEDS
// RUN

// DISABLE SUBSYS-configuration name[,LOCATION-location name] [,LINE-line number]

// END
```

S9020389-1

Example

Disable the subsystem configuration SUB1.

```
// LOAD $IEDS
// RUN
// DISABLE SUBSYS-SUB1
// END
```

$IENBL Utility

The $IENBL utility program allows you to enable an SSP-ICF subsystem, MSRJE, 3270 device emulation, or PC Support/36.

See the "ENABLE Procedure" on page 4-170 for more information.

```
// LOAD $IENBL
// RUN
// ENABLE SUBSYS-configuration name[,LIBRNAME-{library name          }]
                                             {current session library}

          [,LINE-{1 }]  [,SHOW-{NO }]  [,LOCATION-location name]
                 {2 }          {YES}
                 {3 }
                 {4 }
                 {5 }
                 {6 }
                 {7 }
                 {8 }
                 {9 }
                 {10}

          [,LINEMEM-line member name]
// END
```

S9020390-4

Example

In the following example, the $IENBL utility enables the subsystem specified by the configuration member SUB1 found in the library named MYLIB. SUB1 is to use communications line 1, and the parameters for this configuration are to be displayed.

```
// LOAD $IENBL
// RUN
// ENABLE SUBSYS-SUB1,LIBRNAME-MYLIB,LINE-1,SHOW-YES
// END
```

$INIT Utility

The $INIT utility program allows you to initialize one or more diskettes.

See the "INIT Procedure" on page 4-233 for more information.

```
// LOAD $INIT
// RUN

// UIN OPTION-{RENAME }  [,LOCATION-{S1  }]  [,ENDLOC-{S1  }]
              {FORMAT  }             {S2  }          {S2  }
              {FORMAT2 }             {S3  }          {S3  }
              {DELETE  }             {M1  }          {M1  }
              {DIAG    }             {M2  }          {M2  }
                                     {M1.nn}         {M1.nn}
                                     {M2.nn}         {M2.nn}

      [,ACTFLMSG-{YES}]  [,RECL-{record length}]
                 {NO }          {80           }

// VOL PACK-volume id,ID-owner id
// END
```

S9020391-1

Differences from the Procedure Parameters

ACTFLMSG specifies whether a check for active files should be made. The INIT procedure always checks for active files. YES specifies that active files should be checked for by $INIT. If any files on the diskette are active, a message should be displayed. NO specifies that active files should not be checked for by $INIT.

RECL-record length specifies the record length to assign each diskette sector. If no value is specified, 80 is assumed. This data is not used by the System/36; information is placed in the diskette sectors regardless of the record length specified for the sectors.

Example

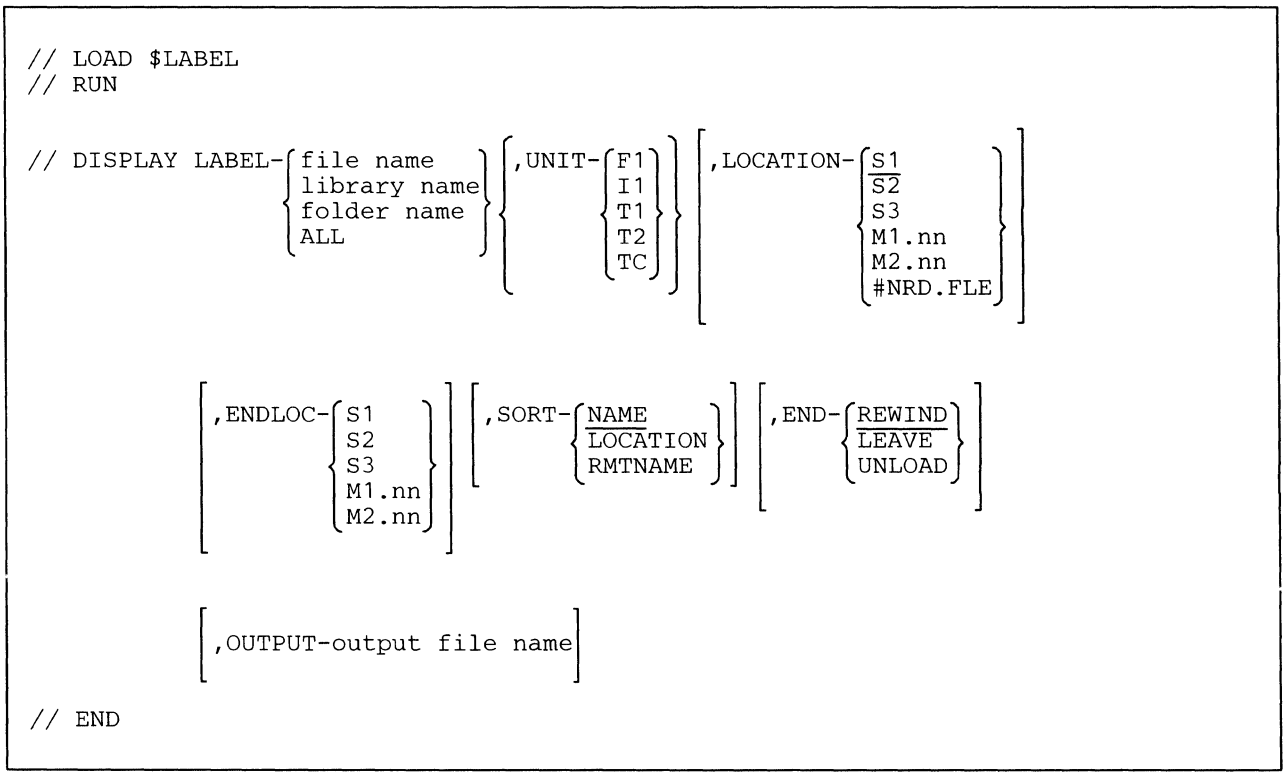
Rename a diskette so that the new volume ID is VOL001 and the new owner ID is YOURNAME. The diskette is in diskette slot S1.

```
// LOAD $INIT
// RUN
// UIN OPTION-RENAME
// VOL PACK-VOL001,ID-YOURNAME
// END
```

$LABEL Utility

The $LABEL utility program allows you to list the names of files, libraries, and folders on disk, diskette, or tape. You can also list one or all entries in the system network directory (NRD).

See the "CATALOG Procedure" on page 4-73 and the "LISTNRD Procedure" on page 4-292 for more information.



S9020392-2

LOCATION - #NRD.FLE is valid only with UNIT-F1. If this parameter is specified, only the entries in the network resource directory (NRD) will be listed.

SORT - RMTNAME is valid only with LOCATION-#NRD.FLE.

OUTPUT - output file name is valid only with UNIT-F1, LOCATION-not specified.

Example 1

Display the contents of a diskette in slot S1:

```
// LOAD $LABEL
// RUN
// DISPLAY LABEL-ALL,UNIT-I1
// END
```

Example 2

Display the contents of the system network directory (NRD) sorted by remote name:

```
// LOAD $LABEL
// RUN
// DISPLAY LABEL-ALL,UNIT-F1,LOCATION-#NRD.FLE,SORT-RMTNAME
// END
```

Example 3

List the names of all disk files, libraries, and folders to a disk file called OUTFILE.

```
// LOAD $LABEL
// RUN
// DISPLAY LABEL-ALL,UNIT-F1,OUTPUT-OUTFILE
// END
```

$MAINT Utility

The $MAINT utility program allows you to:

- Create a library on disk (BLDLIBR procedure)
- Create source or procedure members
- Change the size of a library or directory or change the location of a library (ALOCLIBR procedure)
- Change information about a library member (CHNGEMEM procedure)
- Gather unused library space (CONDENSE procedure)
- Copy a member from one library to another (LIBRLIBR procedure)
- Copy a member from a library to disk, diskette, tape, or tape cartridge (FROMLIBR procedure)
- Copy members from disk, diskette, tape, or tape cartridge to a library (TOLIBR procedure)
- Start a job from disk, diskette, tape, or tape cartridge (JOBSTR procedure)
- List library members and information (LISTLIBR procedure)
- List information about libraries on disk, diskette, tape, or tape cartridge (LISTFILE procedure)
- Remove members from a library (REMOVE procedure)
- Save a library on diskette, tape, or tape cartridge (SAVELIBR procedure)
- Restore a library from diskette, tape, or tape cartridge (RESTLIBR procedure)

Also shown are $MAINT utility control statements that are supported for compatibility with the IBM System/34, and the format of the COPY and CEND statements used by the $MAINT utility program.

Create a Library (BLDLIBR Procedure)

See the "BLDLIBR Procedure" on page 4-62 for more information.

```
// LOAD $MAINT

[// FILE NAME-file name,UNIT-{I1}]
                            {F1}
                            {T1}
                            {T2}
                            {TC}

// RUN
// ALLOCATE LIBRNAME-library name,LIBRSIZE-library size,STATUS-CREATE

          [,DIRSIZE-directory size] [,LOCATION-{A1          }]
                                               {A2          }
                                               {A3          }
                                               {A4          }
                                               {block number}

[// COPY FROM-{DISK},TO-library name,FILE-file name]
            {TAPE}

// END
```

S9020393-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43, the "FILE OCL Statement (for Disk Files)" on page 5-32, or the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

STATUS-CREATE specifies that the library is being created.

If the file containing members to be copied is a record-mode file, each member in the file must begin with a COPY statement and end with a CEND statement. For the formats of the COPY and CEND statements, see "COPY and CEND Statements" on page A-69. The COPY and CEND statements are automatically inserted into members created by $MAINT. **You** must insert them at the beginning and end of members that are not created by $MAINT. Do not insert more than the one required CEND statement, however. If a CEND statement exists within the member, an error message will be issued.

If the record-mode file is organized as a direct file, you must insert an END statement following the CEND statement that ends the last member in the file. The format of the END statement is:

```
// END
```

where only one blank must separate the // and the END.

Example

Create a new library called MYLIB. It should have a size of 100 blocks and a directory size of 20 sectors.

```
// LOAD $MAINT
// RUN
// ALLOCATE STATUS-CREATE,LIBRNAME-MYLIB,LIBRSIZE-100,
//          DIRSIZE-20
// END
```

Create Source or Procedure Members

The $MAINT utility program allows you to create source and procedure members.

The easiest way to create library source or procedure members is to use a program such as SEU (source entry utility), which allows you to enter and change library source and procedure members.

If you do not have SEU, you can use the $MAINT utility program to create source or procedure members entered from the keyboard. (Note that you can only create members using $MAINT, you cannot change a statement in a library member using $MAINT.)

The format of the statements to create a source or procedure member is:

```
// LOAD $MAINT
// RUN
// COPY FROM-READER,LIBRARY-{P},NAME-member name,TO-{library name}
                            {S}                  {F1          }

      [,RETAIN-{P}] [,RECL-{statement length}] [,MRT-YES] [,PDATA-{NO }]
               {R}         {120              }                {YES}

      [,HIST-{YES}] [,REF-number] [,SUB-subtype]
             {NO }

statements for library member

// CEND
// END
```

S9020394-0

FROM-READER specifies that the information for the $MAINT program will be supplied from SYSIN. The OCL statements may be entered from the keyboard or included in a procedure.

LIBRARY-P specifies that the statements being entered are to be placed in a library procedure member.

LIBRARY-S specifies that the statements being entered are to be placed in a library source member.

NAME-member name specifies the name to assign the library member. A member name can be up to 8 characters long and must begin with an alphabetic character (A through Z, #, $, or @). Periods (.) cannot be used within a member name The remaining characters can be any combination of characters (numeric, alphabetic, and special). You should avoid using the following characters because these have special meanings in procedures: commas (,), apostrophes ('), blanks, question marks (?), slashes (/), greater than signs (>), plus signs (+), minus signs (-), and equal signs (=). Do not use DIR, LIBRARY, or ALL as a member name.

TO-library name specifies the name of the library that is to contain the member.

F1 specifies the system library (#LIBRARY).

RETAIN-P specifies that if a library member with the same name as **member name** exists in the library, a message should be displayed. The operator can then take an option to replace the existing member or cancel the $MAINT program. If the RETAIN parameter is not specified, RETAIN-P is assumed.

RETAIN-R specifies that if a library member with the same name as **member name** exists in the library, the existing member should be replaced.

RECL-statement length specifies the length of the statements, in characters, for the source or procedure member, and can be any decimal number from 40 to 120. If the RECL parameter is not specified, 120 is assumed.

MRT-YES specifies that the procedure member is a multiple requester terminal (MRT) procedure. If MRT-YES is not specified, the member is not an MRT procedure.

PDATA-YES specifies that data (not parameters) can be passed on the procedure command that causes the procedure to be run. The data starts with the first nonblank character following the procedure name and ends with the last nonblank character in the statement. The data is passed on the first input operation from the requester display station from the first program in the procedure. Every MRT procedure has this attribute whether PDATA is specified.

Data passed on the INCLUDE OCL statement can contain IGC characters; however, parameters passed on the INCLUDE OCL statement cannot contain IGC characters.

PDATA-NO specifies that if the procedure is not an MRT procedure, parameters are passed to the procedure.

HIST-YES specifies that the OCL statements contained in the procedure are to be logged in the history file. If the HIST parameter is not specified, HIST-YES is assumed.

HIST-NO specifies that the OCL statements contained in the procedure are not to be logged in the history file.

REF-number specifies the reference number for the member. The number can be up to 6 digits long. For example, 000042 is the 42nd revision of a member.

SUB-subtype specifies the subtype for the members. Valid subtypes are:

ARP RPG auto report member
ARS Automatic response member
ASM Assembler member
BAP BASIC procedure (source member)
BAS BASIC member
BGC Business graphics chart
BGD Business graphics data
BGF Business graphics format
COB COBOL member
DFU Data file utility member
DTA Data member
FMT Display format member
FOR FORTRAN member
MNU Menu member
MSG Message member
PHL Phone list member
RPG RPG member
SRT Sort member
UNS Unspecified
WSU Work station utility member

statements for library member specifies the statements for the library member being created. You must enter at least one statement between the COPY and CEND statements.

Example

This example shows how to copy the following procedure into a procedure member named PROC1. The name of the library that is to contain the procedure is MYLIB. The statements in the procedure are as follows:

```
// LOAD PROG1
// FILE NAME-FILE1
// RUN
```

The procedure runs a program named PROG1. The procedure is not an MRT procedure.

The statements needed to create the procedure are as follows.

```
// LOAD $MAINT
// RUN
// COPY FROM-READER,LIBRARY-P,NAME-PROC1,TO-MYLIB,RETAIN-R
// LOAD PROG1
// FILE NAME-FILE1
// RUN
// CEND
// END
```

Change Library or Directory Size (ALOCLIBR Procedure)

See the "ALOCLIBR Procedure" on page 4-9 for more information.

```
// LOAD $MAINT
// RUN

// ALLOCATE LIBRNAME-library name,STATUS-{REALOC} [,LIBRSIZE-library size]
                                         {INCR  }
                                         {DECR  }

          [,DIRSIZE-directory size] [,LOCATION-{A1          }]
                                               {A2          }
                                               {A3          }
                                               {A4          }
                                               {block number}

// END
```

S9020395-0

Differences from the Procedure Parameters

STATUS-REALOC specifies that an existing library is to be changed (reallocated). The sizes specified for LIBRSIZE and DIRSIZE are to be the new sizes.

Example 1

This example shows a library named MYLIB being changed to 250 blocks. The size of the directory is being changed to 25 sectors.

```
// LOAD $MAINT
// RUN
// ALLOCATE LIBRNAME-MYLIB,STATUS-REALOC,LIBRSIZE-250,
//          DIRSIZE-25
// END
```

Example 2

This example shows the size of a library named MYLIB being increased by 50 blocks. The size of the directory is also being increased by 10 sectors.

```
// LOAD $MAINT
// RUN
// ALLOCATE LIBRNAME-MYLIB,STATUS-INCR,LIBRSIZE-50,
//          DIRSIZE-10
// END
```

The following statements can also be used to change the size of a library, and are supported for compatibility with the IBM System/34.

```
// LOAD $MAINT
// RUN

// ALLOCATE [STATUS-CHANGE],[INCREASE-blocks ],[DIRSIZE-sectors]
                            [DECREASE-blocks]

            [,LIBRNAME-{library name}]
                       {#LIBRARY   }

// END
```

S9020396-0

STATUS-CHANGE specifies a change in the size of a library or in the size of a library directory is to be made. If no STATUS parameter is specified, STATUS-CHANGE is assumed.

INCREASE specifies the number of blocks by which the library size is to be increased. INCREASE cannot be specified with DIRSIZE.

DECREASE specifies the number of blocks by which the library size is to be decreased. DECREASE cannot be specified with DIRSIZE.

DIRSIZE specifies the number of sectors for the library directory. DIRSIZE cannot be specified with INCREASE or DECREASE.

LIBRNAME specifies the name of the library whose size is to be changed. If no parameter is specified, the system library (#LIBRARY) is assumed.

Example

To decrease the number of blocks allocated to a library named MYLIB by 20.

```
// LOAD $MAINT
// RUN
// ALLOCATE DECREASE-20,LIBRNAME-MYLIB
// END
```

Change Library Member Information (CHNGEMEM Procedure)

See the "CHNGEMEM Procedure" on page 4-94 for more information.

```
// LOAD $MAINT
// RUN

// CHANGE NAME-{member name     },LIBRARY-{S  } [,LIBRNAME-{library name}]
               {member name.ALL}           {P  }          {#LIBRARY   }
               {ALL            }           {O  }
                                           {R  }
                                           {ALL}

          [,NEWNAME-new member name] [,SUB-subtype]

          [,REF-reference number]

// END
```

S9020397-0

Example

Change the name of the library procedure member PAYROLL1 to PAYROLL2 in the library named FINANCE:

```
// LOAD $MAINT
// RUN
// CHANGE NAME-PAYROLL1,LIBRARY-P,LIBRNAME-FINANCE,
   NEWNAME-PAYROLL2
// END
```

Gather Unused Library Space (CONDENSE Procedure)

See the "CONDENSE Procedure" on page 4-109 for more information.

```
// LOAD $MAINT
// RUN

// COMPRESS [LIBRNAME-{library name}]
                      {#LIBRARY   }

// END
```

S9020398-0

Example

To condense the system library (#LIBRARY):

```
// LOAD $MAINT
// RUN
// COMPRESS LIBRNAME-#LIBRARY
// END
```

Copy Members from One Library to Another (LIBRLIBR Procedure)

See the "LIBRLIBR Procedure" on page 4-255 for more information.

```
// LOAD $MAINT
// RUN

// COPY FROM-{from library name},LIBRARY-{S   },NAME-{member name     },
             {F1                }          {P   }      {member name.ALL}
                                           {O   }      {ALL            }
                                           {R   }
                                           {ALL }

        TO-{to library name},NEWNAME-new member name [,RETAIN-{P}]
           {F1             }                         [        {R}]

        [,OMIT-{member name     }]
               {member name.ALL}
               {SYSTEM          }

// END
```

S9020399-0

Differences from the Procedure Parameters

NAME-ALL specifies that all members, except IBM-supplied members, are to be copied or listed when LIBRARY-ALL is specified, or that all members of the type specified by the LIBRARY parameter (including IBM-supplied members) are to be copied.

OMIT specifies the name of one or more members to be omitted from the copy:

member name specifies the members that are to be omitted.

member name.ALL specifies that all members whose names begin with the specified characters are to be omitted. Up to 7 characters can be specified.

SYSTEM specifies that all IBM-supplied library members are to be omitted.

The LIBRLIBR procedure always omits IBM-supplied library members. If OMIT-SYSTEM is not specified, both IBM and non-IBM members are copied.

Example 1

Copy all the library members named TEST from a library named MYLIB into a library named YOURLIB, and replace any members named TEST in YOURLIB.

```
// LOAD $MAINT
// RUN
// COPY FROM-MYLIB,TO-YOURLIB,NAME-TEST,LIBRARY-ALL,
//      RETAIN-R
// END
```

Example 2

Copy all the procedures beginning with PAY from a library named MYLIB to a library named YOURLIB. However, those library members beginning with PAYR are not to be copied.

```
// LOAD $MAINT
// RUN
// COPY FROM-MYLIB,TO-YOURLIB,NAME-PAY.ALL,LIBRARY-P,
//      RETAIN-R,OMIT-PAYR.ALL
// END
```

Copy Members from a Library (FROMLIBR Procedure)

See the "FROMLIBR Procedure" on page 4-193 for more information.

These statements allow you to do more tasks than the FROMLIBR procedure allows. You can, for example:

- Create a basic data exchange diskette file that contains one or more library members.
- Omit one or more members from the copy operation.

To Create a Sector Mode File

Sector mode files are created by specifying the TO-DISK parameter without the RECL parameter. A sector mode copy can be specified for any type of library member (source, procedure, load, or subroutine). In sector mode, copies are in system format and consist of control information and PTF (program temporary fix) numbers for any PTFs that have been applied to a member, followed by the member as it exists in the library. The record length of a sector mode file is 8.

```
// LOAD $MAINT

// FILE NAME-{file name   },UNIT-{I1},PACK-volume id
             {member name}       {F1}
                                 {T1},VOLID-volume id
                                 {T2},VOLID-volume id
                                 {TC}

// RUN
// COPY FROM-{library name},TO-{DISK},FILE-{file name   } ,
             {F1           }     {TAPE}     {member name}

        NAME-{member name     },LIBRARY-{S  } [,ADD-{NO }]
             {member name.ALL}            {P  }    {YES}
             {ALL            }            {R  }
                                          {O  }
                                          {ALL}

        [,PTF-{NO }] [,OMIT-{SYSTEM     }]
              {YES}        {member name  }
                           {member name.ALL}

// END
```

S9020400-1

Differences from the Procedure Parameters

If T1, T2, or TC is specified for the UNIT parameter in the FILE statement, then RECFM-FB must be specified.

PTF specifies whether library members that have program temporary fixes (PTFs) should be copied. The PTF parameter is only allowed for sector mode files.

NO specifies that PTFs have no particular significance. If PTF is not specified, PTF-NO is assumed.

YES specifies that only members that have PTFs applied are to be copied.

OMIT specifies the name of one or more library members to be omitted from the copy function.

member name specifies that members having the specified name are to be omitted from the copy function.

member name.ALL specifies that all members whose names begin with the specified characters are to be omitted from the copy function. Up to 7 characters can be specified.

SYSTEM indicates that all IBM-supplied members are to be omitted.

To Create a Record Mode File

The RECL parameter indicates that the copy of a file is in record mode, not sector mode. Record mode can be specified only for source and procedure members. Source and procedure member copies made in record mode are preceded by a COPY statement and followed by a CEND statement.

```
// LOAD $MAINT

// FILE NAME-{file name   },UNIT-{I1},PACK-volume id
             {member name}       {F1}
                                 {T1},VOLID-volume id
                                 {T2},VOLID-volume id
                                 {TC}

// RUN
// COPY FROM-{library name},TO-{DISK},FILE-{file name   },
             {F1           }    {TAPE}      {member name}


      NAME-{member name     },LIBRARY-{S  },{RECL-record length}
           {member name.ALL}          {P  }{ADD-YES           }
           {ALL            }          {ALL}

      [,OMIT-{SYSTEM     }] [,BASIC-{NO }] [,SVATTR-{NO }]
             {member name}          {YES}          {YES}
             {member name.ALL}

// END
```

S9020401-1

For more information about the FILE statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, or the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

If T1, T2, or TC is specified for the UNIT parameter in the FILE statement, then RECFM-FB must be specified.

TO-DISK specifies that the library members are to be copied to a disk or diskette file.

TO-TAPE specifies that the library members are to be copied to a tape file.

LIBRARY-ALL specifies that all source (S), procedure (P), subroutine (R), and load (O) members are to be copied when a sector more file is being created. When a record mode file is being created, ALL specifies that all source (S) and procedure (P) members are to be copied.

ADD specifies whether one or more library members are to be added to a file that already contains members. Either YES or NO can be specified. If no parameter is specified, ADD-NO is assumed.

When you are adding a member to a disk file, the file must contain enough unused space to hold the member. When you are adding a member to a diskette or tape file, the file must be the last active (unexpired) file on the diskette.

The RECL parameter is ignored if ADD-YES is specified. For record mode files, the record length is determined by the record length of the existing file.

RECL specifies the record length, in bytes, of a source or procedure member. The record length can be from 40 through 120. You must specify a record length of at least 73 bytes if you also specify SVATTR-YES, or an error message will be issued.

BASIC specifies whether the copied output is to be placed in a basic data exchange diskette file.

NO specifies that a basic data exchange diskette file is not to be created. If BASIC is not specified, BASIC-NO is assumed.

YES specifies that a basic data exchange diskette is to be created.

Note: If ADD-YES is specified, BASIC-YES should not be specified. When you add records to a basic exchange file, do not specify BASIC-YES.

Example 1

This example shows how to save a library procedure member named TEST on a diskette having a volume ID of VOL003. The procedure is in the library named MYLIB and the diskette is in magazine position M2.05.

```
// LOAD $MAINT
// FILE NAME-TEST,UNIT-I1,PACK-VOL003,LOCATION-M2.05,AUTO-NO
// RUN
// COPY FROM-MYLIB,TO-DISK,LIBRARY-P,NAME-TEST,FILE-TEST
// END
```

Example 2

This example shows how to create a record mode file from a library source member. The name of the file is to be FILE1 and it is to have a record length of 80. The library member is named TEST and is contained in the library named MYLIB.

```
// LOAD $MAINT
// FILE NAME-FILE1,UNIT-F1,BLOCKS-30
// RUN
// COPY FROM-MYLIB,TO-DISK,LIBRARY-S,NAME-TEST,FILE-FILE1,
//      RECL-80
// END
```

Example 3

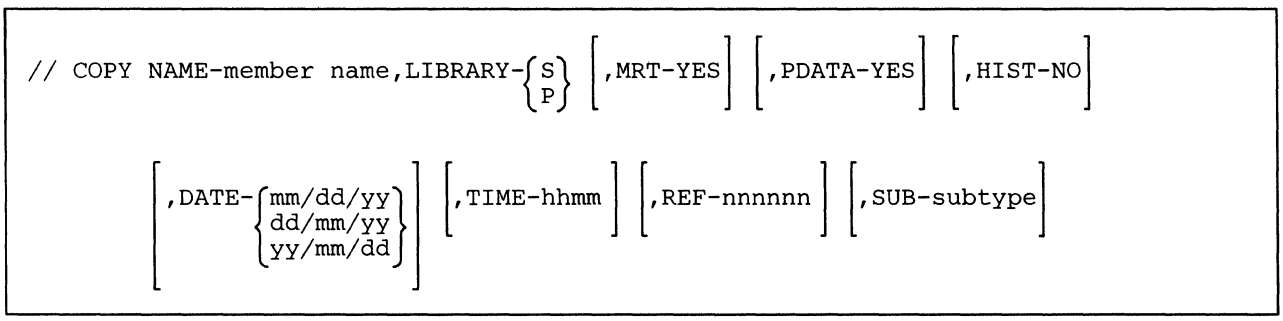
This example shows how to create a basic data exchange diskette record mode file from a library source member. The name of the file is to be FILE2 and it is to have a record length of 80. The library member is named TEST and is in the library named MYLIB. The diskette to be used has a volume ID of VOL001.

```
// LOAD $MAINT
// FILE NAME-FILE2,UNIT-I1,PACK-VOL001
// RUN
// COPY FROM-MYLIB,TO-DISK,LIBRARY-S,NAME-TEST,FILE-FILE2,
//      RECL-80,BASIC-YES
// END
```

COPY and CEND Statements

When copying one or more library members to a record mode file, the $MAINT utility adds a COPY statement before each member and a CEND statement after each member.

The COPY statement has the following format:



S9020409-2

NAME-member name specifies the library member to be placed in the specified library.

LIBRARY-S specifies that the member is a source member.

LIBRARY-P specifies that the member is a procedure member.

MRT-YES specifies that the procedure member is a multiple requester terminal (MRT) procedure. If MRT-YES is not specified, the member is not an MRT procedure.

PDATA-YES specifies that data (not parameters) can be passed on the procedure command that causes the procedure to be run. The data starts with the first nonblank character following the procedure name and ends with the last nonblank character in the statement. The data is passed on the first input operation from the requester display station from the first program in the procedure. Every MRT procedure has this indicator whether PDATA-YES is specified.

If PDATA-YES is not specified and if the procedure is not an MRT procedure, parameters are passed to the procedure.

HIST-NO specifies that the OCL statements contained in the procedure are not to be logged in the history file. If HIST-NO is not specified, the OCL statements contained in the procedure are logged in the history file.

DATE specifies the date that the member was created or last changed. The date must be specified in the same format as the session date.

TIME specifies the time that the member was created or last changed. The time is in the form: hhmm (hours, minutes).

REF specifies the reference number of the member. The number is 6 digits long and is right-justified (for example, 000042 is the 42nd revision of a member).

SUB specifies the subtype of the library member. The following subtypes are indicated:

ARP Auto report member
ARS Automatic response member
ASM Assembler member
BAP BASIC procedure member
BAS BASIC member
BGC Business graphics chart
BGD Business graphics data
BGF Business graphics format
COB COBOL member
CSM Communications and Systems Management member
CSP Cross-system product member
DFU Data file utility member
DLS Document library services member
DTA Data member
FMT Display format member
FOR FORTRAN member
ICF Interactive communications feature member
KEY KEYS procedure
MNU Menu member
MSG Message member
PHL Phone or call list member
QDE Query data entry member
QRY Query member
RPG RPG member
SRT Sort member
SSP CNFIGSSP procedure system configuration member
TXT Text member
WSU Work station utility member
UNS Unspecified
X25 X.25 packet switching link control

The CEND statement has the following format:

```
// CEND
```

S9020410-0

Copy Members to a Library (TOLIBR Procedure)

See the "TOLIBR Procedure" on page 4-537 for more information.

```
// LOAD $MAINT
// FILE NAME-file name,UNIT-{F1}
                            {I1}
                            {T1}
                            {T2}
                            {TC}

// RUN
// COPY TO-{library name},FROM-{DISK},FILE-file name [,NAME-{member name     }]
        {F1          }       {TAPE}                    {member name,ALL}

        [,RETAIN-{P}] [,PTF-{NO    }] [,SUB-subtype] [,LIBRARY-{S  }]
                 {R}       {number}                          {P  }
                                                             {O  }
                                                             {R  }
                                                             {ALL}

        [,OMIT-NEW]

// END
```

S9020402-1

Only some of the parameters are shown for the FILE OCL statement. For information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43, the "FILE OCL Statement (for Disk Files)" on page 5-32, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

For the **COPY** statement:

FROM-DISK specifies that the library members are being copied from a diskette or disk file.

FROM-TAPE specifies that the library members are to be copied from a tape file.

RETAIN-P specifies that if a member with the same name already exists in the library, a message is to be displayed. You can then decide whether to replace the duplicate member. If no RETAIN parameter is specified, P is assumed.

RETAIN-R specifies that the duplicate member is to be replaced without a message.

PTF specifies that library members that have program temporary fixes (PTFs) applied should be copied into the library. The PTF parameter can only be specified for sector-mode diskette or disk files.

NO specifies that PTFs have no particular significance. A library member is copied regardless of its PTF status. If no parameter is specified, NO is assumed.

number specifies that only those members that have the specified PTF number are copied. The number can be from 1 through 65535.

OMIT-NEW specifies that new members are not to be copied.

Example

Copy the members from a diskette file called PAY into a library named MYLIB and replace any duplicate members.

```
// LOAD $MAINT
// FILE NAME-PAY,UNIT-I1
// RUN
// COPY FROM-DISK,TO-MYLIB,NAME-PAY,FILE-PAY,RETAIN-R
// END
```

Start a Job (JOBSTR Procedure)

See the "JOBSTR Procedure" on page 4-246 for more information.

```
// LOAD $MAINT
// FILE NAME-file name,UNIT-{I1}
// RUN                      {T1}
                            {T2}
                            {TC}

// COPY FROM-{DISK},FILE-file name,TO-{library name}
             {TAPE}                   {F1          }

// END

{procedure name[,library name]                                  }
{                                                               }
{// JOBQ [jobq prty,] [library name],procedure name             }
```

S9020403-1

Example

Copy the procedure members (PROC1 and PROC2) and the source member (MEMBER1) into the library named MYLIB from a diskette file named JOBS, and then run procedure PROC1. See the JOBSTR procedure for the format of the file named JOBS.

```
// LOAD $MAINT
// FILE NAME-JOBS,UNIT-I1
// RUN
// COPY FROM-DISK,FILE-JOBS,TO-MYLIB
// END
PROC1,MYLIB
```

List Library Members and Information (LISTLIBR Procedure)

See the "LISTLIBR Procedure" on page 4-272 for more information.

```
// LOAD $MAINT
// RUN
// COPY TO-PRINT,NAME-{DIR              },LIBRARY-{S     } [,DISPLAY-{MEMBERS}],
                      {member name      }          {P     }         {DIRINFO}
                      {member name.ALL  }          {R     }
                      {ALL              }          {O     }
                                                   {ALL   }
                                                   {SYSTEM}

        FROM-{library name} [,OMIT-{member name     }]
             {F1           }       {member name.ALL}
                                   {SYSTEM         }

        [,LIST-{USER  }] [,PAGE-{NO }] [,SUB-subtype] [,PRTFILE-output file name]
               {DETAIL}         {YES}

// END
```

S9020404-1

The COPY statement can be repeated to list several members or libraries with only one load and run of the $MAINT utility program.

Differences from the Procedure Parameters

TO-PRINT specifies that the output is to go to the system list device.

OMIT specifies one or more library members to be omitted from the copy or display:

member name specifies the members that are to be omitted.

member name.ALL specifies that all members whose names begin with the specified characters are to be omitted. Up to 7 characters can be specified.

SYSTEM specifies that all IBM-supplied library members are to be omitted. The LISTLIBR procedure always omits IBM-supplied library members.

Example 1

List all library procedure members that have names beginning with PAY in the library named PAYLIB. However, do not list the members beginning with PAYR.

```
// LOAD $MAINT
// RUN
// COPY FROM-PAYLIB,TO-PRINT,NAME-PAY.ALL,LIBRARY-P,
//      OMIT-PAYR.ALL
// END
```

Example 2

To list all the system library procedure members.

```
// LOAD $MAINT
// RUN
// COPY FROM-#LIBRARY,TO-PRINT,NAME-ALL,LIBRARY-P
// END
```

List Information about Libraries (LISTFILE Procedure)

See the "LISTFILE Procedure" on page 4-267 for more information.

```
// LOAD $MAINT
// FILE NAME-file name,UNIT-{I1 | F1 | T1 | T2 | TC}
// RUN
// COPY TO-PRINT,FROM-{DISK | TAPE},FILE-file name [,LIST-{USER | DETAIL}]
// END
```

S9020405-1

Only some of the parameters are shown for the FILE OCL statement. For information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43, the "FILE OCL Statement (for Disk Files)" on page 5-32, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

If T1, T2, or TC is specified for the UNIT parameter in the FILE statement, then RECFM-FB must be specified. If the file to be listed from T1, T2, or TC is a SAVELIBR file, then RECL-256, BLKL-24576 must be specified. If the file to be listed from T1, T2, or TC is a LIBRFILE, then RECL-256, BLKL-4096 must be specified.

For the **COPY** statement:

TO-PRINT specifies that the output is to go to the system list device.

FROM-DISK specifies a listing from disk or diskette. A FILE OCL statement must define the file.

FROM-TAPE specifies a listing from tape. A FILE OCL statement must define the file.

Remove Members from a Library (REMOVE Procedure)

See the "REMOVE Procedure" on page 4-370 for more information.

```
// LOAD $MAINT
// RUN

// DELETE NAME-{member name     },LIBRARY-{S  } [,LIBRNAME-{library name}]
               {member name.ALL}           {P  }          {#LIBRARY   }
               {ALL            }           {O  }
                                           {R  }
                                           {ALL}

          [,OMIT-{member name     }] [,RETAIN-S]
                 {member name.ALL}

// END
```

S9020406-0

The DELETE utility control statement can be repeated to remove several library members with only one load and run of the $MAINT utility program.

Differences from the Procedure Parameters

LIBRNAME-#LIBRARY specifies that members are to be removed from the system library.

OMIT-member name specifies that the members having the specified name are not to be removed from the library. The member name can be from 1 through 8 characters.

OMIT-member name.ALL specifies that the members beginning with the specified characters are not to be removed from the library. The member name can be from 1 through 7 characters.

RETAIN-S specifies that the IBM-supplied members are to be removed. When you are removing members from the system library (#LIBRARY), if LIBRARY-ALL and NAME-ALL is specified, RETAIN-S is not allowed.

Example

Remove a source member named THIS, a procedure member named THAT, and a load member named OTHER from the library MYLIB.

```
// LOAD $MAINT
// RUN
// DELETE NAME-THIS,LIBRARY-S,LIBRNAME-MYLIB
// DELETE NAME-THAT,LIBRARY-P,LIBRNAME-MYLIB
// DELETE NAME-OTHER,LIBRARY-O,LIBRNAME-MYLIB
// END
```

Save a Library (SAVELIBR Procedure)

See the "SAVELIBR Procedure" on page 4-431 for more information.

```
// LOAD $MAINT

[// FILE NAME-#IPLBOOT,UNIT-I1,PACK-volume id]

// FILE NAME-library name,UNIT-{F1}
                              {I1},PACK-volume id
                              {T1},VOLID-volume id
                              {T2},VOLID-volume id
                              {TC},VOLID-volume id

// RUN
// COPYLIBR FROM-library name,TO-{DISK},FILE-library name
                                 {TAPE}

// END
```

S9020407-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

The #IPLBOOT file is only required when you are saving the system library (#LIBRARY); it must be specified in the first FILE OCL statement.

Example 1

Save a library named MYLIB on a diskette with a volume ID of VOL001. The diskette is in slot S1.

```
// LOAD $MAINT
// FILE NAME-MYLIB,UNIT-I1,PACK-VOL001
// RUN
// COPYLIBR FROM-MYLIB,TO-DISK,FILE-MYLIB
// END
```

Example 2

Save the system library on diskettes with a volume id of SYSTEM. The diskettes are in magazine slot M1, starting in position M1.01.

```
// LOAD $MAINT
// FILE NAME-#IPLBOOT,UNIT-I1,PACK-SYSTEM,LOCATION-M1
// FILE NAME-#LIBRARY,UNIT-I1,PACK-SYSTEM,LOCATION-M1
// RUN
// COPYLIBR FROM-#LIBRARY,TO-DISK,FILE-#LIBRARY
// END
```

Restore a Library (RESTLIBR Procedure)

See the "RESTLIBR Procedure" on page 4-386 for more information.

```
// LOAD $MAINT

[// FILE NAME-#IPLBOOT,UNIT-I1]

// FILE NAME-library name,UNIT-{I1|T1|T2|TC}

// RUN

// COPYLIBR FROM-{DISK|TAPE},TO-library name,FILE-library name

          [,LIBRSIZE-library size] [,DIRSIZE-directory size]

          [,LOCATION-{A1|A2|A3|A4|block number}]

// END
```

S9020408-1

For more information about the FILE statements, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48. The file #IPLBOOT is only needed when the system library is being restored.

Example 1

This example shows a library named MYLIB being restored. The size of the library is being changed to 250 blocks. The size of the directory is not being changed.

```
// LOAD $MAINT
// FILE NAME-MYLIB,UNIT-I1
// RUN
// COPYLIBR FROM-DISK,TO-MYLIB,FILE-MYLIB,LIBRSIZE-250
// END
```

Example 2

This example shows the system library being restored.

```
// LOAD $MAINT
// FILE NAME-#IPLBOOT,UNIT-I1
// FILE NAME-#LIBRARY,UNIT-I1
// RUN
// COPYLIBR FROM-DISK,TO-#LIBRARY,FILE-#LIBRARY
// END
```

$MGBLD Utility

The $MGBLD utility program allows you to create message load members from message source members.

See the "CREATE Procedure" on page 4-132 for more information.

```
// LOAD $MGBLD
// RUN

// MGBLD SOURCE-source member name[,REPLACE-{NO}]
                                  {YES}

        [,LIBRARY-{library name}]  [,HALT-{YES}]  [,SSP-{NO}]
                  {#LIBRARY   }            {NO }        {YES}

// END
```

S9020411-0

Differences from the Procedure Parameters

SSP specifies whether the load member is to be an SSP member. YES specifies that the load member being created is an SSP member and cannot be removed by using the REMOVE procedure. NO specifies that the load member is a user member and can be removed by using the REMOVE procedure. If the SSP parameter is not specified, NO is assumed.

Example

To generate a message load member from a source member named MESSAGES that is contained in the library MYLIB, you would enter the following:

```
// LOAD $MGBLD
// RUN
// MGBLD SOURCE-MESSAGES,LIBRARY-MYLIB
// END
```

$MMSP Utility

The $MMSP utility program allows you to stop the monitoring function of a BSC multipoint line.

See the "STOPM Procedure" on page 4-492 for more information.

```
// LOAD $MMSP
// RUN
// STOPM LINE-line number
// END
```

S9020412-0

Example

This example shows how to stop automonitoring line 1.

```
// LOAD $MMSP
// RUN
// STOPM LINE-1
// END
```

$MMST Utility

The $MMST utility program allows you to start the automatic monitoring function of a BSC multipoint line.

See the "STARTM Procedure" on page 4-489 for more information.

```
// LOAD $MMST
// RUN
// STARTM LINE-line number,CODE-{E},STATION-station address
                                {A}
// END
```

S9020413-0

Example

This example shows how to specify line 1 to be automonitored for an EBCDIC station address of C4.

```
// LOAD $MMST
// RUN
// STARTM LINE-1,CODE-E,STATION-C4
// END
```

$PACK Utility

The $PACK utility program allows you to gather together all unused disk space.

See the "COMPRESS Procedure" on page 4-105 for more information.

```
// LOAD $PACK
// RUN
```

S9020414-0

Example

To compress the free space on disk:

```
// LOAD $PACK
// RUN
```

$PNLM Utility

The $PNLM utility program allows you to create or update phone lists for the autocall feature.

See the "DEFINEPN Procedure" on page 4-140 for more information.

```
// MEMBER PROGRAM1-#PN#M1,PROGRAM2-#PN#M2
// LOAD $PNLM
// RUN
```

S9020415-0

Example

To create a phone list, you would enter:

```
// MEMBER PROGRAM1-#PN#M1,PROGRAM2-#PN#M2
// LOAD $PNLM
// RUN
```

$POST Utility

The $POST utility program allows you to:

- Copy special E-format diskette files or basic data exchange diskette files (POST procedure)
- List basic data exchange diskette files

Copy Special E-Format Diskette Files or Basic Data Exchange Diskette Files (POST Procedure)

See the "POST Procedure" on page 4-345 for more information.

To copy special E-format files or basic data exchange files to or from diskette:

```
// LOAD $POST
// FILE NAME-COPYIN,LABEL-input file name,UNIT-{F1}
                                               {I1}
// FILE NAME-COPYO,LABEL-output file name,UNIT-{F1}
                                               {I1}
// RUN
// TRANSFER [ADD-{NO }] [,KEYLEN-key length,KEYLOC-key location]
            [     {YES}]
            [,EOD-{NO }]
            [     {YES}]
// END
```

S9020416-0

For information about the FILE statements, see the "FILE OCL Statement (for Disk Files)" on page 5-32 and the "FILE OCL Statement (for Diskette Files)" on page 5-43.

Example

This example shows how to create an indexed disk file named FILE2 (with the key in positions 1 through 4) from a special E-format diskette file named FILE2, and check for special end-of-data.

```
// LOAD $POST
// FILE NAME-COPYIN,UNIT-I1,LABEL-FILE2
// FILE NAME-COPYO,UNIT-F1,LABEL-FILE2
// RUN
// TRANSFER ADD-NO,KEYLOC-1,KEYLEN-4,EOD-YES
// END
```

List Basic Data Exchange Diskette Files

To list basic data exchange diskette files:

```
// LOAD $POST
// FILE NAME-COPYIN,LABEL-input file name,UNIT-I1
// RUN

// DISPLAY [FROM-first record [,TO-last record]]
           [,

// END
```

S9020417-0

Only some of the parameters are shown for the FILE OCL statement. For information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43.

The **DISPLAY** utility control statement allows you to list basic data exchange diskette files on the system list device. You can use the STATUS SESSION command to determine the system list device.

FROM-first record specifies the number of the first record to be listed. If no first record value is specified, the file is listed beginning with the first record in the file.

TO-last record specifies the number of the last record to be listed. If no last record value is specified, the file is listed ending with the last record in the file.

Example

This example shows how to list a basic data exchange diskette file named FILE3:

```
// LOAD $POST
// FILE NAME-COPYIN,UNIT-I1,LABEL-FILE3
// RUN
// DISPLAY
// END
```

$PRCED Utility

The $PRCED utility program allows you to add, remove, or update location profiles in the user ID file.

See the "SECEDIT Procedure" on page 4-443 for more information.

```
// LOAD $PRCED
// RUN
```

S9020490-0

Example

Edit location profiles in the user ID file.

```
// LOAD $PRCED
// RUN
```

$PRCLT Utility

The $PRCLT utility program allows you to list location profiles in the user ID file.

See the "SECLIST Procedure" on page 4-445 for more information.

```
// LOAD $PRCLT
// RUN
```

S9020491-0

Example

List location profiles in the user ID file.

```
// LOAD $PRCLT
// RUN
```

$PRLST Utility

The $PRLST utility program allows you to list user profiles in the user ID file.

See the "SECLIST Procedure" on page 4-445 for more information.

```
// LOAD $PRLST
// RUN
// OPTIONS PSWDISP-{PW   },LSTFRMT-{USERID}
                   {NOPW}          {CLASS }
// END
```

S9020418-1

Example

List user profiles in the user ID file.

```
// LOAD $PRLST
// RUN
// OPTIONS PSWDISP-NOPW,LSTFRMT-USERID
// END
```

$PRPWD Utility

The $PRPWD utility program allows you to change your password.

See the "PASSWORD Procedure" on page 4-331 for more information.

```
// LOAD $PRPWD
// RUN
```

S9020514-0

Example

Change your password.

```
// LOAD $PRPWD
// RUN
```

$PRUED Utility

The $PRUED utility program allows you to add, remove, or update user profiles in the user ID file.

See the "SECEDIT Procedure" on page 4-443 for more information.

```
// LOAD $PRUED
// RUN
```

S9020419-0

Example

Edit user profiles in the user ID file.

```
// LOAD $PRUED
// RUN
```

$PRUID Utility

The $PRUID utility program allows you to:

- Create or remove the user ID file
- Activate or deactivate password security
- Activate or deactivate badge security
- Start or stop password date checking

See the "SECDEF Procedure" on page 4-441 for more information.

```
// LOAD $PRUID
// RUN
// OPTIONS CONTROL-{CREATE  }
                   {DELETE  }
                   {ACTPW   }
                   {DEACTPW }
                   {ACTBDG  }
                   {DEACTBDG}
                   {STRTDATE}
                   {STOPDATE}
// END
```

S9020420-1

Example

Create the user ID file.

```
// LOAD $PRUID
// RUN
// OPTIONS CONTROL-CREATE
// END
```

$PRURS Utility

The $PRURS utility program allows you to restore the user ID file.

See the "SECREST Procedure" on page 4-448 for more information.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-file name,UNIT-{F1 }
                                        {I1 }
                                        {T1 }
                                        {T2 }
                                        {TC }
                                                ,PACK-volume id

// FILE NAME-COPYO,LABEL-#SECREST,RETAIN-J
// RUN

// COPYFILE [OUTPUT-S]

// END
// LOAD $PRURS
// FILE NAME-#SECREST,LABEL-#SECREST,RETAIN-S
// RUN

// RESTORE NEWSIZE-{number of records}
                   {0               }

// END
```

S9020421-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

OUTPUT-S is required when the file is being restored from diskette. It should not be used for disk or tape.

NEWSIZE-0 specifies that the size of the file is not to be changed.

Example

Restore the user ID file. The file is named UIDFILE and is contained on a diskette having a volume ID of IBM.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-UIDFILE,UNIT-I1,PACK-IBM
// FILE NAME-COPYO,LABEL-#SECREST,RETAIN-J
// RUN
// COPYFILE OUTPUT-S
// END
// LOAD $PRURS
// FILE NAME-#SECREST,LABEL-#SECREST,RETAIN-S
// RUN
// RESTORE NEWSIZE-0
// END
```

$PRUSV Utility

The $PRUSV utility program allows you to save the user ID file.

See the "SECSAVE Procedure" on page 4-451 for more information.

```
// LOAD $PRUSV
// RUN
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-#SECCOPY,RETAIN-S
// FILE NAME-COPYO,LABEL-file name,UNIT-{F1|I1|T1|T2|TC}
// RUN
// COPYFILE [OUTPUT-S]
// END
```

S9020422-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

OUTPUT-S is required when the file is being saved on diskette. It should not be used for disk or tape.

Example

Save the user ID file. The file is named UIDFILE and is to be contained on a diskette having a volume ID of IBM.

```
// LOAD $PRUSV
// RUN
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-#SECCOPY,RETAIN-S
// FILE NAME-COPYO,LABEL-UIDFILE,UNIT-I1,PACK-IBM
// RUN
// COPYFILE OUTPUT-S
// END
```

$RENAM Utility

The $RENAM utility program allows you to rename a file, library, or folder.

See the "RENAME Procedure" on page 4-372 for more information.

```
// LOAD $RENAM
// RUN

// RENAME LABEL-current name,NEWLABEL-new name [,DATE-{mmddyy|ddmmyy|yymmdd}]

// END
```

S9020423-0

The RENAME statement can be repeated to rename several files, libraries, or folders with only one load and run of the $RENAM utility program.

Example

Change the name of an existing file from OLDPAY to NEWPAY, change THIS to THAT, and change MYLIB to YOURLIB.

```
// LOAD $RENAM
// RUN
// RENAME LABEL-OLDPAY,NEWLABEL-NEWPAY
// RENAME LABEL-THIS,NEWLABEL-THAT
// RENAME LABEL-MYLIB,NEWLABEL-YOURLIB
// END
```

$RREDT Utility

The $RREDT utility program allows you to add, remove or update information about files, libraries, and groups in the resource security file.

See the "SECEDIT Procedure" on page 4-443 for more information.

```
// LOAD $RREDT
// RUN
```

S9020424-0

Example

Edit information about files, libraries, and groups in the resource security file.

```
// LOAD $RREDT
// RUN
```

$RRESC Utility

The $RRESC utility program allows you to:

- Create or remove the resource security file
- Activate or deactivate resource security

See the "SECDEF Procedure" on page 4-441 for more information.

```
// LOAD $RRESC
// RUN

// OPTIONS CONTROL-{CREATE  }
                   {DELETE  }
                   {ACTRES  }
                   {DEACTRES}

// END
```

S9020425-0

Example

Create the resource security file.

```
// LOAD $RRESC
// RUN
// OPTIONS CONTROL-CREATE
// END
```

$RRLST Utility

The $RRLST utility program allows you to list information about files, libraries, and groups in the resource security file.

See the "SECLIST Procedure" on page 4-445 for more information.

```
// LOAD $RRLST
// RUN

// OPTIONS LFORMAT-{OWNERID},LQUALFR-{USER}
                   {RNAME  }          {ALL }
                   {USERID }

// END
```

S9020426-0

Example

List information about files, libraries, and groups in the resource security file.

```
// LOAD $RRLST
// RUN
// OPTIONS LFORMAT-RNAME,LQUALFR-ALL
// END
```

$RRSAV Utility

The $RRSAV utility program allows you to save the resource security file.

See the "SECSAVE Procedure" on page 4-451 for more information.

```
// LOAD $RRSAV
// RUN
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-#SECCOPY,RETAIN-S

// FILE NAME-COPYO,LABEL-file name,UNIT-{F1}
                                       {I1}
                                       {T1}
                                       {T2}
                                       {TC}
// RUN

// COPYFILE [OUTPUT-S]

// END
```

S9020428-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

OUTPUT-S is required when the file is being saved on diskette. It should not be used for disk or tape.

Example

Save the resource security file. The file is named RESFILE and is to be contained on a diskette having a volume ID of IBM.

```
// LOAD $RRSAV
// RUN
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-#SECCOPY,RETAIN-S
// FILE NAME-COPYO,LABEL-RESFILE,UNIT-I1,PACK-IBM
// RUN
// COPYFILE OUTPUT-S
// END
```

$RRSTR Utility

The $RRSTR utility program allows you to restore the resource security file.

See the "SECREST Procedure" on page 4-448 for more information.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-file name,UNIT-{F1 }
                                       {I1 }
                                       {T1 },PACK-volume id
                                       {T2 }
                                       {TC }

// FILE NAME-COPYO,LABEL-#SECREST,RETAIN-J
// RUN

// COPYFILE [OUTPUT-S]

// END
// LOAD $RRSTR
// FILE NAME-#SECREST,LABEL-#SECREST,RETAIN-S
// RUN

// RESTORE NEWSIZE-{number of records}
                   {0               }

// END
```

S9020429-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Disk Files)" on page 5-32, the "FILE OCL Statement (for Diskette Files)" on page 5-43, and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Differences from the Procedure Parameters

OUTPUT-S is required when the file is being restored from diskette. It should not be used for disk or tape.

NEWSIZE-0 specifies that the size of the file is not to be changed.

Example

Restore the resource security file. The file is named RESFILE and is contained on a diskette having a volume ID of IBM.

```
// LOAD $COPY
// FILE NAME-COPYIN,LABEL-RESFILE,UNIT-I1,PACK-IBM
// FILE NAME-COPYO,LABEL-#SECREST,RETAIN-J
// RUN
// COPYFILE OUTPUT-S
// END
// LOAD $RRSTR
// FILE NAME-#SECREST,LABEL-#SECREST,RETAIN-S
// RUN
// RESTORE NEWSIZE-0
// END
```

$RRTED Utility

The $RRTED utility program allows you to add, remove, or update information about folders, folder members, and authorization lists in the resource security file.

See the "SECEDIT Procedure" on page 4-443 for more information.

```
// LOAD $RRTED
// RUN
```

S9020492-0

Example

Edit information about folders, folder members, and authorization lists in the resource security file.

```
// LOAD $RRTED
// RUN
```

$RRTLT Utility

The $RRTLT utility program allows you to list information about folders and authorization lists in the resource security file.

See the "SECLIST Procedure" on page 4-445 for more information.

```
// LOAD $RRTLT
// RUN

// OPTIONS LFORMAT-{OWNERID}{,LQUALFR-{USER}
                   {RNAME  }          {ALL }
                   {USERID }

// END
```

S9020427-0

Example

List information about folders and authorization lists in the resource security file.

```
// LOAD $RRTLT
// RUN
// OPTIONS LFORMAT-RNAME,LQUALFR-ALL
// END
```

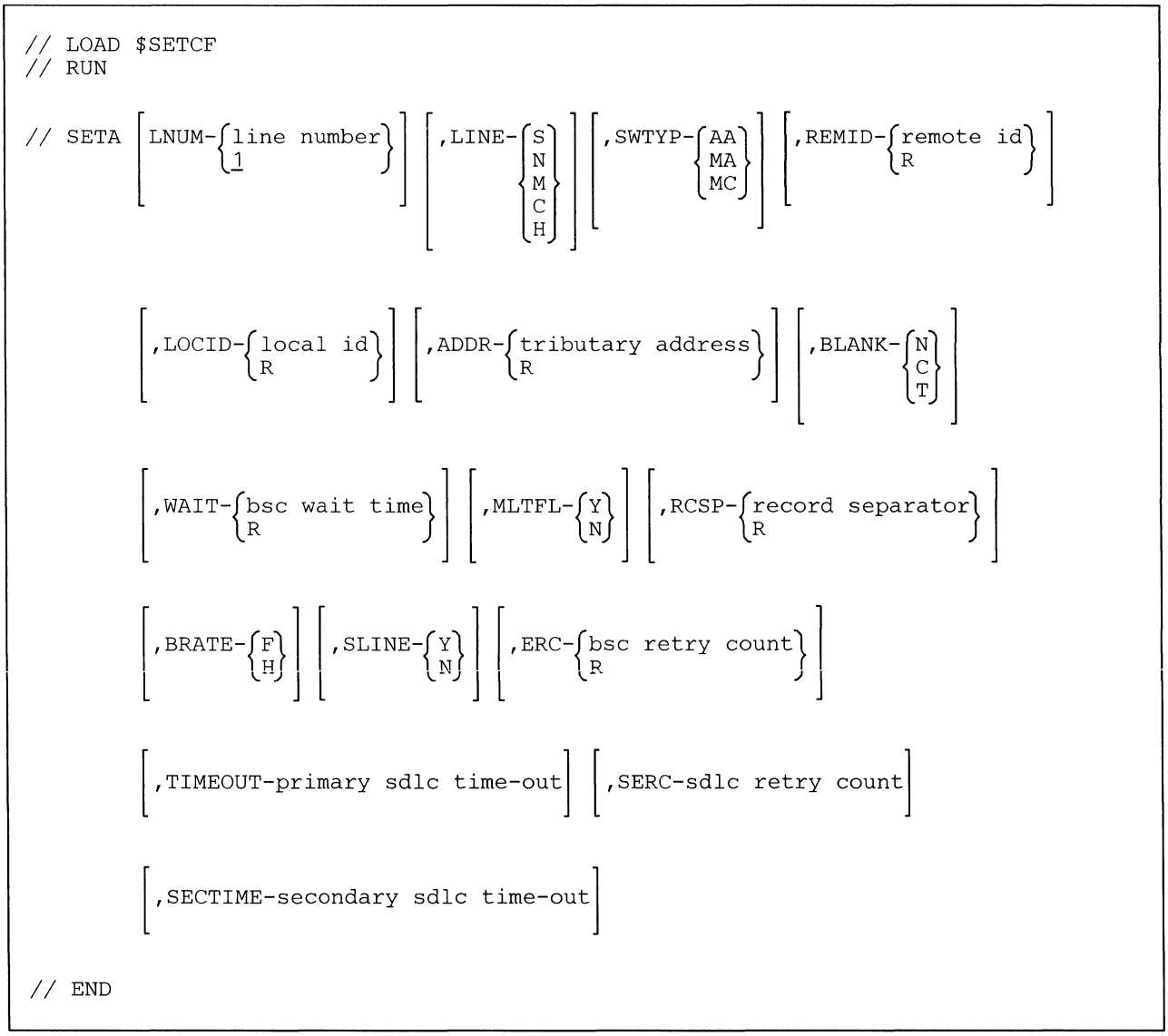
$SETCF Utility

The $SETCF utility program allows you to:

- Change certain items related to batch BSC, SDLC, a local area network (LAN), or a communications line (ALTERCOM procedure).
- Establish display station environment items, such as the printer for display station output (SET procedure).
- Specify information about the printer used for Print key output (PRINTKEY procedure).

Change Communications Attributes (ALTERCOM Procedure)

See the "ALTERCOM Procedure" on page 4-11 for more information.



S9020430-5

Differences from the Procedure Parameters

LINE-S specifies that the line is a point-to-point switched line.

LINE-N specifies that the line is a point-to-point nonswitched line.

LINE-M specifies that the System/36 is a multipoint tributary station.

LINE-C specifies that the System/36 is a multipoint control station.

LINE-H specifies that the line uses X.21 short hold mode.

BLANK-N specifies that neither blank compression nor truncation is to be performed.

BLANK-C specifies that embedded blanks are to be compressed.

BLANK-T specifies that trailing blanks are to be truncated.

Example

This example shows how to change the following:

- The line number is 3
- The tributary address for the multipoint line is 'TT' (EBCDIC E3)

```
// LOAD $SETCF
// RUN
// SETA LNUM-3,LINE-M,ADDR-E3
// END
```

Set Display Station Environment (SET Procedure)

See the "SET Procedure" on page 4-454 for more information.

```
// LOAD $SETCF
[// IMAGE MEMBER,print belt member name]
// RUN
// SETCF [LINES-lines per page] [,IMAGE-{YES|NO}] [,FORMAT-{MDY|DMY|YMD}]
       [,DATE-{mmddyy|ddmmyy|yymmdd}] [,LIBRARY-{library name|#LIBRARY|0}] [,RGSIZE-region size]
       [,PRINTER-{printer id|SYSTEM}] [,FORMSNO-forms number]
       [,ID-print key printer id] [,BORDER-{Y|N}] [,HEADER-{Y|N}]
// END
```

S9020431-0

Differences from the Procedure Parameters

IMAGE specifies whether an IMAGE OCL statement is to change the print belt image.

YES specifies that the print belt image is to be changed.

NO specifies that the print belt image is not to be changed.

Example

Change the print belt to BELT64C.

```
// LOAD $SETCF
// IMAGE MEMBER,BELT64C
// RUN
// SETCF IMAGE-YES
// END
```

Specify Print Key Information (PRINTKEY Procedure)

See the "PRINTKEY Procedure" on page 4-354 for more information.

```
// LOAD $SETCF
// RUN

// SETPK [ID-printer id] [,BORDER-{Y|N}] [,HEADER-{Y|N}]

// END
```

S9020432-0

Example

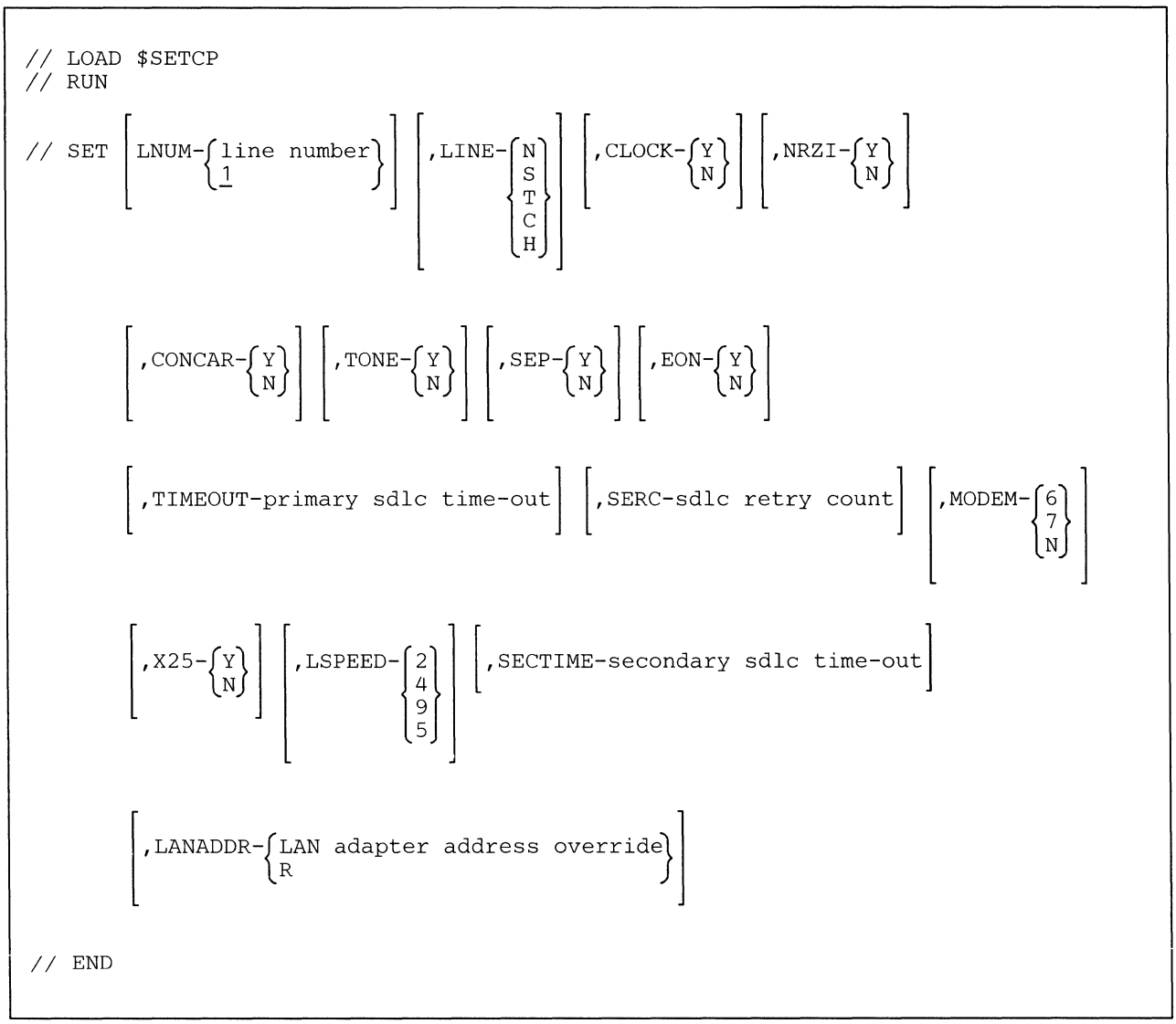
This example shows how to get a border and a header printed, and how to get the printed output to go to printer P4.

```
// LOAD $SETCF
// RUN
// SETPK ID-P4,BORDER-Y,HEADER-Y
// END
```

$SETCP Utility

The $SETCP utility program allows you to describe the communications hardware installed in your system.

See the "SETCOMM Procedure" on page 4-461 for more information.



S9020433-5

Differences from the Procedure Parameters

LINE-N specifies that the line is a point-to-point nonswitched line.

LINE-S specifies that the line is a point-to-point switched line.

LINE-T specifies that the System/36 is a multipoint tributary station.

LINE-C specifies that the System/36 is a multipoint control station.

LINE-H specifies that the line uses X.21 short hold mode.

MODEM-6 specifies an IBM LPDA modem.

MODEM-7 specifies an IBM wrap-capable modem.

MODEM-N specifies a non-IBM modem.

LSPEED-2 specifies a DDSA line with a line speed of 2400 bps.

LSPEED-4 specifies a DDSA line with a line speed of 4800 bps.

LSPEED-9 specifies a DDSA line with a line speed of 9600 bps.

LSPEED-5 specifies a DDSA line with a line speed of 56000 bps.

Example

To set communications line 1 to be nonswitched point-to-point.

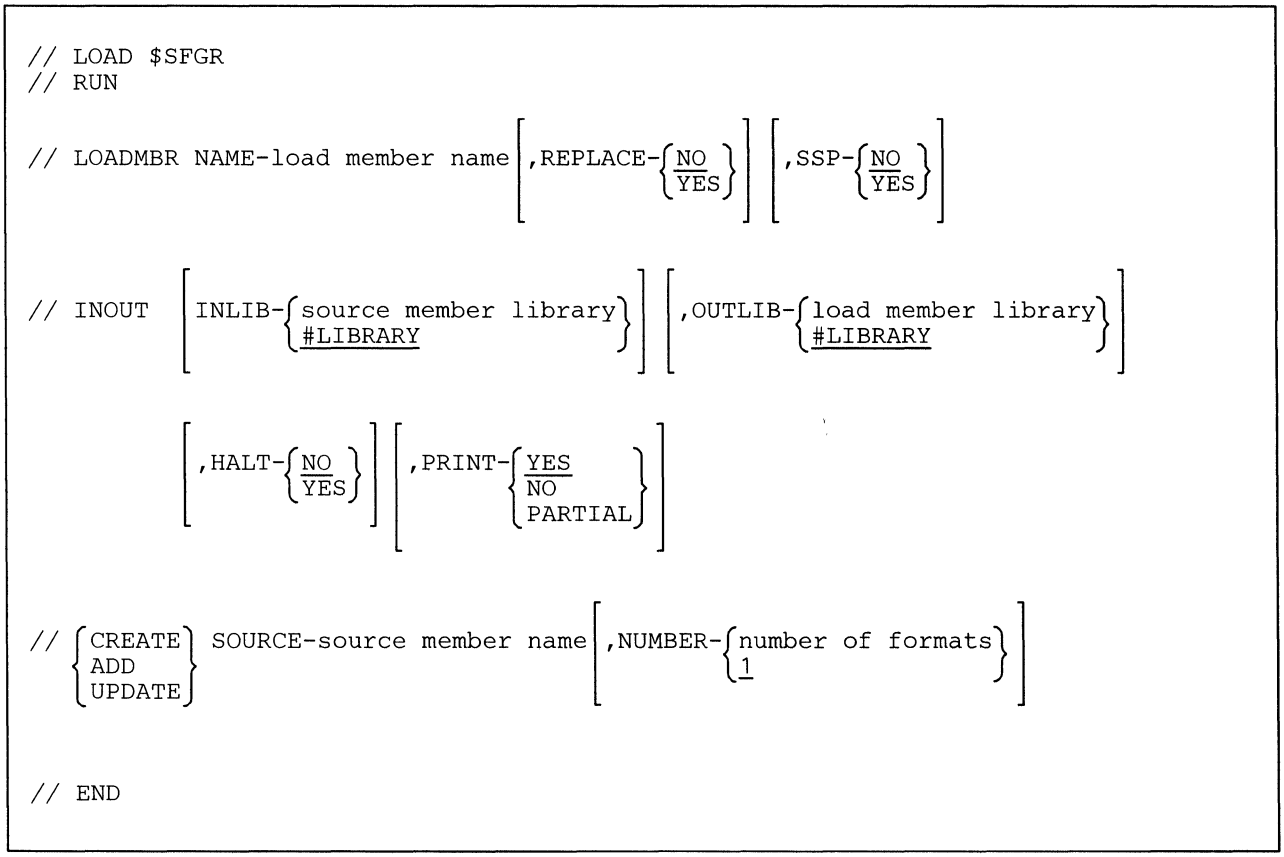
```
// LOAD $SETCP
// RUN
// SET LNUM-1,LINE-N
// END
```

$SFGR Utility

The $SFGR utility program allows you to create, update, add to, or delete display formats.

See the "FORMAT Procedure" on page 4-181 for more information.

For creating, adding to, or updating a display format load member:



S9020434-0

Differences from the Procedure Parameters

The CREATE, ADD, and UPDATE statements can be repeated in any order. You can specify up to 32 statements in any combination. For example: 16 CREATE statements, 8 UPDATE statements, and 8 ADD statements could be specified with only one load and run of the $SFGR utility program.

SSP specifies whether the load member is to be an SSP member. YES specifies that the load member being created is an SSP member and cannot be removed by using the REMOVE procedure. NO specifies the load member is a user member and can be removed by using the REMOVE procedure. If the SSP parameter is not specified, NO is assumed.

For deleting a format from a display format load member:

```
// LOAD $SFGR
// RUN
// LOADMBR NAME-load member name
[// INOUT OUTLIB-load member library]
// DELETE FORMAT-display format name
// END
```

S9020435-0

The DELETE statement can be repeated. You can specify up to 32 statements.

Example

Create a new display format load member named FORMAT2 in the library named MYLIB. Three display format source members are to be used to generate the load member. The three source members are named DISP1, DISP2, and DISP3; they are all stored in a library named MYLIB.

```
// LOAD $SFGR
// RUN
// LOADMBR NAME-FORMAT2,REPLACE-YES
// INOUT INLIB-MYLIB,OUTLIB-MYLIB,PRINT-PARTIAL
// CREATE SOURCE-DISP1,NUMBER-3
// CREATE SOURCE-DISP2,NUMBER-4
// CREATE SOURCE-DISP3,NUMBER-2
// END
```

$SINCT Utility

The $SINCT utility program allows you to restore the network resource directory (#NRD.FLE) as a system file. See the "$COPY Utility" on page A-9 for information about how to restore the network resource directory from diskette or tape to disk.

See the "RESTNRD Procedure" on page 4-389 for more information.

```
// LOAD $SINCT
// RUN
// END
```

S9020493-0

$SINDL Utility

The $SINDL utility program allows you to remove the network resource directory (#NRD.FLE) from disk.

See the "DELNRD Procedure" on page 4-150 for more information.

```
// MEMBER USER1-#SI#M1,USER2-#SI#M2
// LOAD $SINDL
// RUN
// END
```

S9020494-0

$SINR Utility

The $SINR utility program allows you to create and edit the network resource directory (#NRD.FLE).

See the "EDITNRD Procedure" on page 4-167 for more information.

```
// MEMBER USER1-#SI#M1,USER2-#SI#M2
// LOAD $SINR
// WORKSTN UNIT-?WS?,RESTORE-YES
// RUN
// END
```

S9020495-0

$SVCASRV Utility

The $SVCASRV utility program allows the system operator to create, change, or delete a disk cache.

See the "CACHE Procedure" on page 4-72 for more information.

```
// LOAD $SVCASRV
// RUN

// CACHE  [FUNC-{ALTER}]  [,SIZE-size]  [,PAGESIZE-pagesize]
          [      {START}]
          [      {STOP }]

// END
```

S9020523-0

Example

This example shows how to start a 64 kilobyte cache with a page size of 2 kilobytes.

```
// LOAD $SVCASRV
// RUN
// CACHE FUNC-START,SIZE-64,PAGESIZE-2
// END
```

$TCOPY Utility

The $TCOPY utility program allows you to:

- Copy data to or from tape
- List files on tape

Note: The $TCOPY utility program cannot be used with a tape cartridge.

Copy Data To or From Tape (TAPECOPY Procedure)

See the "TAPECOPY Procedure" on page 4-500 for more information.

To transfer data from disk to tape or tape to disk:

```
// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT-{F1 }
                       {T1 }
                       {T2 },LABEL-from file label

// FILE NAME-COPYO,UNIT-{F1 }
                       {T1 }
                       {T2 },LABEL-to file label

// RUN

// TRANSFER [ADD-{YES}] ,[CHKLAST-{YES}] ,[KEYLEN-value] ,[KEYLOC-value]
            [    {NO }]       [     {NO }]

// END
```

S9020436-0

To display the contents of tape file:

```
// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT-{T1 }
                       {T2 },LABEL-from file label

// RUN

// DISPLAY [FROM-value] ,[TO-value]

// END
```

S9020437-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statements, see the "FILE OCL Statement (for Disk Files)" on page 5-32 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Example 1

This example takes a file on disk named FILE1 and copies it to a tape file named FILE2. The tape is mounted on tape drive 1.

```
// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT-F1,LABEL-FILE1
// FILE NAME-COPYO,UNIT-T1,LABEL-FILE2
// RUN
// TRANSFER ADD-NO,CHKLAST-YES
// END
```

Example 2

This example takes a file from a tape mounted on tape drive 1 named SHOW and displays the first 100 records.

```
// LOAD $TCOPY
// FILE NAME-COPYIN,UNIT-T1,LABEL-SHOW
// RUN
// DISPLAY FROM-1,TO-100
// END
```

$TINIT Utility

The $TINIT utility program allows you to initialize a tape.

See the "TAPEINIT Procedure" on page 4-507 for more information about what this procedure does.

```
// LOAD $TINIT

// RUN

// VOL UNIT-{T1}  ,REEL-{STDLABEL}  [,VOLID-volume id]  [,TYPE-{CHECK}]
            {T2}      {(SL)      }                       {CLEAR}
            {TC}      {NONLABEL  }
                      {(NL)      }

      [,ID-characters]  [,SECURITY-{ERASE  }]  [,END-{REWIND}]
                                   {NOERASE}           {UNLOAD}

// END
```

S9020438-1

Example

Rename a tape so that the new volume ID is VOL001 and the owner ID is YOURNAME. Do not check for unexpired files. Erase the rest of the tape. The tape is mounted on tape drive 1 and should be rewound and unloaded after the tape has been prepared.

```
// LOAD $TINIT
// RUN
// VOL UNIT-T1,REEL-STDLABEL,VOLID-VOL001,TYPE-CLEAR,
       ID-YOURNAME,SECURITY-ERASE,END-UNLOAD
// END
```

$TMSERV Utility

The $TMSERV utility program allows you to:

- Copy a folder member or marked folder members onto disk, diskette, tape, or tape cartridge (ARCHIVE procedure)
- Copy a folder onto disk, diskette, tape, or tape cartridge (SAVEFLDR procedure)
- List the contents of a folder member that has been saved on disk, diskette, tape, or tape cartridge (LISTFILE procedure)
- Reorganize a folder (ALOCFLDR and CONDENSE procedures)
- Restore a folder member from disk, diskette, tape, or tape cartridge (RETRIEVE procedure)
- Restore a folder from disk, diskette, tape, or tape cartridge (RESTFLDR procedure)
- Move a folder from one location on disk to another location (MOVEFLDR procedure)

Copy a Folder Member onto Disk, Diskette, Tape, or Tape Cartridge (ARCHIVE Procedure)

See the "ARCHIVE Procedure" on page 4-22 for more information.

```
// LOAD $TMSERV
// FILE NAME-COPYO,UNIT-{I1|T1|T2|TC|F1}

// RUN
// ARCHIVE FOLDER-folder name,MEMTYPE-DOCUMENT,

          [MARKED-{YES|NO}],[DELETE-{KEEP|TEXT|MEMBER}],[SUBDIR-subdirectory],

          [MEMNAME-member name],[LABEL-file name]

// END
```

S9020439-3

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Copy a Folder onto Disk, Diskette, Tape, or Tape Cartridge (SAVEFLDR Procedure)

See the "SAVEFLDR Procedure" on page 4-428 for more information.

Note: If you are copying to disk, you must specify LABEL-file name, and it must be a different name than the folder name. If you are copying to diskette or tape, the LABEL-file name is optional and defaults to the folder name specified within the FOLDER-folder name parameter. You cannot use the LABEL parameter to change the name of the file to contain the folder when copying to diskette or tape.

```
// LOAD $TMSERV
// FILE NAME-COPYO,UNIT-{I1 }  ,LABEL-{file name  }
                        {T1 }         {folder name}
                        {T2 }
                        {TC }
                        {F1 }

// RUN

// SAVEFLDR [COMPRESS-{YES}] ,{FOLDER-folder name}
                       {NO }

// END
```

S9020440-4

```
// LOAD $TMSERV
// FILE NAME-COPYO,UNIT-{I1 }
                        {T1 }
                        {T2 }
                        {TC }

// RUN

// SAVEFLDR [COMPRESS-{YES}] ,{FOLDER-ALL}
                       {NO }

// END
```

S9020606-1

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

:SUBJECT=$TMSERV (LISTFILE).

List the Contents of an Archived Folder Member (LISTFILE Procedure)

See the "LISTFILE Procedure" on page 4-267 for more information.

```
// LOAD $TMSERV
// FILE NAME-COPYIN,UNIT-{I1}
                         {T1}
                         {T2}
                         {TC}

// RUN
// LISTARCH,LABEL-file name
// END
```

S9020441-2

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Reorganize a Folder (ALOCFLDR and CONDENSE Procedures)

See the "ALOCFLDR Procedure" on page 4-8 and the "CONDENSE Procedure" on page 4-109 for more information.

To reorganize a folder and, optionally, increase or decrease the size of the folder (ALOCFLDR procedure):

```
// LOAD $TMSERV
// RUN

// RORGFLDR FOLDER-folder name,[CHANGE-{DECR}],[BLOCKS-value]
                                {INCR}

// END
```

S9020442-0

To reorganize a folder and make the folder as small as possible (CONDENSE procedure or ALOCFLDR procedure):

```
// LOAD $TMSERV
// RUN
// RORGFLDR FOLDER-folder name,CHANGE-MIN
// END
```

S9020443-0

Move a Folder (MOVEFLDR Procedure)

See the "MOVEFLDR Procedure" on page 4-298 for more information.

To move a folder from one location on disk to another location on disk (MOVEFLDR procedure):

```
// LOAD $TMSERV
// RUN
// MOVEFLDR FLDRNAME-folder name,[FLDRLOC-{A1          }],[EXTLOC-{A1          }]
                                         {A2          }          {A2          }
                                         {A3          }          {A3          }
                                         {A4          }          {A4          }
                                         {block number}          {block number}

// END
```

S9020584-0

Restore a Folder Member from Disk, Diskette, Tape, or Tape Cartridge (RETRIEVE Procedure)

See the "RETRIEVE Procedure" on page 4-401 for more information.

```
// LOAD $TMSERV
// FILE NAME-COPYIN,UNIT-{I1|T1|T2|TC|F1}

// RUN
// RETRIEVE FOLDER-folder name,[DATE-date archived],[SUBDIR-subdirectory],

          [NMEMNAME-new member name],LABEL-file name

// END
```

S9020444-4

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

Restore a Folder from Disk, Diskette, Tape, or Tape Cartridge (RESTFLDR Procedure)

See the "RESTFLDR Procedure" on page 4-383 for more information.

```
// LOAD $TMSERV
// FILE NAME-COPYIN,UNIT-{I1 }  ,LABEL-{file name  }
                         {T1 }         {folder name}
                         {T2 }
                         {TC }
                         {F1 }

// RUN

                [                 ]
                [          {A1  } ]
// RESTFLDR     [DISKLOC-  {A2  } ]
                [          {A3  } ]
                [          {A4  } ]
                [          {value} ]

// END
```

S9020445-2

Only some of the parameters are shown for the FILE OCL statement. For more information about the FILE OCL statement, see the "FILE OCL Statement (for Diskette Files)" on page 5-43 and the "FILE OCL Statement (for Tape Files)" on page 5-48.

$UASC Utility

The $UASC utility program allows you to display or print entries copied from the spool file.

See the "COPYPRT Procedure" on page 4-126 for more information.

```
// LOAD $UASC
// FILE NAME-file name,DISP-SHR
// RUN
```

S9020446-0

Example

Display the spool file entries that were copied into a file named SPOOLENT.

```
// LOAD $UASC
// FILE NAME-SPOOLENT
// RUN
```

$UASF Utility

The $UASF utility program allows you to copy entries from the spool file to a disk file.

See the "COPYPRT Procedure" on page 4-126 for more information.

```
// LOAD $UASF
// RUN
// SPOOL [SPOOLID-{ALL      }] [,NAME-file name] [,RELCANS-{RELEASE}]
                  {spool id }                            {CANCEL }
                  {Fxxxx    }
                  {SYSTEM   }

          [,RETAIN-{T}]
                   {J}
                   {S}

// END
```

S9020447-0

Differences from the Procedure Parameters

RETAIN classifies the file as a **resident (T)**, **job (J)**, or **scratch (S)** file when it is copied. If no parameter is specified, T is assumed.

Example 1

Copy all spool file entries into a file named SPFILE. The statements are entered from the system console by the system operator.

```
// LOAD $UASF
// RUN
// SPOOL SPOOLID-SYSTEM,NAME-SPFILE
// END
```

Example 2

Copy all spool file entries with forms number 0017 into a file named F0017W3. The statements are entered from display station W3.

```
// LOAD $UASF
// RUN
// SPOOL SPOOLID-F0017,NAME-F0017W3
// END
```

$XNLM Utility

The $XNLM utility program allows you to create or update lists of connection numbers for an X.21 public data network.

See the "DEFINX21 Procedure" on page 4-141 for more information.

```
// MEMBER PROGRAM1-#XN#M1,PROGRAM2-#XN#M2
// LOAD $XNLM
// RUN
```

S9020448-0

Example

To create a call list, you would enter:

```
// MEMBER PROGRAM1-#XN#M1,PROGRAM2-#XN#M2
// LOAD $XNLM
// RUN
```

$XNSH Utility

The $XNSH utility program allows you to create, update, print, or remove an X.21 short hold mode line configuration.

See the "DEFINX21 Procedure" on page 4-141 for more information.

```
// MEMBER PROGRAM1-#XN#M1,PROGRAM2-#XN#M2
// LOAD $XNSH
// RUN
```

S9020515-0

Example

To create an X.21 short hold mode line configuration, you would enter:

```
// MEMBER PROGRAM1-#XN#M1,PROGRAM2-#XN#M2
// LOAD $XNSH
// RUN
```

$XREST Utility

The $XREST utility program allows you to restore the extended character file.

See the "RESTEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-380 for more information.

```
// LOAD $XREST
// FILE NAME-COPYIN,UNIT-I1,LABEL-file name
// FILE NAME-COPYO,LABEL-{#EXT1818}
                         {#EXT2424}

// RUN


// SELECT [FROM-{ALL           }] [,TO-{ending value}] [,REPLACE-{YES}]
                {IBM           }       {IGC number }          {NO }
                {USER          }
                {starting value}
                {IGC number    }


// END
```

S9020449-0

For more information about the FILE statements, see the "FILE OCL Statement (for Diskette Files)" on page 5-43.

Example

This example copies only the user-defined extended characters from an extended character file on diskette name #EXT2424 to a disk file with the same name.

```
// LOAD $XREST
// FILE NAME-COPYIN,UNIT-I1,LABEL-#EXT2424
// FILE NAME-COPYO,LABEL-#EXT2424
// RUN
// SELECT FROM-USER
// END
```

$XSAVE Utility

The $XSAVE utility program allows you to save the extended character file.

See the "SAVEEXTN Procedure (for the Ideographic Version of the SSP)" on page 4-425 for more information.

```
// LOAD $XSAVE
// FILE NAME-COPYO,UNIT-I1,LABEL-file name
// FILE NAME-COPYIN,LABEL-{#EXT1818}
                          {#EXT2424}

// RUN


// SELECT [FROM-{ALL            }] [,TO-{ending value}]
                {IBM            }       {IGC number }
                {USER           }
                {starting value }
                {IGC number     }


// END
```

S9020450-0

For more information about the FILE statements, see the "FILE OCL Statement (for Diskette Files)" on page 5-43.

Example

This example saves only the user-defined extended characters from an extended character file on disk named #EXT2424 to a diskette file with the same name.

```
// LOAD $XSAVE
// FILE NAME-COPYO,UNIT-I1,LABEL-#EXT2424,PACK-IGC
// FILE NAME-COPYIN,LABEL-#EXT2424
// RUN
// SELECT FROM-USER
// END
```

#GCFR Utility

The #GCFR utility program allows you to register or cancel an available user facility on an X.21 public data network.

See the "REQUESTX Procedure" on page 4-373 for more information.

```
[SEU member name,S,#GC@REQX,,library name]

 // LOAD #GCFR
 // RUN
 // REQX SOURCE-member name,LIBRARY-library name
 // END
```

S9020451-0

Example 1

Start the REQUESTX procedure, which will then prompt for the requests.

```
// LOAD #GCFR
// RUN
// END
```

Example 2

Start the REQUESTX procedure, which will then process the library source member REQ from the current library (named MYLIB).

```
// LOAD #GCFR
// RUN
// REQX SOURCE-REQ,LIBRARY-MYLIB
// END
```

Appendix B. Conversions

Converting Hexadecimal to Decimal

You can use the following chart to convert a decimal number to a hexadecimal number, or to convert a hexadecimal number to a decimal number. Examples of how to use the chart are included.

Byte 2				Byte 1			
Position 4		Position 3		Position 2		Position 1	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	57344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

Hexadecimal to Decimal Example

To find the decimal value of hex 1FA, you would find that:

1. In position 3, hex 1 equals decimal 256
2. In position 2, hex F equals decimal 240
3. In position 1, hex A equals decimal 10

By adding these three decimal numbers together, you have the decimal value of hex 1FA.

```
256 + 240 + 10 = 506.
```

Byte 2				Byte 1			
Position 4		Position 3		Position 2		Position 1	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	1	256	F	240	A	10

Decimal to Hexadecimal Example

To find the hexadecimal value of decimal 538, you would find that:

1. The next lower decimal number in the chart is 512 in position 3. This is equal to hex 2.
2. Then subtract 512 from 538 and use the difference to find the next hexadecimal value.

   ```
   538 - 512 = 26
   ```

3. The next lower number in the chart from 26 is 16 in position 2. This is equal to hex 1.
4. Then subtract 16 from 26 and use the difference to find the next hexadecimal value.

   ```
   26 - 16 = 10
   ```

5. The remaining 10 is found in position 1 of the chart. This is equal to hex A.

You then combine the positions of the hexadecimal values. Thus, decimal 538 equals hex 21A.

Byte 2				Byte 1			
Position 4		Position 3		Position 2		Position 1	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	2	512	1	16	A	10

Records to Blocks Conversion for Disk Files

This section helps you determine how many blocks a disk file uses when the file is allocated by number of records. One disk block contains 10 sectors. Each sector contains 256 bytes.

Determining the Number of Blocks in a Sequential or Direct File

To determine the number of blocks in a sequential or direct file:

1. Multiply the number of records by the record length to determine the number of bytes in the data portion of the file.

   ```
   (number of records) x (record length) = (number of bytes)
   ```

2. Divide the number of bytes by the number of bytes per block. One block contains 2560 bytes.

   ```
   (number of bytes) / 2560 = (number of blocks)
   ```

   Round the number of blocks up to the next whole number.

Example

A sequential disk file was allocated with 200 records. The record length of the file is 15.

1. The number of bytes in the file is 3000:

   ```
   200 x 15 = 3000
   ```

2. The number of blocks used by the file is 2:

   ```
   3000 / 2560 = 1.1 (round up)
                 2   (number of blocks)
   ```

Determining the Number of Blocks in an Indexed File

To determine the number of blocks in an indexed file, you must first determine the number of disk sectors in an indexed file:

1. Multiply the number of records in the file by the record length of the file to determine the number of bytes in the file.

   ```
   (number of records) x (record length) = (number of bytes)
   ```

2. Divide the number of bytes required for the data area by the number of bytes per disk sector to determine the number of data sectors used by the file. One disk sector contains 256 bytes. If the number contains a fraction, round the number up to the next whole number.

   ```
   (number of bytes) / 256 = (number of data sectors)
   ```

Next, you must determine the number of disk sectors used by the index of the indexed file:

3. Add the length of the key field to the number 3 to determine the length of each index entry. The 3-byte addition is for the 3-byte relative record number that is associated with each key in the index.

```
(length of key field) + 3 = (length of index entry)
```

4. Divide the number of bytes in a sector (256) by the length of the index entry to determine the number of index entries per sector. If the number has a fraction, use only the whole number.

```
256 / (length of index entry) = (number of index   )
                                (entries per sector)
```

5. Divide the number of records in the indexed file by the number of index entries per sector to determine the number of sectors in the index. If the number contains a fraction, round the number up to the next whole number. Then add 1 to the result.

```
(number of) / (number of index   ) + 1 = (number of sectors)
(records  )   (entries per sector)       (in the index     )
```

Now determine the total number of blocks used by an indexed file:

6. Add the number of data sectors from step 2 to the number of sectors in the index to determine the total number of sectors in the file.

```
(number of   ) + (number of sectors) = (total number)
(data sectors)   (in the index     )   (of sectors  )
```

7. Divide the total number of sectors by the number of sectors per block to determine the total number of blocks for the file. One block contains 10 sectors. If there is a remainder, round the number up to the next whole number.

```
(total number of sectors) / 10 = (number of blocks)
```

Example

An indexed file was allocated with 200 records. The record length of the file is 15; the length of the key field is 12.

The number of data sectors in the file is 12.

```
200 x 15 = 3000 (number of characters)

3000 / 256 = 11.7 (round up)
             12   (number of data sectors)
```

The number of sectors in the index is 13.

```
12 + 3 = 15 (length of index entry)

256 / 15 = 17.1 (drop fraction)
           17   (number of index entries per sector)

200 / 17 = 11.8 (round up)
           12
12 + 1 = 13     (add 1)
         13     (number of sectors in index)
```

The total number of blocks is 3.

```
12 + 13 = 25 (total number of sectors in the file)

25 / 10 = 2.5 (round up)
          3   (number of blocks)
```

Appendix C. Service Aid Procedures

This appendix describes the System/36 service aid procedures. You will not normally have to run these procedures; however, you may be asked to run some of these procedures to do problem determination and correction.

For information about problem determination, see the manual *System Problem Determination*. For more information about these service aid procedures, see the manual *Program Problem Diagnosis and Diagnostic Aids*, SY21-0593.

APAR Procedure

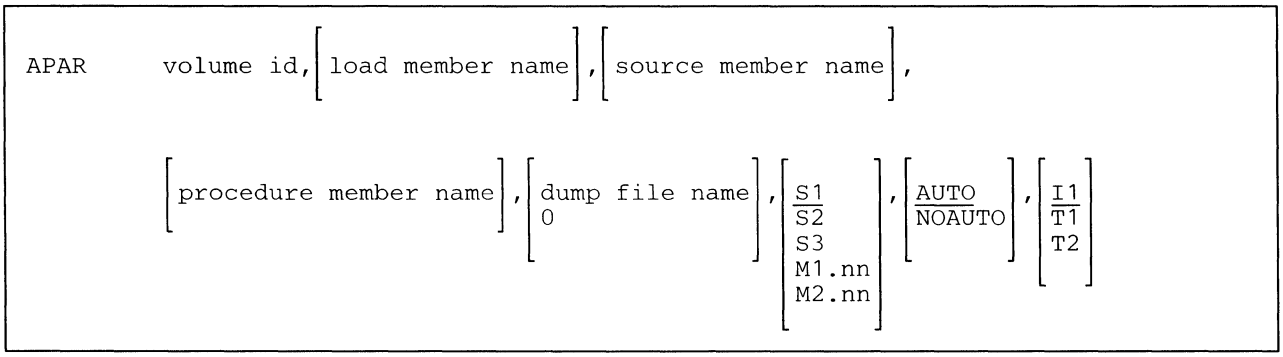
The APAR procedure collects diagnostic information that helps IBM service people to correct programming problems that might occur in the system. The APAR procedure creates one or more diskette or tape files that contain the following:

- Control storage dump area.
- The input/output controller storage dump area.
- The system work area (if you are not running the APAR procedure during IPL after a system dump), including:
  - The system configuration
  - The disk volume table of contents (VTOC)
  - The #SYSWORK index
  - The trace work area
  - The security work area
  - The PTF work area
  - The diskette VTOC
  - The volume label
  - The IPL bootstrap
- IBM program product library and system library program temporary fix (PTF) logs.
- The system service log.
- The disk trace files. If you are not running the APAR procedure during IPL and you choose not to copy a task dump, a trace file prompt display is shown and you can select up to 16-trace files to copy.
- Microcode tables.
- Error logging (ERAP) tables.
- Task dump file (optional).
- The history file.
- The spool file (optional).
- The job queue (optional).
- The message file (optional).
- The product level data file.

Also, the APAR procedure can copy a specified load member to a diskette or tape file named APARLOAD, a specified source member to a diskette or tape file named APARSRCE, and a specified procedure member to a diskette or tape file named APARPROC. When the APAR procedure begins running, a display appears, and you can select the spool file, job queue, message file, and user file index to be copied to the diskette or tape.

Most of the data areas copied to diskette or tape by the APAR procedure can be displayed using the DUMP procedure, or the diskettes or tapes can be included with an Authorized Program Analysis Report (APAR).

The APAR procedure should be run during IPL after a system dump has been taken.



S9020453-0

Parameters 6 and 7 are ignored for systems without diskette magazine drives.

volume id specifies the volume ID of the one or more diskettes or tapes to receive the system data areas.

load member name specifies the load member containing the program that caused the program check to occur. The load member is placed in a diskette or tape file named APARLOAD.

source member name specifies the source member from which the program was created. The source member is placed in a diskette or tape file named APARSRCE.

procedure member name specifies the procedure member from which the program was called. The procedure member is placed in a diskette or tape file labeled APARPROC.

dump file name specifies the file created by a task dump. 0 (zero) specifies that the most recent dump file is to be copied. If no file name is specified and the APAR procedure is being run from a display station, the status of all dump files is displayed, and you can select one (or none) of the files to copy. If no file name is specified and the APAR procedure is not being run from a display station, no dump file is copied to diskette or tape.

Dump files are named #DUMP.nn on disk, where nn is a number from 00 through 99.

S1, S2, or S3 specifies the diskette slot containing the first diskette to be used. If a parameter is not specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette to be used. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three diskette slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

  If a parameter is not specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

I1 specifies that the information collected by the APAR procedure is to be copied to diskette. If no parameter is specified, I1 is assumed.

T1 or T2 specifies that the information collected by the APAR procedure is to be copied to tape. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2.

Example

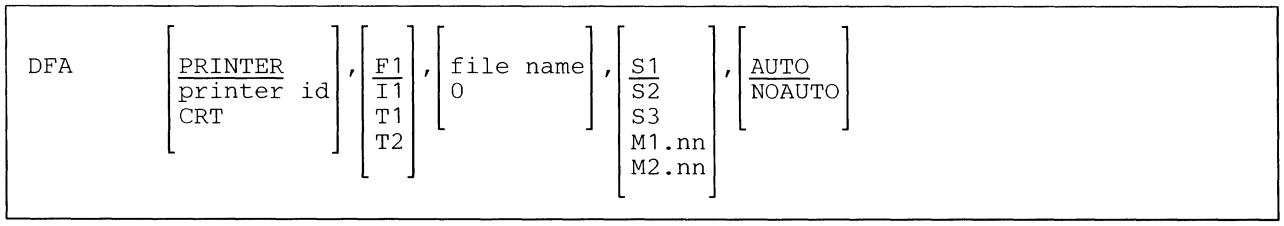
The following example shows how to enter the APAR procedure to copy the system data areas to one or more diskettes. A file named #DUMP.03 was created by a task dump. The diskettes have volume IDs of VOL001, and are located in diskette magazine slots S1, S2, and S3. The load member named PROGRAM is also to be copied.

```
APAR VOL001,PROGRAM,,,#DUMP.03
```

DFA Procedure

The DFA (dump file analysis) procedure retrieves selected information from a dump file, formats the information, and either prints or displays it. The DFA procedure can also be used to format dump files copied to diskette or tape by the APAR procedure.

The DFA procedure can be run from any display station. If password security is active, the operator must have service aid authority to run the DFA procedure.



S9020454-0

Parameters 4 and 5 are ignored for systems without diskette magazine drives.

PRINTER specifies that the output should be printed on the session printer assigned to the display station running DFA. If no parameter is specified, PRINTER is assumed.

printer id specifies the work station ID of the printer that is to print the output.

CRT specifies that the output is to be displayed at the display station running the DFA procedure.

F1 specifies that the input to the DFA procedure is a dump file on disk. If no parameter is specified, F1 is assumed.

I1 specifies that the input to the DFA procedure is a dump file on diskette that was created by the APAR procedure, by a system dump, or by a task dump.

T1 or T2 specifies that the input to the DFA procedure is a dump file on tape that was created by the APAR procedure, by a system dump, or by a task dump. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2.

file name specifies the file to be processed by DFA. If F1 is specified, this file must have been created by a task dump, the DUMP procedure, or a system dump. Dump files are named #DUMP.nn on disk, where nn is a number from 00 through 99.

If I1 is specified, the file must be a diskette file created by the APAR procedure, by a system dump, or by a task dump. The DFA procedure first copies the APAR file to disk. If the file name specified already exists on the disk as a dump file, it is replaced. If the file name specified already exists on disk as a user file, a message is displayed, and you can either replace the file or cancel the procedure.

0: If F1 is specified, 0 (zero) indicates that DFA is to use the most recent dump file. If you place the DFA procedure on the job queue, 0 is assumed if no file name is specified.

If I1 is specified, 0 indicates that the APARFILE file should be copied from diskette to a disk file. The disk file is automatically deleted when DFA ends.

S1, S2, or S3 specifies the diskette slot containing the first diskette from which members are to be processed. If no parameter is specified, S1 is assumed.

M1.nn or M2.nn specifies the magazine location containing the first diskette from which members are to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

- If T1 or T2 is specified, when the tape reel on the original tape drive is finished, the system will switch to the other tape drive, if it is available, to continue processing (T1 to T2 or T2 to T1). If the other drive is not available, the original drive will be used. A prompt will be issued to verify that the mounted reel is the correct reel to continue processing.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

- If T1 or T2 is specified, only that specified tape drive will be used for all tape volumes.

Example

To format and print the information in the most recent disk dump file.

```
DFA ,,0
```

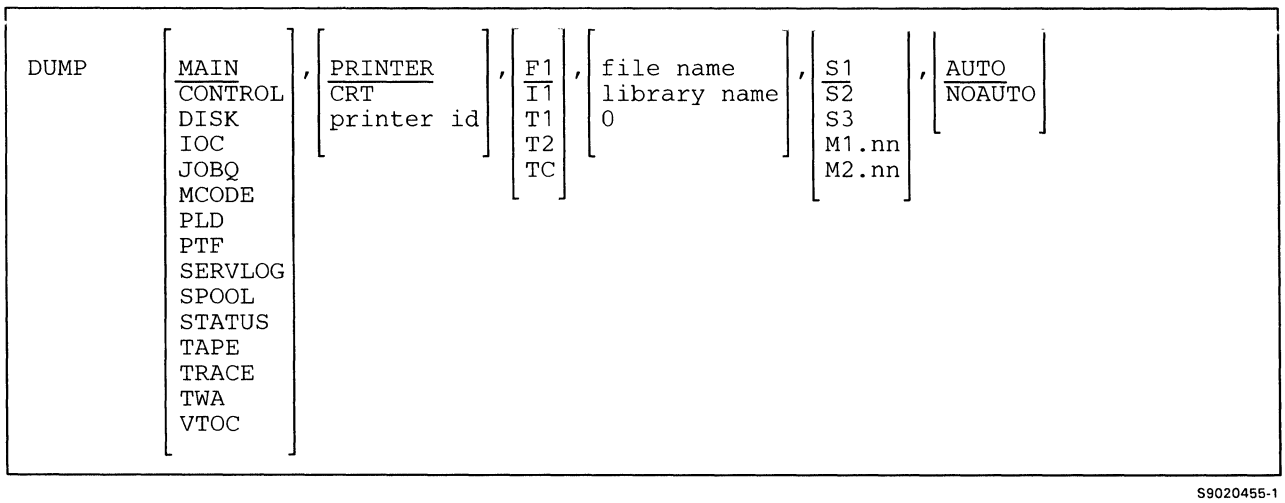
DUMP Procedure

The DUMP procedure prints or displays any of the following areas from disk, or from a diskette or tape created by the APAR procedure:

- Task dump file
- System dump file
- Control storage dump file
- Input/output controller storage dump file
- Disk trace file
- Library PTF log
- Disk VTOC (volume table of contents)
- Spool file
- Task work area
- Input job queue
- Microcode tables
- System service log
- Product level data file

DUMP can also print or display selected sectors from the disk or diskette, and print selected blocks from a tape or tape cartridge.

The DUMP procedure can be run from any display station. If password security is active, some functions of the DUMP procedure are restricted to operators who have service aid authority. For more information about service aid authority, see the *System Security Guide*.



S9020455-1

MAIN specifies that selected portions of a task or system dump file are to be listed. If password security is active, the operator must have service aid authority to use DUMP MAIN.

If the information is displayed, the DUMP procedure displays the following:

- A storage dump summary
- The first segment of main storage
- The address of the task control block and request block for the abnormally ended task
- The last set of saved register values for the abnormally ended task

You can use the roll keys to page forward and backward through the data. You can also specify a different storage address or storage type. The possible storage types are:

M Untranslated main storage

V Untranslated virtual storage

X Translated task storage

C Control storage tables

F Data storage controller tables

If the information is printed, the DUMP procedure first displays and prints the storage dump summary. You can then select one of the following options:

M Untranslated main storage between limits

V Untranslated virtual storage between limits

X Translated task storage between limits

C Control storage tables

F Data storage controller tables

N Nucleus storage dump

P Formatted control block dump, where frequently used storage areas are printed including the control storage tables

S System dump, which includes the following:

- A formatted control block dump
- A nucleus storage dump
- A dump of the translated task storage for each program of each task included in the dump file
- A dump of the virtual storage associated with the task in error
- A trace table dump
- A DFA listing

T All existing main and alternate trace tables and files

To list the translated storage of another task, change the address of the task block shown on the display. A system dump contains the translated storage for all tasks that were active at the time of the abnormal ending, but a task dump contains only the storage for selected tasks.

To list the translated storage of another program for this task, change the address of the request block shown on the display.

To list virtual storage, change the address of the storage block shown on the display.

The control storage data areas contain the following:

- Direct area 0 (hex 0000 through hex 007F)
- Direct area 1 (hex 1000 through hex 107F)
- Direct area 2 (hex 2000 through hex 207F)
- Direct area 3 (hex 3000 through hex 307F)
- Direct area 4 (hex 4000 through hex 407F)
- Direct area 5 (hex 5000 through hex 507F)
- Direct area 6 (hex 6000 through hex 607F)
- Direct area 7 (hex 7000 through hex 707F)
- Machine check logout area (hex 7A00 through hex 7B7F)
- Other miscellaneous areas

If DUMP MAIN is started using the EVOKE OCL statement or is run from the job queue, a system dump (S) will be printed.

CONTROL specifies that selected portions of a control storage dump file are to be listed. If password security is active, the operator must have service aid authority to use DUMP CONTROL.

If the information is displayed, the DUMP procedure displays the first segment of control storage. You can use the Roll keys to page forward and backward through the data. You can also specify a different control storage address.

If the information is printed, the DUMP procedure first prompts for the control storage limits.

If DUMP CONTROL is started using the EVOKE OCL statement or is run from the job queue, all of control storage will be printed.

DISK specifies that selected disk or diskette sectors are to be listed. You are prompted for the address of the first sector to be displayed. If the information is printed, you are prompted for the number of sectors to be printed. If password security is active, the operator must have service aid authority to use DUMP DISK.

IOC specifies that selected portions of the input/output controller storage dump file are to be listed. If password security is active, the operator must have service aid authority to use DUMP IOC. If the information is to be displayed, the DUMP procedure displays the following:

- The first segment of controller storage for the first device found in the dump file
- The address of the first device found in the dump file

To display other portions of the dump file, you can use the Roll keys to page forward and backward through the data for this device. You can use command keys 1 and 2 to page forward and backward through the devices in the dump file. You can also specify a different device address.

If the information is printed, the DUMP procedure prompts for the address of the device to be listed. If you enter 00, the data for all devices in the dump file are listed.

For a list of the possible device addresses, press command key 8.

If DUMP IOC is started using the EVOKE OCL statement or is run from the job queue, all the data for all the devices in the dump file is listed.

JOBQ specifies that the job queue is to be listed. If password security is active, the operator must have service aid authority to use DUMP JOBQ. If the information is to be displayed, the DUMP procedure displays the following:

- The first segment of the job queue.
- If F1 is specified and no file name is specified, the disk sequential sector address for the first segment is displayed.
- If I1, T1, T2, or a file name is specified, the address relative to the start of the job queue is displayed.

To display other portions of the job queue file, you can use the Roll keys to page forward and backward through the data for this device.

If the information is printed, the DUMP procedure prints the entire job queue file.

MCODE specifies that the following microcode tables are to be listed:

- Prerequisite list (including hardware required, SSP release and modification levels, and the last SSP PTF applied)
- Microcode level table (including microcode release and modification levels)
- Patch table (including module identifiers and dates of all patches)

If password security is active, the operator must have service aid authority to use DUMP MCODE.

If the information is printed, the DUMP procedure prints all the microcode tables.

PLD specifies that the product level data file is to be listed or displayed.

PTF specifies that a library PTF log, which identifies all PTFs applied to a library, is to be listed. If no library name is specified, the system library is assumed. If the information is to be displayed, the DUMP procedure displays the first segment of the PTF log. To display other portions of the PTF log, you can use the Roll keys to page forward and backward through the data.

If I1,T1, or T2 is specified or if a file name (not a library name) is specified, you can use command keys 1 and 2 to page forward and backward through other PTF logs.

If the information is printed, the DUMP procedure prints the entire PTF log.

SERVLOG specifies that the system service log is to be listed. If the information is to be displayed, the DUMP procedure displays the newest entry in the service log. To display other entries in the service log, you can use the roll keys to page forward and backward through the data.

If the information is printed, the DUMP procedure prints the entire service log, beginning with the oldest entry.

SPOOL specifies that the one or more spool file extents are to be listed. If password security is active, the operator must have service aid authority to use DUMP SPOOL. If the information is to be displayed, the DUMP procedure displays the following:

- The first segment of the first spool file extent.
- If F1 is specified and a file name is not specified, the disk sequential sector address of the first segment is displayed.
- If I1 or a file is specified, the address relative to the start of the first spool file is displayed.

To display other portions of the spool file, you can use the roll keys to page forward and backward through the data. You can also use command keys 1 and 2 to page forward and backward through other spool file extents.

If the information is printed, the DUMP procedure prints all the spool file extents.

STATUS specifies that the status (a storage dump summary) of one or more system or task dump files is to be listed. If the information is to be displayed, the DUMP procedure displays the status for the most recent task dump file. To display the status of other task dump files, you can use the Roll Up key (↑) to page forward through the data. You can use the Roll Down key (↓) to page backward through the data.

If the information is printed, the DUMP procedure prints the status for all the task dump files, beginning with the most recent task dump file.

TAPE specifies that selected tape blocks are to be listed. You are prompted for the number of blocks to print. If password security is active, the operator must have service aid authority to use DUMP TAPE.

TRACE specifies that the disk trace file is to be listed. If the information is from a task dump file, the DUMP procedure prompts for the specified trace file to be listed. You must select one of the trace files to continue.

If the information is to be displayed, the DUMP procedure displays the trace file starting with the most recent entry. To display other portions of the trace file, you can use the roll keys to page forward and backward through the data.

If the information is printed, the DUMP procedure prints the trace file, beginning with the oldest entry.

TWA specifies that the task work area is to be listed. If password security is active, the operator must have service aid authority to use DUMP TWA. If the information is to be displayed, the DUMP procedure displays the following:

- The first segment of the task work area.
- If F1 is specified and a file name is not specified, the disk sequential sector address of the first segment is displayed.
- If I1, T1, T2, or a file name is specified, the address relative to the start of the task work area is displayed.

To display other portions of the task work area, you can use the roll keys to page forward and backward through the data. You can use command key 1 to page forward through other task work area extents. You can use command key 2 to return to the first task work area extent.

If the information is printed, the DUMP procedure prints the entire task work area.

VTOC specifies that the disk VTOC is to be listed. If the information is to be displayed, the DUMP procedure displays the following:

- The first segment of the disk VTOC.
- If F1 is specified and a file name is not specified, the disk sequential sector address of the first segment is displayed.
- If I1 or a file name is specified, the address relative to the start of the disk VTOC is displayed.

To display other portions of the disk VTOC, you can use the roll keys to page forward and backward through the data.

If the information is printed, the DUMP procedure prints the entire disk VTOC.

PRINTER specifies that output is to be printed on the session printer assigned to the display station. If the parameter is not specified, PRINTER is assumed.

CRT specifies that the output is to be displayed at the display station running the DUMP procedure. This parameter is not allowed if TAPE is specified.

F1 specifies that the DUMP procedure is to process the dump information from a disk file. If the parameter is not specified, F1 is assumed.

I1 specifies that the DUMP procedure is to process the dump information from a diskette created by the APAR procedure, by a system dump, or by a task dump. If DISK was selected for the first parameter, then information from specified diskette sectors is listed for any diskette. If DISK was selected and the system has a diskette magazine drive, the diskette must be in slot S1.

T1 or T2 specifies that the DUMP procedure is to process the dump information from a tape. This may be a tape created by the APAR procedure, a system dump, a task dump, or simply a tape to be listed using TAPE as the first parameter. T1 indicates that the tape is mounted on tape drive 1. T2 indicates that the tape is mounted on tape drive 2. This parameter is not valid for DUMP DISK.

TC specifies that the DUMP procedure is to print the dump information from a tape in the tape cartridge drive. TC is valid only if TAPE is specified as the first parameter.

file name: If F1 is specified, this parameter specifies the file to be used by the DUMP procedure to display or print information. The file must have been created by a task dump or by the TRACE procedure. Dump files are named #DUMP.nn on disk, where nn is a number from 00 through 99. If no file name is specified, the DUMP procedure uses the following table to obtain information to be listed:

Parameter	Input Assumed if No File Name Specified
MAIN	If started by EVOKE OCL statement or run from job queue, the most recent task dump file is assumed; otherwise, the status of all dump files is displayed. You must select one of the dump files to continue.
CONTROL	The control storage library.
JOBQ	The system job queue.
MCODE	The control storage library.
PTF	The system library PTF log.
SERVLOG	The system service log.
SPOOL	The system spool files.
TRACE	If started by EVOKE statement or run from job queue, the system trace file is assumed. If DUMP procedure is run from the keyboard, the status of all task dump files that contain trace files, followed by a list of all trace files within the selected task dump file, is displayed. You must select one of the files to continue.
TWA	The system task work area.
VTOC	The system disk VTOC.
PLD	The system product level data file.

If I1 is specified, the DUMP procedure creates a resident disk file named **file name**, and copies the APARFILE file from diskette. If a disk dump file exists with the specified file name, that dump file is removed and the new dump file is created. If the existing disk file is not a dump file, a message is displayed and you can either replace the disk file with the new dump file or cancel the DUMP procedure.

library name specifies the library whose PTF information is to be listed. PTF must be specified for parameter 1. If I1 is specified, the PTF library information for the system library is listed from the diskette file. If no library name is specified, the system library is assumed.

0: If F1 is specified, 0 (zero) indicates that DUMP MAIN, DUMP STATUS, or DUMP TRACE is to use the most recent dump file. If no parameter is specified and the DUMP procedure is placed on the job queue, 0 is assumed.

If F1 is specified for DUMP MAIN and this parameter is omitted, you are shown the status of each task dump file on the system; you then choose the one to process.

S1, S2, or S3 specifies the diskette slot containing the first diskette from which members are to be processed. If no parameter is specified, S1 is assumed. For DUMP DISK, only S1 is valid.

M1.nn or M2.nn specifies the magazine location containing the first diskette from which members are to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.

- If M1.nn or M2.nn is specified, only the specified magazine can be used. Processing begins with the diskette at the specified location and continues through the last diskette in the magazine. If more diskettes are needed, a message is displayed and the operator must insert the next magazine in the magazine slot being used. Processing resumes at the first location in the magazine.

Example

This example shows how to print the dump information about main storage. The information is contained in a disk dump file named #DUMP.03, which was created because of a task dump.

```
DUMP ,,,#DUMP.03
```

ERAP Procedure

The ERAP procedure displays or prints data that was logged for the devices on the system. Depending on the device, the logged data is contained in one or more of the following tables:

- An input/output counter table that contains accumulated statistics reflecting the amount of activity for the device. (For example, the number of verify, write, read or scan read, and nonzero seek operations on a disk drive.)
- An error counter table that contains accumulated totals of specific types of errors for the device.
- An error history table that contains a data field, a time field, a system reference code (SRC) representing an error on the device, and status bytes that provide more detail about the error. The first entry in the table represents the most recent entry. After the table is filled, the oldest entry is dropped from the table each time a new entry is added.

Besides printing or displaying the logged information, the ERAP procedure allows you to reset the information in the input/output counter tables and the error counter tables.

When the ERAP procedure begins, it displays a series of menus from which you can select:

- The reports, device, or devices, for which logged information is to be displayed or printed
- The logged information to be displayed or printed
- The printer where the output is to be printed (if printing is selected)
- The time period for which logged information is to be printed or displayed

The ERAP procedure runs the $FEAIDS utility program.

```
ERAP
```

S9020456-0

The ERAP procedure has no parameters.

Example

This example shows how to start the ERAP procedure.

```
ERAP
```

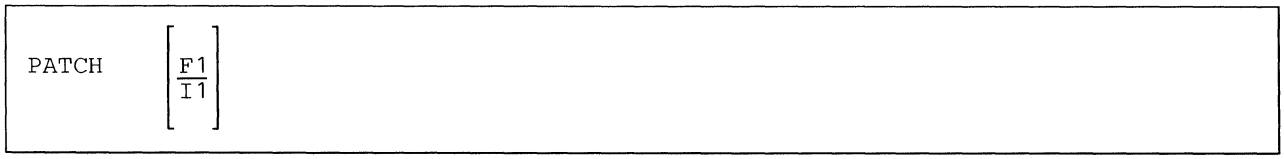
PATCH Procedure

The PATCH procedure displays selected disk or diskette sectors and allows you to modify the data in those sectors.

The PATCH procedure prompts you for the sector number of the sector you want to display. The sector number can be specified in hexadecimal or decimal. When you display diskette sectors, you can also specify the cylinder number, head number, and record value in hexadecimal or decimal format. The first 256 bytes of the selected sector are then displayed along with the sector address. Other portions of the disk or diskette can be displayed by entering a new sector address or by using the function keys to scroll the disk or diskette storage data. You can modify disk or diskette information by replacing data displayed on the display with new data.

Note: You should proceed with caution when using the PATCH procedure because it allows you to modify several critical system data areas.

Because PATCH allows you to change disk data, PATCH can be run only from the system console if password security is not active. If password security is active, the procedure can be run from any display station, but running the PATCH procedure is restricted to operators who have service aid authority. For more information about service aid authority, see the *System Security Guide*.



S9020457-0

F1 specifies that the disk is to be used. If a parameter is not specified, F1 is assumed.

I1 specifies that a diskette is to be used. If the system has a diskette magazine drive, the diskette must be in diskette slot S1.

Example

This example shows how to use the PATCH procedure to modify one or more diskette sectors. The diskette is in slot S1.

```
PATCH I1
```

PTF Procedure

The PTF procedure allows you to apply program temporary fixes (PTFs) to a specified library. The PTF procedure allows you to do the following:

- Copy PTFs from a PTF diskette to a PTF library.
- Apply PTFs from a PTF library to a specified library.
- Apply PTFs directly from a PTF diskette to specified library.
- List the PTF log for a specified library.
- Remove a PTF from a specified library.
- Save a PTF backup library on diskette.
- Restore a PTF backup library from diskette.
- Delete a PTF backup library from diskette or disk.
- Patch a library member on disk.
- Copy a microcode PTF to a PTF library.
- Apply a microcode PTF from a PTF library to the control storage library.
- Remove a microcode PTF.
- Restore the PTFNEWS library from diskette.
- Delete the PTFNEWS library from disk.
- Print or display one or more PTF newsletters in the PTFNEWS library.

PTF diskettes must be processed in ascending order by volume number. For example, volume 01 must be inserted first, followed by volume 02, until the last diskette is processed.

The suggested steps to be followed when applying a PTF to your system are:

- PTF COPY
- PTF APPLY
- PTF SAVE
- PTF DELETE

Notes:

1. ***PTF APPLY** requires a dedicated system with no tasks active.*
2. *If password security is active, the operator running **PTF APPLY** must have service aid authority.*
3. *If the PTF function used updates #LIBRARY or the control storage library, an IPL from disk will be started.*

For further information about the PTF procedure, see the manual *Operating Your System* for your system unit.

To copy PTFs for the SSP or IBM-supplied program products to a disk PTF library:

```
PTF     COPY,[OLD          ],[ptf type      ],[ptf library name],
             [ALL          ] [ALLPTF        ]
             [ptf log number] [ptf file name]

        [CHECK  ],[S1  ],[AUTO  ],[work library size],[directory size]
        [NOCHECK] [S2  ] [NOAUTO]
                  [S3  ]
                  [M1.nn]
                  [M2.nn]
```

S9020458-1

To apply PTFs from a PTF library to the SSP or a program product library:

```
PTF     APPLY,,[ptf type        ],[target library name]
               [ALLPTF          ]
               [ptf library name]
```

S9020459-0

To directly apply PTFs from a PTF diskette to a specified library:

```
PTF     DIRECT,[OLD          ],[ptf type      ],[target library name],
               [ALL          ] [ALLPTF        ]
               [ptf log number] [ptf file name]

               [S1  ],[AUTO  ]
               [S2  ] [NOAUTO]
               [S3  ]
               [M1.nn]
               [M2.nn]
```

S9020460-0

Note: The DIRECT parameter on the PTF procedure does not provide a backup of the programs being changed; therefore, any PTFs installed using this parameter cannot be removed using PTF REMOVE.

If you remove PTFs that were installed using PTF DIRECT, reinstall the system microcode, the SSP, and all the installed program products.

To list PTF information about a library:

```
PTF      LIST,target library name,[ PRINTER ]
                                  [ CRT     ]
                                  [ printer id ]
```

S9020461-0

To remove a PTF from the SSP or a program product library:

```
PTF      REMOVE,[ ALL           ],[ ptf type       ],[ backup library name ]
                [ ptf log number ] [ ALLPTF         ]
                                   [ target library name ]
```

S9020462-0

To save a PTF backup library on diskette:

```
PTF      SAVE,[ ptf type       ],volume id,[ S1    ],[ AUTO   ]
              [ ALLPTF         ]           [ S2    ]  [ NOAUTO ]
              [ backup library name ]      [ S3    ]
                                           [ M1.nn ]
                                           [ M2.nn ]
```

S9020463-0

To restore a PTF backup library from diskette:

```
PTF      RESTORE,[ ptf type       ],[ A1 ],[ S1    ],[ AUTO   ]
                 [ ALLPTF         ]  [ A2 ]  [ S2    ]  [ NOAUTO ]
                 [ backup library name ] [ A3 ]  [ S3    ]
                                         [ A4 ]  [ M1.nn ]
                                                 [ M2.nn ]
```

S9020464-0

To delete a PTF backup library from disk or diskette:

```
PTF      DELETE,[ptf type          ],[F1],[S1      ]
                [ALLPTF            ]     [I1]  [S2      ]
                [backup library name]           [S3      ]
                                                [M1.nn   ]
                                                [M2.nn   ]
```

S9020465-0

To patch a library member on disk:

```
PTF      PATCH
```

S9020466-0

To copy microcode PTFs to a PTF library:

```
PTF MCOPY,[CSPTFLIB          ],[S1      ],[AUTO  ],
          [PTF library name  ]  [S2      ]  [NOAUTO]
                                [S3      ]
                                [M1.nn   ]
                                [M2.nn   ]

    [PTF library size],[directory size]
```

S9020467-1

To apply a microcode PTF from a PTF library to the control storage library:

```
PTF MAPPLY,[ALL           ],[CSPTFLIB          ]
           [PTF log number]  [PTF library name]
```

S9020468-0

To remove a microcode PTF:

```
PTF MREMOVE,PTF log number,[CSPTFLIB          ]
                           [PTF library name]
```

S9020469-1

To process PTF newsletters:

```
PTF   NEWS,[ALL      ],[DISPLAY],[S1  ],[AUTO  ]
           [RESTORE  ]  [D      ]  [S2  ]  [NOAUTO]
           [DELETE   ]  [PRINT  ]  [S3  ]
           [PTFINDEX ]  [P      ]  [M1.nn]
           [PTFXREF  ]             [M2.nn]
           [BULLETIN ]
           [name     ]
```

S9020587-2

COPY specifies that PTFs are to be copied from a PTF diskette to a PTF library. The COPY function creates a PTF library. This library contains the PTFs that are to be applied to the specified library, for SS1 for example, the library would be the system library, #LIBRARY. The name of the PTF library is PTFxxxxx, where xxxxx is the first 5 characters of the diskette file name; for example, for SS101, the name would be PTFSS101.

APPLY specifies that PTFs are to be applied from a PTF library to a specified library. The PTF library must have been created using the COPY function. The modules receiving PTFs are copied to a PTF backup library. The backup library contain the modules that are being replaced by PTFs; that is, the backup library contains down-level modules. The PTF modules are copied to the specified library; for example, for SS101, the system library.

The down-level modules remain in the PTF backup library until a new set of PTFs are applied to the specified library. The PTF backup library is named PTBxxxxx, where xxxxx is the last 5 characters of the PTF library name; for example, for PTF library PTFSS101, the backup library is named PTBSS101.

If there is not enough room in the target library for the modules to be applied, the CONDENSE procedure is run for the target library before the modules are moved. #LIBRARY is always condensed, even if there is enough room.

DIRECT specifies that PTFs are to be applied directly from a PTF diskette to a specified library.

LIST specifies that the PTF log (#PTFLOG) for a library is to be printed or displayed.

REMOVE specifies that PTFs are to be removed from a specified library. The down level modules are copied from the PTF backup library to the specified library, replacing the PTF modules.

SAVE specifies that a PTF backup library is to be saved on diskette.

RESTORE specifies that a PTF backup library or PTFNEWS library is to be restored from diskette.

DELETE specifies that a PTF backup library or PTFNEWS library is to be deleted from disk or diskette.

PATCH specifies that a library or library member can be directly modified using the display station. If password security is active, PTF PATCH can be run from any display station, but only by an operator that has service aid authority.

PTF PATCH runs the $FEFIX utility program. See the manual *Program Problem Diagnosis and Diagnostic Aids* for information about the control statements required by this utility.

MCOPY specifies that the microcode PTFs are to be copied from the PTF diskette to the microcode PTF library (CSPTFLIB).

MAPPLY specifies that the microcode PTFs are to be applied from the PTF library (CSPTFLIB) to the control storage library.

MREMOVE specifies that the microcode PTF is to be removed from the control storage library. The removed PTF is in the PTF library (CSPTFLIB).

NEWS specifies that the PTFNEWS library is to be processed.

OLD specifies that PTFs are only to be processed for existing library members. If no parameter is specified, OLD is assumed.

ALL specifies that all PTFs are to be processed.

ptf log number specifies that only the PTF corresponding to the number entered is to be processed. This number specifies the PTF log number, which is indicated in the source member named PTFXREF on each diskette.

ALLPTF specifies that all PTFs that are to be used with the current system configuration are to be processed. For example, if your system had the SSP, the Utilities, RPG, COBOL, and BASIC, only those PTFs for those licensed programs would be processed. Any other PTFs, such as FORTRAN, would not be processed.

ptf type specifies the type of PTF to be processed. The following table lists the PTF types with the associated library and a description of the library. **nn** is the SSP release level. For example, to process PTFs for release 2 of the SSP, you would enter SS102.

PTF Type	Library	Description
SS1nn	#LIBRARY	System Support Program
AP1nn	#APFLIB	Advanced Printer Function
AS1nn	#ASMLIB	Assembler Program Product
BA1nn	#BLLIB	BASIC Program Product
BH1nn	#BLHPLIB	BASIC Help Support
BNHnn	#BGUHLIB	Business Graphics Help Support
BNWnn	#BGULIB	Business Graphics Utility
BRJnn	#POPLIB	Programmer & Operator Productivity Aid (POP)
CB1nn	#COBLIB	COBOL Program Product
DF1nn	#DFULIB	Data File Utility (DFU)
DS1nn	#DSULIB	Development Support Utility (DSU)
EP1nn	#LIBRARY	3278 Emulation by IBM PC
FO1nn	#FORTLIB	FORTRAN Program Product
IG1nn	#CGULIB	Ideographic Support
IS1nn	#SRTXLIB	Ideographic Sort Utility
IW1nn	#IWLIB	PC Support/36
LC1nn	#LIBRARY	Local Area Network (LAN)
MA1nn	#MIGRLIB	S/34 to S/36 Migration Aid
OLPnn	#ONLPD	Online Problem Determination
QU1nn	#QRYLIB	Query/36
RG1nn	#RPGLIB	RPG II Program Product
SD1nn	#SDALIB	Screen Design Aid (SDA)
SE1nn	#SEULIB	Source Entry Utility (SEU)
TX1nn	#TMSLIB	Text & Office Management Systems (TMS & OMS)
TXSnn	#LEXLIB	Text Editor Languages
WP1nn	#TULIB	DisplayWrite/36
WP3nn	#OFCLIB	Personal Services/36
WS1nn	#WSULIB	Work Station Utility (WSU)

When APPLY or DIRECT are specified, PTFs for SS1nn or ALLPTF can only be processed when:

- No other jobs are being run
- No other users are signed on at any other display stations
- Remote work stations are offline
- Interactive communication feature (SSP-ICF) sessions are disabled
- Communications lines are offline

ptf file name specifies the name of the file on the PTF diskette that contains the PTFs to be copied and applied to the system.

ptf library name specifies the name of the library that is to contain the PTFs. This library name is automatically determined by the type of PTFs that are being copied or applied. However, entering a value for this parameter overrides the automatic naming process.

target library name specifies the name of the library to which PTFs will be applied. This library name is automatically determined by the type of PTFs that are being copied or applied. However, entering a value for this parameter overrides the automatic naming process.

backup library name specifies the name of the library that will contain copies of each module that PTFs were applied to.

PTFINDEX specifies that an index is displayed or printed.

PTFXREF specifies that a PTF cross reference listing is displayed or printed.

BULLETIN specifies that a special bulletin in the PTFNEWS library is displayed or printed.

name specifies that the named PTF description is displayed or printed. The **name** parameter must be in the form of PTFxxxxx, where xxxxx is a PTF log number.

DISPLAY or D specifies that the information is displayed on the CRT.

PRINT or P specifies that the information is printed.

CHECK specifies that prerequisite level checking is to be done as the diskette file is being copied to the PTF library on disk. If a PTF being copied has a prerequisite PTF that is not already on the system, it must also be copied into the PTF library, before applying the PTFs to the specified library.

NOCHECK specifies that no prerequisite or release level checking is to be performed.

A1 specifies that the preferred disk file location is on the first disk. If space is available, the PTF backup library is placed on the first disk; otherwise, the library is placed on the second disk.

A2 specifies that the preferred disk file location is on the second disk. If space is available, the PTF backup library is placed on the second disk; otherwise, the library is placed on A1 if you have two or four disk drives, or on A3 if you have three disk drives.

A3 specifies that the preferred disk file location is on the third disk. If space is available, the PTF backup library is placed on the third disk; otherwise, the library is placed on A2 if you have three disk drives, or on A4 if you have four disk drives.

A4 specifies that the preferred disk file location is on the fourth disk. If space is available, the PTF backup library is placed on the fourth disk; otherwise, the library is placed on A3.

S1, S2, or S3 specifies the diskette slot containing the first diskette from which members are to be processed. If no parameter is specified, S1 is assumed. For the DELETE function, this indicates the only diskette to be processed, no other slots are processed.

M1.nn or M2.nn specifies the magazine location containing the first diskette from which members are to be processed. M1 indicates the first magazine, and M2 indicates the second magazine. For the DELETE function, this indicates the only diskette to be processed, no other magazine slots are processed. **nn** is a number from 01 through 10 that identifies the location of the diskette in the magazine. Specifying M1 is the same as specifying M1.01; specifying M2 is the same as specifying M2.01.

AUTO specifies the following:

- If S1, S2, or S3 is specified, all three individual slots (S1, S2, and S3) can be used. Processing begins with the diskette in the specified slot and continues through the diskette in slot S3. If more diskettes are needed, a message is displayed and the operator must then insert the next diskettes. Processing resumes with the diskette in slot S1.

- If M1.nn or M2.nn is specified, both magazine slots (M1 and M2) can be used. Processing begins with the diskette at the specified location and continues through the diskette in location M2.10. If more diskettes are needed, a message is displayed and the operator must then insert the next magazines. Processing resumes at location M1.01.

If no parameter is specified, AUTO is assumed.

NOAUTO specifies the following:

- If S1, S2, or S3 is specified, only the specified slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the specified slot.
- If M1.nn or M2.nn is specified, only the specified magazine slot can be used. If more diskettes are needed, a message is displayed and the operator must insert the next diskette in the magazine slot being used. Processing resumes at the magazine location specified.

work library size specifies the size (in blocks) of the temporary library that is used in the COPY process. If no parameter is specified, the necessary value is calculated. The calculated value will be at least 50 blocks.

directory size specifies the size (in sectors) of the directory for the temporary library. If no parameter is specified, 2 sectors will be allocated for every block.

Example 1

This example shows how to apply PTFs for release 1 of the SSP. The PTF COPY copies all the PTFs from the PTF diskette to a PTF library on disk; the library will be named PTFSS101. The PTF APPLY actually applies the PTFs to the SSP. The APPLY function also copies SSP members that are receiving PTFs to a PTF backup library, named PTBSS101.

```
PTF COPY,OLD,SS101
PTF APPLY,,SS101
```

Example 2

This example shows how to print the PTFs that have been applied to the system library.

```
PTF LIST,#LIBRARY
```

SERVICE Procedure

The SERVICE procedure displays a menu that allows you to do the various tasks you may be asked to do for problem determination and correction. These tasks include:

- Running the service aid procedures for determining program related problems (such as: DUMP, APAR, DFA, and TRACE)
- Running the service aid procedures for determining hardware related problems (such as: ERAP)
- Copying and applying program temporary fixes (PTFs) to the licensed programs
- Adding entries to or listing the system service log
- Running the PATCH procedure to display or change disk or diskette sectors

```
SERVICE
```

S9020452-0

The SERVICE procedure has no parameters.

Example

This example shows how to enter the SERVICE procedure to display the SERVICE menu.

```
SERVICE
```

SERVLOG Procedure

The SERVLOG procedure allows you to add entries to the system service log. The SERVLOG procedure can be run from any display station.

To display or print entries in the service log, see the "DUMP Procedure" on page C-7.

```
SERVLOG  'message text to log'
```

S9020470-0

message text to log specifies the entry to be made into the service log. The text must be enclosed by apostrophes (').

Example

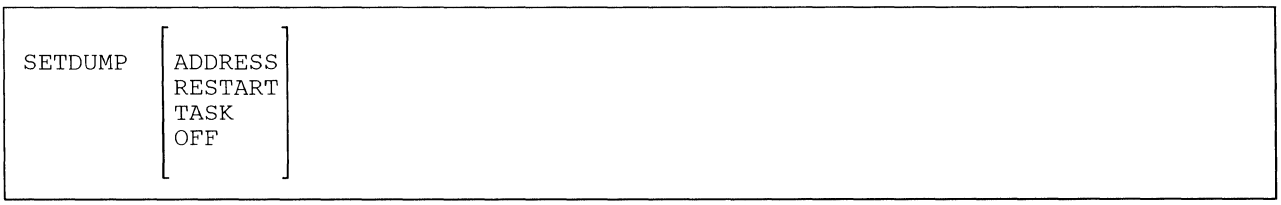
This example shows how to add the entry `NEW KEYBOARD PLACED ON SYSTEM CONSOLE` to the system service log.

```
SERVLOG 'NEW KEYBOARD PLACED ON SYSTEM CONSOLE'
```

SETDUMP Procedure

The SETDUMP procedure allows you to debug programs running in main storage at predetermined break points or addresses without having to stop the main storage processor. The SETDUMP procedure allows you to take a task dump of your program when an address in your program is referenced.

Dump files are named #DUMP.nn on disk, where nn is a number from 00 through 99.



S9020471-0

If no parameters are specified, a menu is displayed and you can choose the task you want.

ADDRESS specifies that the prompt used to set the address compare values is to be displayed. A job that was previously suspended by the address compare dump facility will be resumed.

Note: On a 5360 Model D System Unit, always use the verify option of the ADDRESS parameter when you specify a real main storage address of 000000 through 0007FF.

RESTART specifies that the address compare dump facility is to be enabled to allow another address compare dump using the previously defined address values. A job that was previously suspended by the address compare dump facility will be resumed.

TASK specifies that a task dump is to be taken immediately for a specified task. A job that was previously suspended by the address compare dump facility will be resumed.

OFF specifies that the address compare dump facility is to be disabled. A job that was previously suspended by the address compare dump facility will be resumed.

Example

The following example shows how to enter the SETDUMP procedure to define address compare dump values.

```
SETDUMP ADDRESS
```

TRACE Procedure

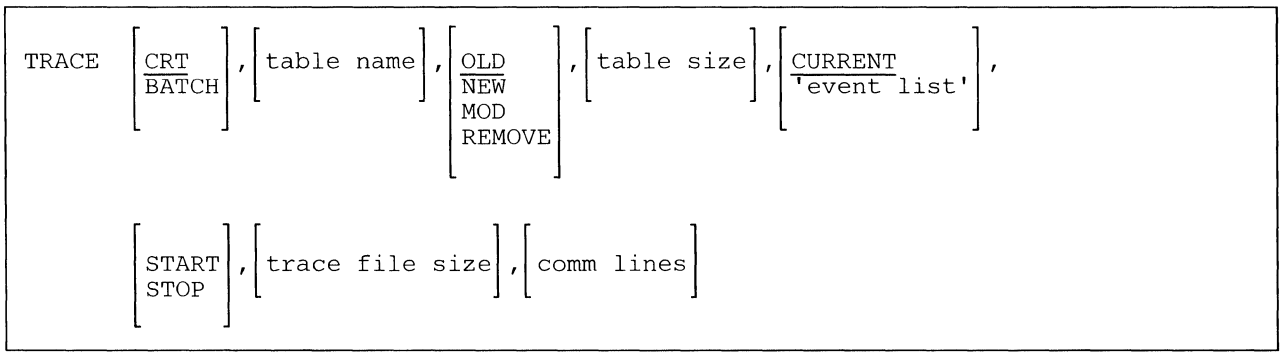
The TRACE procedure allows you to keep a history of events that occur in the system. Selected system events are recorded as they occur in a variable-length, wrap-around table in main storage. Only one copy of the utility may be active at a time.

The TRACE procedure can be run in two different ways: 1) The procedure can be run interactively; that is, the procedure will display the status of each trace table and prompt for any changes. 2) The procedure can be entered with parameters, allowing the procedure to be run from the job queue, run from another procedure, or even be one of the IPL start up procedures.

From 16 through 512 entries can be traced in the main storage trace table. Alternate trace tables can also be created or deleted. You can select tracing for any of the following events:

- Any one or a combination of main storage supervisor calls (SVCs)
- Control storage processor supervisor calls
- Task dispatches
- Task swaps
- Disk input/output operations
- Component related events, such as:
  - Work station utility (WSU) functions
  - Remote work station functions
  - Asynchronous communications support
  - BSC
  - SNA
  - SDLC
  - X.25
  - MSRJE
  - SSP-ICF
  - APPC
  - LAN

You can also select disk logging options, and if you have selected any communications related events, you can select the communications lines to be traced.



S9020472-0

If no parameters are specified, TRACE will prompt you for the parameter values.

CRT specifies that the procedure is to be run interactively. You will be prompted for certain TRACE parameter values. If this parameter is not entered, CRT is assumed.

BATCH specifies that all parameter values have been specified, whether they be the default or something different.

table name specifies the name of the trace table to which events will be traced. If no name is specified, the main trace table is assumed.

OLD specifies that the trace table already exists. If the table cannot be found, then an error message is displayed. If this parameter is not specified, OLD is assumed.

NEW specifies that the table named is a new alternate trace table. If a table already exists with this name, an error message is displayed.

MOD specifies that if the table already exists, it will be changed according to other parameters specified. If the table does not exist, then a trace table will be created with the values of the other parameters specified.

REMOVE specifies that the alternate trace table named is to be deleted. If logging to disk is active, then logging is stopped before the table is deleted. The trace disk file is not deleted. The main trace table cannot be deleted.

table size specifies the maximum number of entries the trace table can hold. The valid number of entries that can be logged to a table are 16, 24, 32, 40, 48, 56, 64, 128, 192, 256, 320, 384, 448, and 512. Any other number within the range of 16 through 512 can be specified, but will be rounded up to the next valid number. If this parameter is not specified, and the table is a new table, the table size of 16 entries is assumed.

CURRENT specifies that the events previously selected for tracing will be traced. If CURRENT is specified for a new trace table, an error message is displayed.

'event list' specifies the events or tracing profiles to be traced. If more than one event or profile is to be traced, then the values for this parameter must be enclosed within quotes and the events or profile names must be separated by commas. The total length of the parameter cannot exceed 82 characters including the commas and quotes. Any event listed using the interactive method of this procedure can be specified. If you want to remove events being traced by a trace table, enter a minus sign (-) in front of the event to be removed. Putting a minus sign in front of a profile does not remove the profile from the trace table, but will remove the events that the profile represents. Events are added to or removed from the trace table profile in the order that they occur in the list.

START specifies that the trace table should start logging to disk. The trace table name must be different from the name of an existing user disk file name or an error message is displayed.

STOP specifies that the trace table should stop logging to disk.

trace file size specifies the size of the trace file in blocks. Each block of the file will hold approximately 80 trace entries. If this parameter is not specified and logging to disk is started, and no trace file exists, then a 10-block trace file is created.

comm lines specifies for which communications lines events are to be traced. If more than one line number is specified, they must be enclosed in quotes (') and separated by commas. For example, to trace lines 1 and 3 specify '1,3' or '3,1'. All communications lines are traced if the listed events or profiles require selecting a communications line, **comm lines** is not specified, and no previous lines were selected.

Appendix D. Multinational Character Set Conversion Utility

Introduction

The multinational character set (MCS) feature is a System/36 utility that makes available an expanded multinational character set of 188 characters for most language groups in countries where the System/36 is available. The set includes 112 alphabetic characters, 10 numeric characters, and 66 special characters. Some characters of the expanded character set have been assigned different hexadecimal values than they had in the past. Because of these changes in hexadecimal values, you may need this conversion utility to change data in the national language version (NLV) value to the MCS value so that printed or displayed characters correspond to the proper MCS value.

The multinational character set conversion utility (MCSCU) helps convert library source members, library procedure members, data files, text folders, and data dictionaries from the national language hexadecimal value to the MCS hexadecimal value, and vice versa.

See "Changed Characters by Language Group" on page D-39 for a description of specific language group characters affected by the multinational character set

The MCSCU programs and procedures:

- Prompt for necessary information to run the utility
- Convert library members in a work file and place the results in the same or another specified library
- Print an audit list of examined members
- Convert data files, text folders, and data dictionaries
- Print a list of affected statements or records (optional)
- Modify statements or records (optional)
- Count affected statements or records in a member or a file (optional)

When you are using the multinational character set conversion utility, you must specify the type of records being converted: source statements, procedure statements, data files, text folders, or data dictionaries. The types of statements in library members that are subject to conversion include RPG, COBOL, FORTRAN, assembler, sort, DFU, message member, display formats, WSU, OCL, and menus.

For information about installing MCSCU, see the manual *Changing Your System Configuration.*

5250 Hardware Support

Besides the standard System/36 devices, the MCSCU supports models of the 5250 Display Stations that have the MCS feature. Installing the 5250 Multinational Character Set feature requires converting any internally stored or saved character data that has a different hexadecimal value in the MCS.

The MCS hardware feature does not have to be installed on the system to run MCSCU. But if it is not, all features may not be accessible; for example, keying the new hexadecimal values for manual changes, keying user-supplied alternate conversion tables, or printing new characters.

Starting the Conversion Utility

On the System/36 command display, enter the following procedure command:

```
MCSCONV
```

There are no parameters.

You are prompted for the type of conversion work you want to perform; working either with library members, data files, text folders, or data dictionaries. The display is shown in Figure D-1.

```
                                MCSCU

Work with the multinational character set conversion utility (MCSCU).


Select one of the following:

   1. Convert source or procedure members
   2. Convert data files
   3. Convert document folders
   4. Convert data dictionaries
   5. End the conversion utility

Option:  _




Cmd3-Previous menu            Cmd7-End                  COPR IBM Corp. 1986
```

Figure D-1. Conversion Option Display

If you select option 1, you are prompted for information necessary to process library members. See "Library Member Conversion" on page D-4.

If you select option 2, you are prompted for information necessary to process a data file. See "Data File Conversion" on page D-18.

If you select option 3, you are prompted for information necessary to process a document folder. See "Document Folder Conversion" on page D-23.

If you select option 4, you are prompted for information necessary to process a data dictionary. See "Data Dictionary Conversion" on page D-27.

To end the conversion utility, select option 5.

Library Member Conversion

Basic components of any installation are the application programming resources. These resources take on four forms in a library: procedure members, source members, subroutine members, and load members. Source members in turn can be categorized into different types: RPG, COBOL, FORTRAN, assembler, sort, DFU, message members, display formats, WSU, and menu members.

Each type of source statement may contain characters in the form of constants, display prompts, printer headings, and so on, that have taken on a different hexadecimal value under MCS. When a program is run on a system with the MCS feature, display formats and printed reports show the character associated with the MCS value instead of the national language version character. MCSCU helps you in finding characters within source statements and procedures that are MCS sensitive. The word **sensitive** means that a character match has been found in the record being scanned with a from character in the conversion table being used. Depending on where that character exists in a record, MCSCU may or may not be able to modify it. MCSCU cannot do a complete automatic conversion of the entire record in all types of library members. For example, if you have the following OCL statement in a procedure:

```
// LOAD $ABC
```

The dollar sign ($) is an MCS sensitive character (in some languages), but it cannot be automatically converted because $ABC is the name of the program. Another example is Interactive Communications Feature format names that begin with $$ and must not be changed. You must examine the statements that were found to be sensitive to determine if a manual change is desirable or required. MCSCU does not process load members. Load members are updated when the associated source is recompiled.

MCSCU copies user-selected source or procedure members to a work file, does the analysis and updates on the work file, and then copies the members in the work file back to the specified output library.

You must select which library members are to be converted. The library members selected must all be of the same type; for example, RPG, display formats, or procedures. Do not mix member types in the work file. To help you in selecting members and creating the work file, MCSCU prompts you for the necessary information. By responding to the prompts in Figure D-2 you can:

- Get a directory listing of a specified library from which to select members.
- Copy individual members or groups of members from a library to the work file.
- Run the MCSCU against a work file previously created.

You should use the directory list (option 1, Figure D-2) to get a listing of all procedure or source members in the specified library. If the System/36 MCS feature is installed, characters other than what the user is expecting to see may appear in those member names. If new characters do appear, you must enter those values when selecting the members.

It is a good practice to save your original source and procedures on diskette as a precautionary measure. See the "SAVELIBR Procedure" on page 4-431 or the "Copy Members from a Library (FROMLIBR Procedure)" on page A-66 for information about saving libraries or library members.

Figure D-2 is the prompt display for doing the selection process. An explanation of each prompt and option follows the display.

You first copy the members to be converted to a work file. After you have copied all desired procedure members or source members of a specific type, you should select option 3 or 4 to run the conversion utility.

```
                CONVERT SOURCE OR PROCEDURE MEMBERS



Select one of the following:
   1. Library directory        4. Run
   2. Copy                     5. End
   3. Copy and run
Option . . . . . . . . . . . . . . . . . . . . . . . . . . . . . _

Name of member, partial name of a group of members
  or to list all members, enter ALL . . . . . . . . . . . . . . . ________

  If partial name is entered, enter ALL . . . . . . . . . . . ALL  ___

Member type . . . . . . . . . . . . . . . . . . . . . . . . . P,S  P

Input library name  . . . . . . . . . . . . . . . . . . . . . . . ________

Work file name  . . . . . . . . . . . . . . . . . . . . . . . . . ________

Number of blocks for work file  . . . . . . . . . . . . 1-65535  100

Cmd3-Previous Menu            Cmd7-End
```

Figure D-2. Library Member Copy and Convert Display

Select one of the following specifies what type of function is to be performed. You can enter 1, 2, 3, 4, or 5.

1 Lists the source or procedure member names in the specified library. This directory listing can be used to select members for conversion. After the listing, the library member copy prompt display is reshown.

2 Copies selected members to the specified work file. It does not immediately run the conversion utility. After the copy, the library member copy prompt display is reshown to allow you to enter more members to copy to the work file.

3 Copies selected members to the specified work file and runs the conversion utility. See "Library Member Modification" on page D-7.

4 Runs the conversion utility using the specified work file. See "Library Member Modification" on page D-7.

5 Ends the library member conversion display.

Name of member, partial name of group of members allows you to enter a specific member name or a partial name of a group of members to be copied. This entry is required for options 2 and 3. If ALL is entered, all procedures or source members in the library are copied to the work file. You must ensure that all source members in the library are of the same type if ALL is entered.

If partial name is entered, enter ALL: You must enter ALL to get a group copy. Anything other than ALL is ignored. This parameter is ignored if ALL is entered for a specific member or partial name parameter.

Member type specifies whether source or procedure members are to be copied to the work file. This entry is required for options 1, 2, and 3.

P indicates a procedure member.

S indicates a source member.

Input library name specifies the library from which the members are to be copied. The default is the session library. This entry is required for options 1, 2, and 3.

Work file name specifies the work file that will, for options 2 and 3, contain the copied members. For options 3 and 4, this specifies the work file that contains library members to be analyzed and processed. If the file already exists, additional members are copied to it. This entry is required for options 2, 3, and 4.

Number of blocks for work file specifies the size of the work file in blocks if the file needs to be allocated. The default is 100 blocks. Each block can accommodate about 21 records.

If the work file is too small, it is automatically extended when it becomes full.

Library Member Modification

After you have selected the library members to be converted and have copied them to the MCSCU work file, you must specify information specifically related to how the work file should be analyzed. The information needed includes:

- What type of source or procedure members are in the work file.
- Which library should the modified members be copied back to.
- Which national language conversion table should be used.
- What is the direction of the conversion.
  - National language to MCS
  - MCS to national language
- What type of analysis options are desired.

You are prompted for this additional analysis information by the display shown in Figure D-3. A description of each of the types of analysis information follows the display.

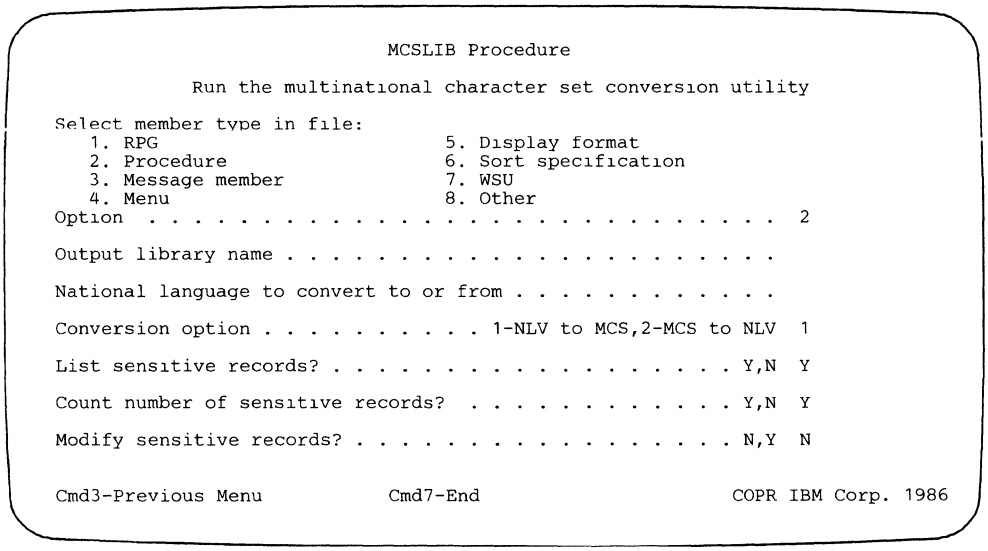


Figure D-3. Library Member Analysis Information

Select member type in file: Indicates the type of library members contained in the work file. A number from 1 through 8 can be entered. This option does the list, count, and modify functions (described later) that were requested for all characters in each record. This entry is required. The default is 2, procedure member.

1 RPG source member.

2 Procedure member.

3 Message source member.

4 Menu source member created for the BLDMENU procedure.

5 Display format source member. Also, menus created by the screen design aid (SDA) utility.

6 Sort specifications source member.

7 WSU source members.

8 Some other source member (COBOL, FORTRAN, and so on) that is not supported by a specific member type option.

Note: While doing the copy function, MCSCU did not need to know the type of source member being selected. But in order for MCSCU to do the analysis of the members in the work file (being sensitive to the different statement types), you now must specify this information.

Output library name specifies the name of the library where the work file is to be copied after the conversion occurs. This copy function occurs only if you choose to modify sensitive statements. This is a required entry when you specify Y (yes) to the modify sensitive records prompt.

National language to convert to or from specifies the language group requested from the following list:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

MCSCU supplies the conversion tables for each national language. If you wish to supply your own table, you do so by completing the messages in the MCSCU message member (#MN#M2) and specifying OTHER (option 8). See "Conversion Tables" on page D-10 for additional information.

Conversion option MCSCU allows conversion from NLV to MCS by option 1, or MCS to NLV by option 2. This entry is required. The default is option 1.

List sensitive records? MCSCU lists the statements within a member that contain the sensitive characters. Option Y is list; option N is do not list. The default is Y. See "MCSCU Library Member Listing" on page D-11 for additional information.

Count number of sensitive records? MCSCU counts the number of statements that are sensitive in a given member and the total number for all members in the work file. Option Y is count; option N is do not count. The default is Y. See "MCSCU Library Member Listing" on page D-11 for additional information.

Modify sensitive records? MCSCU either modifies the statements as it scans them or leaves them as is. Option Y is modify; option N is do not modify. The default is N. If option Y is specified, DW/36 must be installed on your system. See "Library Statement Modification" on page D-13 for an explanation of what modify means for each member type.

MCSCU for library members allows you to:

1. Get an audit list of all members examined in the disk file.

2. Get a list of the specific statements in a source or procedure member that have MCS sensitive characters.

3. Count all the sensitive records in a member and the file.

4. Modify the source statements, based on the specific statement type.

Note: This conversion utility should not be run against any System/36 SSP or program product supplied message member, display format member, or procedure member.

If you indicate no for list, count, and modify, an error is issued.

The program that does the actual work file analysis and modification is an evoked program. This allows the actual conversion program to run as a separate job, and allows control to immediately return to the MCSCONV procedure to prompt for additional conversion activity. Thus it is possible to have several different work files being analyzed and modified at the same time. Because the analysis and modification of the work file is still in progress, no additional members should be copied to the work file, or the work file should not be deleted, until the evoked job has completed.

Conversion Tables

All conversion tables for the valid language groups are supplied in the MCSCU message member #MN#M1. These tables allow you to convert from a specific language group to MCS and vice versa. Two additional messages in the MCSCU message member #MN#M2 allow you to supply your own conversion tables for converting from and converting to languages. If you wish to supply your own tables:

- Modify the specified messages in the source message member named #MN#M2 using SEU or DSU (the member is contained in the system library, named #LIBRARY), or create a new #MN#M2 using $MAINT.
- Rebuild the MCSCU message member #MN#M2, using the following operation control language:

```
// LOAD $MGBLD
// RUN
// MGBLD SOURCE-#MN#M2,SSP-YES,REPLACE-YES,LIBRARY-xxx
// END
```

  where xxx identifies the library containing MCSCU support.

There are two messages in message member #MN#M2. Message number 0001 allows you to define the conversion table for going from NLV to MCS, and message number 0002 is used to convert MCS to NLV. You must enter contiguous pairs of from-to values. Message 0001, in the following example, maps ¢ to <, $ to {, and # to }.

```
0001 ¢<${#}
```

To translate back, message 0002 would be

```
0002 <¢{$}#
```

#MN#M2 is a second-level message member and allows 112 pairs of from-values and to-values. The first blank *from* position in the message is the end of the table.

For additional information on character conversion, see "3262 Print Belts" on page E-1.

Source Statement Length

You can have different source statement lengths in your library that vary based on the statement type. When MCSCU copies the members from the library to the work file, all statements will have a length of 120. This may expand your library space requirements.

MCSCU Library Member Listing

A sample listing produced by the MCSCU for a library member is as follows:

```
REC NUM    1.......10........20........30........40........50........60........70........80........90.......100.......110.......120-(B)

***    (A)-MEMBER NAME-MCSTEST ,    LIST-(Y),    COUNT-(Y),    MODIFY-(N),     DATE-10/24/82,       TIME-11:18:18

(C)-1 OLD First Line: ¢   (cent symbol)
(D)       C89AA D8987  4   4889A AA98995
          69923 3955A  A   D3553 284263D

  (E)-NEW              ¢
                       4
                       A
                       *

    2 OLD Second Line: |   (logical or symbol)
          E88998 D8987 4   49988889 99 AA98995
          253654 3955A F   D3679313 69 284263D

      NEW              |
                       4
                       F
                       *

    3 OLD Third Line:  !   (exclamation point)
          E8898 D8987  5   48A89898A899 9989A5
          38994 3955A  A   D57331413965 76953D

      NEW              !
                       5
                       A
                       *

    4 OLD Fourth Line: ¬   (logical not symbol)
          C9A9A8 D8987 5   49988889 99A AA98995
          664938 3955A F   D3679313 563 284263D

      NEW              ¬
                       5
                       F
                       *

***    (F)-MEMBER NAME-MCSTEST ,    COMPLETE,    COUNT-     4

          TOTAL SENSITIVE RECORD COUNT-     4
```

Figure D-4. Sample MCSCU Library Member Listing

Ⓐ An audit line containing the member name and the options in effect are listed with the sensitive statements in the member. LIST, COUNT, and MODIFY indicate Y or N, based on the response to the prompt display. The audit line is printed regardless of whether you selected the list option. The audit line is useful as a history of:

- Which members have been processed in the work file
- Which processing options were requested

For recovery, you can determine members needing to be processed by displaying the file and eliminating those on the audit list (after they have been copied to the output library). If the word PARTIAL appears instead of COMPLETE, this indicates the work file was not large enough to contain the entire member when the member selection copy was performed. A message will be issued by $MAINT when the work file is copied back to the library. You should recopy the partial member to a new work file and reprocess the entire member.

Ⓑ Column indicators help you determine the column in which the sensitive character was found.

Ⓒ The number of this statement within the member being processed.

Ⓓ OLD indicates the original statement. The statement is printed in both character and hexadecimal format.

Ⓔ NEW indicates that the statement contained sensitive characters. The sensitive characters have data printed below the original statement. The characters printed on the NEW line indicate where the sensitive data was found.

The new sensitive character is also shown. If you chose the option to not cause the sensitive statements to be modified, the old and new characters will be equal. If you chose the option to cause the sensitive statements to be modified:

- If the character is in a position that MCSCU is allowed to modify, the new substituted character is printed.
- If the character is in a position that MCSCU was not allowed to modify (for example, before the name of a program), the original character is printed.

The character (if the printer allows) and the hexadecimal value of the character are printed. An asterisk (*) is printed below each NEW line character that was sensitive but was not modified.

Ⓕ The number of statements in a member that were sensitive to MCS. It also shows all the statements in the entire work file that are sensitive to MCS.

Library Statement Modification

Based on the type of member being processed, MCSCU becomes sensitive to different types of specifications within a source member. Certain columns and fields can be changed and others can only be indicated with an asterisk (*). Each of the following sections describes the activity that is done for each specification type. If, within a specific member, a statement is encountered that does not fit the criteria for that source type, sensitive data is indicated but not modified. For example, a Z in column 6 of RPG is invalid and the statement would be scanned but not modified.

Note: MCSCU assumes that the members to be processed are valid in syntax and would be accepted by the System/36 SSP or program product that would normally be using them.

RPG Source Members

If columns 1 and 2 contain asterisks (**), then all statements for that member following the ** record are scanned but not modified. Two asterisks indicate compile time table or array data, alternate sequence, and file translation table.

If column 7 contains a single asterisk (*), all columns are scanned and sensitive characters are modified.

If column 6 contains an H:

- Columns 1 through 74 are scanned and sensitive characters are modified.
- Column 18 has the floating currency symbol filled in from the conversion table if (1) column 18 is blank; (2) the conversion table contains a hexadecimal 5B in a from comparison value; and (3) you are converting from NLV to MCS. When you are converting from MCS to NLV, column 18 will be blank. All NLV-to-MCS tables supplied with MCSCU have a hex 5B value when the to-value is other than hex 5B.
- Columns 75 through 80 are only indicated.

If column 6 contains an F:

- All columns are scanned.
- Data is only indicated.

If column 6 contains an I:

- Columns 27 through 41 are scanned and sensitive characters are modified.
- Other columns are only indicated.

If column 6 contains a C:

- All columns are scanned and indicated.
- Columns 18 through 27 and 33 through 42 are modified, if column 18 or 33 contains an apostrophe (') indicating a literal.

If column 6 contains a 0:

- All columns are scanned and indicated.
- Columns 45 through 70 are modified, if columns 40-43 do not contain a K followed by a number (right justified) from 1 through 8, which denotes a format name.

If column 6 contains an L, E, T, or U, the records are only scanned and indicated.

If, when starting in column 7, MCSCU detects a /COPY, /SPACE, /EJECT, or /TITLE, the statements are only indicated.

Procedure Members

All statements are indicated when an MCS sensitive character is detected. Only the following OCL statements are modified:

- Comment statements in a procedure are modified completely. A comment is denoted by an asterisk (*) in column 1.
- // *, // **, // MSG, and // PAUSE with text have the message text and any comments modified.
- For any statement not beginning with // , all columns are scanned and indicated.

Note: Because the work file is a record mode file, the $MAINT utility program has inserted a COPY statement before each member in the file and a CEND statement after each member. If you have a procedure that contains internal CEND statements, the internal CEND prematurely ends the procedure processing for that member. It also may cause incorrect audit line printing. Internal CEND statements are valid in user procedures only when FROM-READER is specified for $MAINT.

Message Members

All statements in a member are examined and processed in the following manner:

- All comments are scanned and modified.
- The control specification in the member is only indicated.
- All message definition statements are scanned and modified.

The assembler programmer should be aware that messages indicating character substitution by # using the message display utility ($CPPE) will have their contents changed if # has an alternate MCS value from NLV.

Menu Members

Menu members are assumed to have been created for the BLDMENU procedure. These menu members are treated the same as message members. If the command message member for a menu contains OCL, sensitive data is changed.

Menus can also be created by using the screen design aid (SDA) utility. The display text source members SDA on the System/36 creates are display format source members, not message source members. (Message source members are used by the BLDMENU procedure). To determine the type of source member used to create the menu, list the member by using the LISTLIBR procedure (see the "LISTLIBR Procedure" on page 4-272 for information on running LISTLIBR). If the member contains S- and D-specifications, it was created by SDA, and you should use option 5 to convert the member.

For more information about menus, see the manual *Creating Displays*.

Display Formats

All comments are scanned and sensitive characters are modified.

If column 6 contains an S, H, or D, then:

- Columns 7 through 14 are indicated.
- All other columns are scanned and sensitive characters are modified.
- Columns 7 through 14 of a continued D specification are modified.

Sort Members

All statements in the member are subject to modification except the ALTSEQ statement. MCSCU only indicates sensitive characters in the ALTSEQ specifications.

Work Station Utility

If column 7 contains an asterisk (*), indicating a comment, all columns will be scanned and sensitive characters will be modified.

If column 6 contains a J, T, or M, all columns are scanned and sensitive data is indicated.

If column 6 contains an S or D, the statement is the same as display formats.

If column 6 contains a C or E, the statement is processed like the RPG C- or E-specifications described earlier. In addition, IMSG, MSG, and compare to a table operation codes that have sensitive data within apostrophes (') will be modified.

Other Statement Type

If the source member is not uniquely identified by a previously described option, this option could be used. This option scans all statements in the member and modifies any MCS sensitive character that is detected when modify is selected.

Copying Back into the Library

When the conversion program is complete, the work file is copied back to the specified output library if modify was selected. It is your responsibility to ensure that the output library is large enough to hold all the converted source. The conversion utility uses the $MAINT utility program to copy the work file into the library.

Additional Library Member Considerations

MCSCU copies library members to a work file, modifies the members in the work file, and then copies them to the specified output library. Because some library member statements cannot be totally modified by MCSCU, some manual examination of the LIST option results should be performed. Any necessary manual changes should be made using SEU. Members that require recompilation or regeneration (such as RPG, COBOL, FORTRAN, assembler, DFU, message members, display formats, WSU, and menu members) must be recompiled. When you are finished running MCSCU against a work file, you should delete the work file using the DELETE procedure command.

If the output library becomes filled while the modified source is being copied back to the library from the work file, you can do the following steps:

- Use the ALOCLIBR procedure to increase the size of the output library.
- Enter the following OCL, which will copy the work file back to the output library:

```
// LOAD $MAINT
// FILE NAME-xxx
// RUN
// COPY FROM-DISK,TO-yyy,FILE-xxx
// END
```

where xxx is the work file name, and yyy is the output library name.

Data File Conversion

Your data files may contain characters that have taken on a different hexadecimal value under MCS. When a program displays or prints these data areas, the character associated with the MCS value instead of the character associated with the national language version (NLV) value is displayed. MCSCU helps in finding data within records that are MCS sensitive. You should be careful to avoid having MCSCU modify data fields containing binary or packed values that may appear to MCSCU to be sensitive.

If you select data file conversion, the prompt display shown in Figure D-5 appears. A description of each prompt follows the display.

```
                MCSDATA PROCEDURE                              Optional-*
                Convert data files

File name . . . . . . . . . . . . . . . . . . . . . . . . . . . _______
Creation date of the file . . . . . . . . . . . . . . . . . . . ______     *
Start scan value  . . . . . . . . . . . . . . . . . . 0001-4096  0001
End scan value  . . . . . . . . . . . . . . . . . . . 0001-4096  ____     *
National language . . . . . . . . . . . . . . . . . . . . . . . _______
Conversion option . . . . . . . . . . 1-NVL to MCS,2-MCS to NLV  1
List sensitive records? . . . . . . . . . . . . . . . . . Y,N  Y
Count number of sensitive records?  . . . . . . . . . . . . Y,N  Y
Modify sensitive records? . . . . . . . . . . . . . . . . . N,Y  N
Cmd3-Previous menu          Cmd7-End                  COPR IBM Corp. 1986
```

Figure D-5. Data File Conversion

File name specifies the data file to convert or examine.

Creation date of the file specifies the 6-digit creation date for this file, if you need to distinguish between files with the same name.

Start scan value: If you want the record scanned or modified within certain positions of the record, you can indicate starting position value. The value must be greater than zero and less than or equal to the record length. The default value is 1.

End scan value: You can indicate the end value for the scan within limits. The value must be greater than or equal to the start scan value and less than or equal to the record length. The default is the last byte of the record.

National language specifies the language group requested from the following list:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

MCSCU supplies the conversion tables for each of the national languages. If you wish to supply your own tables, you can do so by completing the message in the MCSCU message member (#MN#M2) for this purpose. When OTHER is selected, the messages in #MN#M2 are used. See "Conversion Tables" on page D-10.

Conversion option: MCSCU allows conversion from NLV to MCS by taking option 1, or MCS to NLV by taking option 2. The default is option 1.

List sensitive records? specifies whether MCSCU should list the records that contain sensitive characters. Option Y is list; option N is do not list. The default is Y. See "MCSCU Data File Listing" on page D-21 for more information.

Count number of sensitive records? specifies whether MCSCU should count the number of records in the file that contain sensitive characters. Option Y is count; option N is do not count. The default is Y. See "MCSCU Data File Listing" on page D-21 for more information.

Modify sensitive records? specifies whether MCSCU should modify the records as it scans them, or leave them as is. Option Y is modify; option N is do not modify. The default is N. See "Data File Modification" on page D-22 for a further explanation of what modify means for files.

MCSCU automatically determines the type of file (indexed, sequential, or direct) and does the modification accordingly.

As with library members, you can cause MCSCU to:

- List the MCS data sensitive records in the file
- Count the MCS sensitive records in the file
- Modify the MCS sensitive records in the file

You have the same options as you did for library member conversion in terms of:

- Determining language group to be converted
- Converting from NLV to MCS
- Converting from MCS to NLV
- Providing your own conversion table

Refer to "Library Member Modification" on page D-7 for an explanation.

The data conversion is an update-in-place operation. The file is not written to an intermediate file. It is recommended that you save your files before beginning the conversion. See the "SAVE Procedure" on page 4-416 for information about saving your files.

MCSCU automatically determines the type of file (sequential, indexed, or direct) that was selected and other attributes about the file such as record length, key length, and key location. The list, count, and modify options cannot all be N. An error message is issued if the list, count, and modify options are all N.

If scan limits are invalid, an error message is displayed and MCSCU ends.

The program that does the actual work file analysis and modification is an evoked program. This allows the actual conversion program to execute as a separate job, and allows control to immediately return to the MCSCONV procedure to prompt for additional conversion activity. Thus it is possible to have several different work files being analyzed and modified at the same time. Because the analysis and modification of the work file is still in progress, no additional members should be copied to the work file, or the work file should not be deleted, until the evoked job has completed.

MCSCU Data File Listing

A sample listing produced for a data file is as follows:

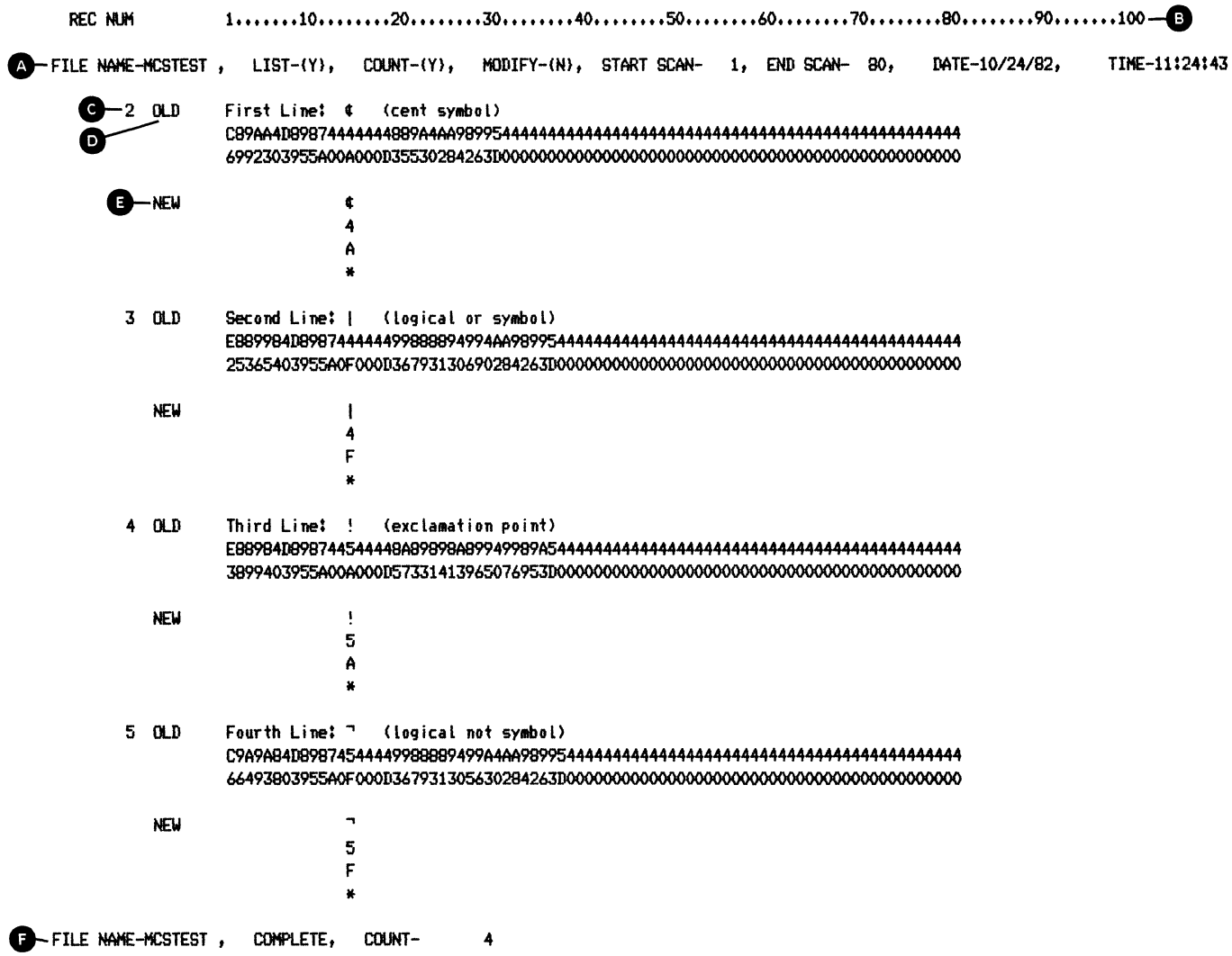


Figure D-6. Sample MCSCU Data File Listing

A An audit line containing the file name with the options in effect is listed with the sensitive records. LIST, COUNT, and MODIFY indicate either Y or N, based on the response to the prompt display.

B Column indicators help you determine the column where the sensitive character was found.

C The relative record number of the record within the file being processed.

D OLD indicates the original record. The record is printed in both character and hexadecimal format.

Ⓔ NEW indicates the record containing the sensitive characters. The sensitive characters have data printed below the original record. The characters printed on the NEW line indicate where the sensitive data was found.

The new sensitive character is also shown. If you selected N to the modify prompt, the original and new character will be equal. If you selected Y to the modify prompt:

- If it is a record position that MCSCU was allowed to modify, the new substituted character is printed.
- If it is a record position that MCSCU was not allowed to modify (in the index, outside limits), the original character is printed.

The character (if printing allows) and the hexadecimal value of the character are printed. An asterisk (*) is printed below each NEW line character that was sensitive but was not modified.

Ⓕ The number of records that were sensitive to MCS.

Data File Modification

The modify option uses the specified conversion table; wherever it finds an MCS sensitive character, it substitutes the alternate value. The only exception is that MCSCU does not modify the key field of an indexed file. It only indicates MCS sensitive data in the key field. You should use the scan limit values to avoid having MCSCU accidentally process binary and packed fields that could contain what appear to be sensitive characters. The same is true for fields in direct files that have a hashing technique used against them to locate a record. If MCSCU changes the value in a field that is hashed, the way the program runs is changed.

If you need to convert indexed files:

- Copy the indexed file to a sequential file under a new file name. See the "COPYDATA Procedure" on page 4-111 for information about copying data files.
- Run MCSCU against the sequential file.
- Delete the original indexed file. See the "DELETE Procedure" on page 4-144 for information about deleted data files.
- Copy the sequential file back to a new file with the original file name and attributes (indexed, key placement, key location, and so on). See the "COPYDATA Procedure" on page 4-111 for information about copying data files.

Document Folder Conversion

Your document folders may contain characters that have taken on a different hexadecimal value under MCS. When a program displays or prints these data areas, the character associated with the MCS value is displayed instead of the character associated with the national language version (NLV) value. MCSCU helps in finding data within records that are MCS sensitive.

If you select text folder conversion, the prompt display shown in Figure D-7 appears. A description of each prompt follows the display.

```
                          MCSFOLD PROCEDURE
                       Convert document folders

Document name, or to convert all documents, enter ALL . . . . .  _________
Folder name . . . . . . . . . . . . . . . . . . . . . . . . . .  _________
National language to convert to or from . . . . . . . . . . . .  _________
Convert folder description. . . . . . . . . . . . . . . . Y,N  N
Conversion option . . . . . . . . . 1-NLV to MCS,2-MCS to NLV  1
List sensitive records? . . . . . . . . . . . . . . . . . Y,N  Y
Count number of sensitive records?  . . . . . . . . . . . Y,N  Y
Modify sensitive records? . . . . . . . . . . . . . . . . N,Y  N


Cmd3-Previous menu            Cmd7-End                COPR IBM Corp. 1986
```

Figure D-7. Document Folder Conversion

Note: If DisplayWrite/36 (DW/36) is installed on your system, the first and second prompts are slightly different from those shown here.

Document name, or to convert all documents, enter ALL specifies the name of the document you want to convert or examine. If you enter ALL, all documents will be converted. If DW/36 is installed and you want to list all documents, leave this parameter blank.

Folder name specifies the name of the folder to be converted. If DW/36 is installed and you want to list all folders, leave this parameter blank.

National language to convert to or from specifies the language group requested from the following list:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

This entry is required. MCSCU supplies the conversion tables for each of the national languages. If you wish to supply your own tables, you can do so by completing the message in the MCSCU message member (#MN#M2) for this purpose. When OTHER is selected, the messages in #MN#M2 are used. See "Conversion Tables" on page D-10.

Convert folder description specifies whether MCSCU should convert the folder description. Option Y is convert folder description; option N is do not convert folder description. The default is N. See "MCSCU Document Folder Listing" on page D-25 for more information.

Conversion option: MCSCU allows conversion from NLV to MCS by option 1, or MCS to NLV by option 2. The default is option 1.

List sensitive records? specifies whether MCSCU should list the records that contain sensitive characters. Option Y is list; option N is do not list. The default is Y. See "MCSCU Document Folder Listing" on page D-25 for more information.

Count number of sensitive records? specifies whether MCSCU should count the number of records that contain sensitive characters. Option Y is count; option N is do not count. The default is Y. See "MCSCU Document Folder Listing" on page D-25 for more information.

Modify sensitive records? specifies whether MCSCU should modify the records as it scans them, or leave them as is. Option Y is modify; option N is do not modify. The default is N. DW/36 must be loaded on your system if you specify Y.

MCSCU Document Folder Listing

A sample listing produced for a document folder is as follows.

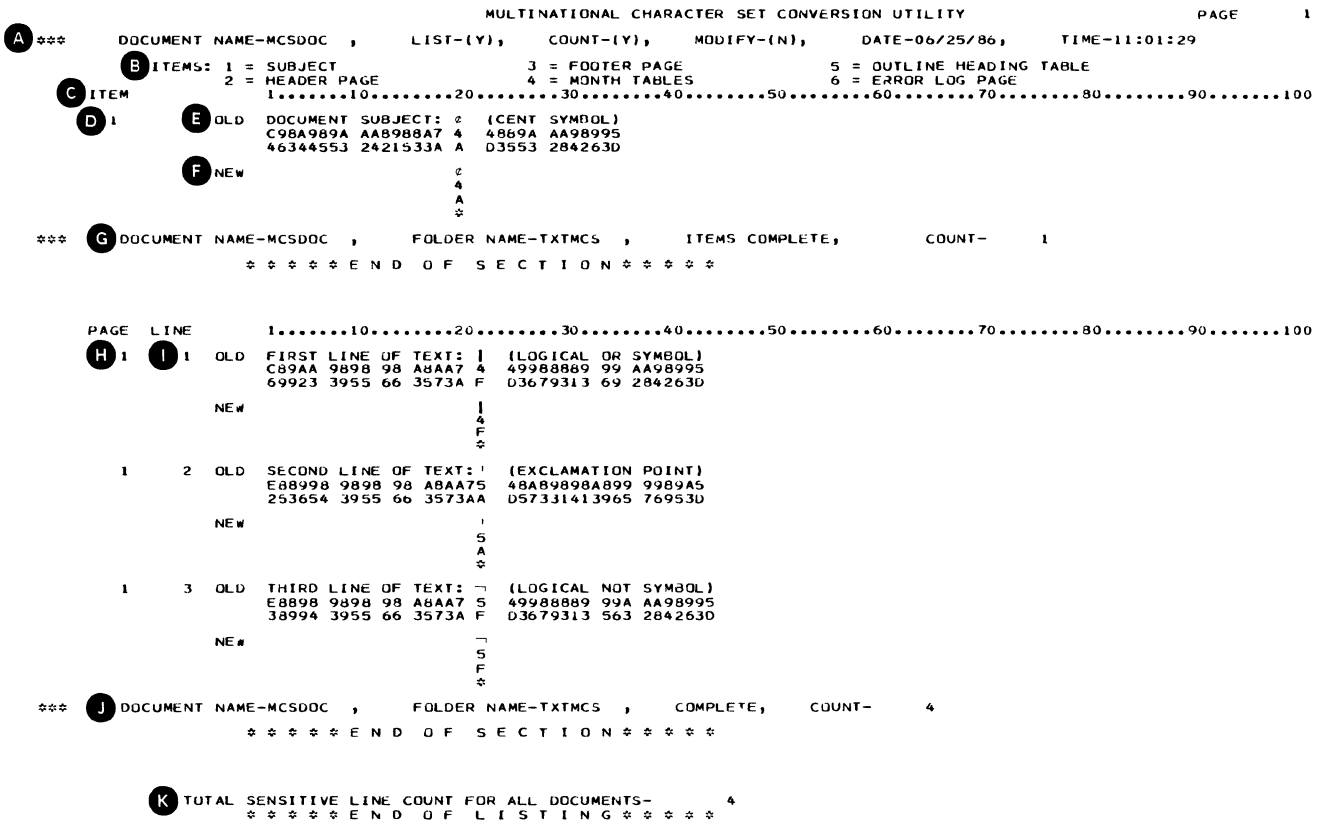


Figure D-8. Sample MCSCU Document Folder Listing

A An audit line containing the document name and the options in effect are listed along with the current date and time. LIST, COUNT, and MODIFY indicate Y or N, based on the response to the prompt display. The audit line is printed regardless of whether you selected the list option. The audit line is useful as a history of:

- Which documents have been processed in the folder
- Which processing options were requested

Ⓑ ITEMS indicates the different parts of a folder.

Ⓒ Column indicators help you determine the column in which the sensitive character was found.

Ⓓ This number corresponds to the items above.

Ⓔ OLD indicates the original statement. The statement is printed in both character and hexadecimal format.

Ⓕ NEW indicates that the statement contains sensitive characters. The sensitive characters have data printed below the original statement. The characters printed on the NEW line indicate where the sensitive data was found.

The new sensitive character is also shown. If you chose the option to not cause the sensitive statements to be modified, the old and new characters are equal. If you chose the option to cause the sensitive statements to be modified, the new substituted character is printed.

The character (if the printer allows) and the hexadecimal value of the character are printed. An asterisk (*) is printed below each NEW line character that was sensitive but was not modified.

Ⓖ A summary line containing document name, folder name, the message *Items complete* and the number of items that were sensitive to MCS. This line will be printed after all items have been processed and before the text part of the document is processed.

Ⓗ The number of the page within the document being processed.

Ⓘ The line number within the page of the document being processed.

Ⓙ This is the final summary line for the document. It shows the total count of sensitive items and sensitive lines.

Ⓚ After all documents have been processed, the total of all sensitives found in each document will be shown on this line.

Data Dictionary Conversion

Your dictionary members may contain characters that have taken on a new hexadecimal value under MCS. When a program displays or prints these data areas, the character associated with the MCS value instead of the character associated with the national language version (NLV) value is displayed. MCSCU helps to find data within records that are MCS sensitive.

If you select data dictionary conversion, the prompt display shown in Figure D-9 appears. A description of each prompt follows the display.

```
                          MCSDICT PROCEDURE

                      Convert data dictionaries

 Dictionary name . . . . . . . . . . . . . . . . . . . . . . . .  |||||||||

 National language to convert to or from . . . . . . . . . . . .  |||||||||

 Conversion option . . . . . . . . . . 1-NLV to MCS,2-MCS to NLV  1

 List sensitive records? . . . . . . . . . . . . . . . . . . Y,N  Y

 Count number of sensitive records?  . . . . . . . . . . . . Y,N  Y

 Modify sensitive records? . . . . . . . . . . . . . . . . . N,Y  N




Cmd3-Previous menu            Cmd7-End                   COPR IBM Corp. 1986
```

Figure D-9. Data Dictionary Conversion

Dictionary name specifies the name of the data dictionary you want to convert or examine. This value is required.

National language to convert to or from specifies the language group requested from the following list:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

This entry is required. MCSCU supplies the conversion tables for each of the national languages. If you wish to supply your own tables, you can do so by completing the message in the MCSCU message member (#MN#M2) for this purpose. When OTHER is selected, the messages in #MN#M2 are used. See "Conversion Tables" on page D-10.

Conversion option: MCSCU allows conversion from NLV to MCS by option 1, or MCS to NLV by option 2. The default is option 1.

List sensitive records? specifies whether MCSCU should list the records that contain sensitive characters. Option Y is list; option N is do not list. The default is Y. See "MCSCU Data Dictionary Listing" on page D-29 for more information.

Count number of sensitive records? specifies whether MCSCU should count the number of records in the file that contain sensitive characters. Option Y is count; option N is do not count. The default is Y. See "MCSCU Data Dictionary Listing" on page D-29 for more information.

Modify sensitive records? specifies whether MCSCU should modify the records as it scans them, or leaves them as is. Option Y is modify; option N is do not modify. The default is N.

MCSCU Data Dictionary Listing

A sample listing produced for a data dictionary is as follows.

```
    (A)      UTILITY: IBM SYSTEM/36 MULTINATIONAL CHARACTER SET CONVERSION UTILITY                          PAGE      1
***   DICTIONARY NAME-MCSDICT ,     LIST-(Y),     COUNT-(Y),     MODIFY-(N),     DATE-06/25/86,     TIME-11:13:12
   (B)ITEMS:    1 = SHORT DESCRIPTION          2 = LONG DESCRIPTION
 (C)ITEM          1.......10........20........30........40........50........60........70........80........90.......100
   (D)1       (E)OLD  SHORT COMMENT: ¢  (CENT SYMBOL)
                      E899A 899989A7 4  4889A AA98995
                      28693 3644553A A  D3553 284263D

              (F)NEW                  ¢
                                      4
                                      A
                                      ≠

      2           OLD  LONG COMMENT: ¦  (LOGICAL OR SYMBOL)
                       D998 899989A7 4  49988889 99 AA98995
                       3657 3644553A F  D3679313 69 284263D

                   NE*                ¦
                                      4
                                      F
                                      ≠

***(G)DICTIONARY NAME-MCSDICT ,       ITEMS COMPLETE,          COUNT-     2
                 * * * * * E N D  O F  S E C T I O N * * * * *


                                      MULTINATIONAL CHARACTER SET CONVERSION UTILITY                      PAGE      2
***  (H) DEFINITION NAME-MCSFLD  ,     DEFINITION TYPE-(FIELD    ), COUNT-(Y),     MODIFY-(N),       TIME-11:13:13
       ITEMS:    1 = SHORT DESCRIPTION          3 = COLUMN HEADINGS
                 2 = LONG DESCRIPTION
     ITEM          1.......10........20........30........40........50........60........70........80........90.......100
       1           OLD  FIELD DEFINITION: ¦  (EXCLAMATION POINT)
                        C4898 888898A8997 5  48A89898A899 9989A5
                        69534 4569593965A A  D57331413965 7b953D

                   NEW                    ¦
                                          5
                                          A
                                          ≠

       2           OLD  FIELD DEFINITION: ¬  (LOGICAL NOT SYMBOL)
                        C4898 888898A8997 5  49988889 99A AA98995
                        69534 4569593965A F  D3679313 563 284263D

                   NEW                    ¬
                                          5
                                          F
                                          ≠

***  (I) DEFINITION NAME-MCSFLD  ,    DICTIONARY NAME-MCSDICT ,     ITEMS COMPLETE,        COUNT-     2
                 * * * * * E N D  O F  S E C T I O N * * * * *

                      (J) TOTAL SENSITIVE ITEM COUNT-     4
                * * * * * E N D  O F  L I S T I N G * * * * *
```

Figure D-10. Sample MCSCU Data Dictionary Listing

Ⓐ An audit line containing the dictionary name and the options in effect are listed along with the current date and time. LIST, COUNT, and MODIFY indicate Y or N, based on the response to the prompt display. The audit line is printed regardless of whether you selected the list option.

Ⓑ ITEMS indicates either a short description or a long description.

Ⓒ Column indicators help you determine the column in which the sensitive character was found.

Ⓓ This number corresponds to the items listed above.

Ⓔ OLD indicates the original statement. The statement is printed in both character and hexadecimal format.

Ⓕ NEW indicates that the statement contained sensitive characters. The sensitive characters have data printed below the original statement. The characters printed on the NEW line indicate where the sensitive data was found.

The new sensitive character is also shown. If you chose the option to not cause the sensitive statements to be modified, the old and new characters will be equal. If you chose the option to cause the sensitive statements to be modified, the new substituted character is printed.

The character (if the printer allows) and the hexadecimal value of the character are printed. An asterisk (*) is printed below each NEW line character that was sensitive but was not modified.

Ⓖ A summary line containing dictionary name, the message *Items complete* and the count of sensitive items found in this section.

Ⓗ An audit line containing the definition name, definition type, and the count and modify options are listed.

Ⓘ A summary line containing definition name, dictionary name, the message *Items complete*, and the number of sensitive items found in the definition.

Ⓙ After all definitions have been processed, the total of all sensitives found will be shown on this line.

Batch Interface

If you want to set up a batch procedure to convert library members, files, text folders, or data dictionaries, you can do this. You might want to do this for a multiple location installation and be able to have a programmer in location X establish procedures for locations X, Y, and Z.

Note: When you run MCSCU using a batch interface, the programs that do the actual data file or work file analysis and modification are not evoked. There are also no parameter defaults from batch interface.

Library Members

The operation control language necessary to do member conversion is:

- Build OCL to run $MAINT and copy members to a disk file. The following OCL copies from library ABC, all source members beginning with PAY, to a file called PAYR with a record length of 120. Then it adds members that begin with AR to file PAYR.

  See the "Copy Members from a Library (FROMLIBR Procedure)" on page A-66 for more information about copying library members to disk files. Note the use of the RECL parameter on the COPY statements.

```
// LOAD $MAINT
// FILE NAME-PAYR,UNIT-F1,BLOCKS-100,RETAIN-T
// RUN
// COPY FROM-ABC,TO-DISK,FILE-PAYR,RECL-120,
// NAME-PAY.ALL,LIBRARY-S,SVATTR-YES
// COPY FROM-ABC,TO-DISK,FILE-PAYR,ADD-YES,
// NAME-AR.ALL,LIBRARY-S,SVATTR-YES
// END
```

- Set UPSI switch 1 on with the following OCL statement:

```
// SWITCH 10000000
```

  This causes all prompt displays to be suppressed.

- Start the MCSCU procedure MCSLIB:

```
MCSLIB   member type,output library name,language,from to,list,count,
         modify,file name
```

S9020473-0

member type is a value 1 through 8.

1 RPG source member.

2 Procedure member.

3 Message source member.

4 Menu source member created for the BLDMENU procedure.

5 Display format source member. Also, menus created by the screen design aid (SDA) utility.

6 Sort specifications source member.

7 WSU source members.

8 Some other source member (COBOL, FORTRAN, and so on) that is not supported by a specific member type option.

output library name is the name of the library that the file is to be copied to. This parameter is required if you specify Y to modify sensitive records.

language is one of the valid national languages described below:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

from to: 1 for NLV to MCS, 2 for MCS to NLV. If this parameter is not specified, 1 is assumed.

list: Y for list sensitive records, N for do not list. If this parameter is not specified, Y is assumed.

count: Y for count sensitive records, N for do not count. If this parameter is not specified, Y is assumed.

modify: Y for modify sensitive records, N for do not modify. If this parameter is not specified, N is assumed.

Note: The list, count, and modify options cannot all be N.

file name is the name of the work file containing the members to be processed.

Disk Files

The following steps are necessary to do file conversion:

- Set UPSI switch 1 on with the following OCL statement:

  ```
  // SWITCH 10000000
  ```

  This causes the prompt display to be suppressed.

- Start the MCSCU procedure MCSDATA:

```
MCSDATA  file name,[mmddyy|ddmmyy|yymmdd],start,[end],language,from to,list,count,modify,
```

S9020474-1

file name is the name of the file you want converted.

mmddyy, ddmmyy, or yymmdd is the creation date of the file you want converted. The date, if specified, must be in the session date format. Use the STATUS SESSION command to determine the session date format.

start is a 4-digit number, 0001 through 4096, that indicates the starting position of the record you want scanned or modified. The value must be greater than zero and less than or equal to the record length. If this parameter is not specified, 0001 is assumed.

end is a 4-digit number, 0001 through 4096, that indicates the ending position of the record you want scanned or modified. The value must be greater than or equal to the start scan value and less than or equal to the record length. The default is the last byte of the record.

language is one of the valid national languages described below:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

from to: 1 for NLV to MCS, 2 for MCS to NLV.

list: Y for list sensitive records, N for do not list.

count: Y for count sensitive records, N for do not count.

modify: Y for modify sensitive records, N for do not modify.

Note: The list, count, and modify options cannot all be N.

Document Folders

The following steps are necessary to do document folder conversion:

- Set UPSI switch 1 on with the following OCL statement:

  ```
  // SWITCH 10000000
  ```

  This causes the prompt display to be suppressed.

- Start the MCSCU procedure MCSFOLD:

```
MCSFOLD  {document name},folder name,language,convert folder description,
         {ALL          }

          fromto,list,count,modify
```

S9020561-1

document name is the name of the text document you want converted.

ALL specifies that all documents in the folder are to be converted.

folder name is the name of the document folder you want converted.

language specifies the language group requested from the following list:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

convert folder description: Y for convert folder description, N for do not convert. If this parameter is not specified, N is assumed.

from to: 1 for NLV to MCS, 2 for MCS to NLV. If this parameter is not specified, 1 is assumed.

list: Y for list sensitive records, N for do not list. If this parameter is not specified, Y is assumed.

count: Y for count sensitive records, N for do not count. If this parameter is not specified, Y is assumed.

modify: Y for modify sensitive records, N for do not modify. If this parameter is not specified, N is assumed. DW/36 must be loaded on your system if you specify Y.

Note: The list, count, and modify options cannot all be N.

Data Dictionaries

The following steps are necessary to do data dictionary conversion:

- Set UPSI switch 1 on with the following OCL statement:

  ```
  // SWITCH 10000000
  ```

  This causes the prompt display to be suppressed.

- Start the MCSCU procedure MCSDICT:

```
MCSDICT dictionary name,language,from to,list,count,modify
```

S9020560-0

dictionary name is the name of the data dictionary you want converted.

language specifies the language group requested from the following list:

USA	United States and Canada
BELGIUM	Belgium
BRAZIL	Brazil
CANADA	Canada (French)
DENMARK	Denmark and Norway
FINLAND	Finland and Sweden
FRANCE	France
GERMANY	Germany and Austria
ITALY	Italy
JAPAN	Japan (English)
PORTUGAL	Portugal
SPAIN	Spain
SPANISH	Spanish-speaking
UK	United Kingdom
OTHER	User-supplied

from to: 1 for NLV to MCS, 2 for MCS to NLV. If this parameter is not specified, 1 is assumed.

list: Y for list sensitive records, N for do not list. If this parameter is not specified, Y is assumed.

count: Y for count sensitive records, N for do not count. If this parameter is not specified, Y is assumed.

modify: Y for modify sensitive records, N for do not modify. If this parameter is not specified, N is assumed.

Note: The list, count, and modify options cannot all be N.

Changed Characters by Language Group

The following table shows, by language group, the System/36 characters that changed with the Multinational Character Set. The column labeled *Old Char* contains the original character. The column labeled *Old Hex* contains the original hexadecimal representation that has to be changed to the value shown in the column labeled *New Hex*. If this conversion is not done, the new character represented by the old hexadecimal value will be displayed and/or printed.

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
US & Canada (Code page 37)			
¢	4A	B0	[
|	4F	BB	!
!	5A	4F	]
¬	5F	BA	^
^	B0	5F	¢
[	BA	4A	¬
]	BB	5A	|
Austria & Germany (Code page 273)			
{	43	C0	ä
Ä	4A	63	[
~	59	A1	ß
Ü	5A	FC	]
[	63	4A	Ä
ö	6A	CC	¦
§	7C	B5	@
ß	A1	59	~
@	B5	7C	§
ä	C0	43	{
¦	CC	6A	ö
ü	D0	DC	}
}	DC	D0	ü
Ö	E0	EC	\
\	EC	E0	Ö
]	FC	5A	Ü

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
Belgium (Code page 274)			
@	44	7C	à
\	48	E0	ç
{	51	C0	é
}	54	D0	è
ù	6A	DD	¦
à	7C	44	@
¨	A1	BD	~
~	BD	A1	¨
é	C0	51	{
è	D0	54	}
¦	DD	6A	ù
ç	E0	48	\
Brazil (Code page 275)			
`	46	79	ã
¦	48	6A	ç
É	4A	71	[
}	51	D0	é
$	5A	5B	]
Ç	5B	68	$
@	66	7C	Ã
]	68	5A	Ç
ç	6A	48	¦
[	71	4A	É
ã	79	46	
Õ	7B	EF	#
Ã	7C	66	@
õ	C0	CF	{
{	CF	C0	õ
é	D0	51	}
#	EF	7B	Õ

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
Canada (French) (Code page 276)			
à	4A	44	[
´	5A	BE	]
ù	6A	DD	¦
¨	A1	BD	~
é	C0	51	{
è	D0	54	}
^	E0	9D	\
Denmark & Norway (Code page 277)			
}	47	D0	å
#	4A	7B	[
¤	5A	9F	]
Å	5B	67	$
$	67	5B	Å
ø	6A	70	¦
¦	70	6A	ø
Æ	7B	9E	#
Ø	7C	80	@
@	80	7C	Ø
{	9C	C0	æ
[	9E	4A	Æ
]	9F	5A	¤
ü	A1	DC	~
æ	C0	9C	{
å	D0	47	}
~	DC	A1	ü

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
Finland & Sweden (Code page 278)			
{	43	C0	ä
}	47	D0	å
§	4A	B5	[
`	51	79	é
¤	5A	9F	]
Å	5B	67	$
#	63	7B	Ä
$	67	5B	Å
ö	6A	CC	¦
\	71	E0	É
é	79	51	`
Ä	7B	63	#
Ö	7C	EC	@
]	9F	5A	¤
ü	A1	DC	~
[	B5	4A	§
ä	C0	43	{
¦	CC	6A	ö
å	D0	47	}
~	DC	A1	ü
É	E0	71	\
@	EC	7C	Ö

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
France (Code page 297)			
@	44	7C	à
\	48	E0	ç
°	4A	90	[
{	51	C0	é
}	54	D0	è
§	5A	B5	]
ù	6A	DD	¦
µ	79	A0	`
£	7B	B1	#
à	7C	44	@
[	90	4A	°
`	A0	79	µ
¨	A1	BD	~
#	B1	7B	£
]	B5	5A	§
~	BD	A1	¨
é	C0	51	{
è	D0	54	}
¦	DD	6A	ù
ç	E0	48	\

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
Italy (Code page 280)			
{	44	C0	à
\	48	E0	ç
°	4A	90	[
]	51	5A	é
}	54	D0	è
~	58	A1	ì
é	5A	51	]
ò	6A	CD	¦
ù	79	DD	`
£	7B	B1	#
§	7C	B5	@
[	90	4A	°
ì	A1	58	~
#	B1	7B	£
@	B5	7C	§
à	C0	44	{
¦	CD	6A	ò
è	D0	54	}
`	DD	79	ù
ç	E0	48	\
Japan (Code page 281)			
£	4A	B1	[
¦	4F	BB	!
!	5A	4F	]
¥	5B	B2	$
¬	5F	BA	^
-	A1	BC	~
[	B1	4A	£
\	B2	E0	¥
^	BA	5F	¬
]	BB	5A	¦
~	BC	A1	-
$	E0	5B	\

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
Portugal (Code page 282)			
{	46	C0	ã
~	48	A1	ç
#	66	7B	Ã
\	68	E0	Ç
õ	6A	CF	¦
Ã	7B	66	#
Õ	7C	EF	@
ç	A1	48	~
}	BE	D0	´
ã	C0	46	{
¦	CF	6A	õ
´	D0	BE	}
Ç	E0	68	\
@	EF	7C	Õ
Spain (Code page 284)			
¦	49	6A	ñ
|	4F	BB	!
¬	5F	BA	^
#	69	7B	Ñ
ñ	6A	49	¦
Ñ	7B	69	#
..	A1	BD	~
^	BA	5F	¬
!	BB	4F	|
~	BD	A1	..

Old Char	Old Hex (From)	New Hex (To)	New Character Represented by the Old Hex (If Not Changed)
Spanish-Speaking (Code page 284)			
¦	49	6A	ñ
|	4F	BB	!
¬	5F	BA	^
#	69	7B	Ñ
ñ	6A	49	¦
Ñ	7B	69	#
¨	A1	BD	~
^	BA	5F	¬
!	BB	4F	|
~	BD	A1	¨
United Kingdom (Code page 285)			
$	4A	5B	[
|	4F	BB	!
!	5A	4F	]
£	5B	B1	$
¬	5F	BA	^
—	A1	BC	~
[	B1	4A	£
^	BA	5F	¬
]	BB	5A	|
~	BC	A1	—

Appendix E. 3262 Printer Print Belts and Translation Tables

This appendix lists the print belt characters and the translation tables for the IBM 3262 Printer.

3262 Print Belts

This section indicates all the available characters and the print belt(s) the characters occur on. If the character is available, a dot (•) is shown under the heading for that belt. If a character is shown instead of the dot, that character corresponds to the hexadecimal value shown for that belt. If you want information on print belts other than the ones shown, contact your IBM sales representative.

The characters are shown in ascending sorting sequence; that is, the blank character is the lowest (it will be sorted before any other character) and the 9 character is the highest.

Character	Hexadecimal Equivalent	Print Belt 48 (BELT48)	48HN (BELT48HN)	64 (BELT64B BELT64C)	96 (BELT96)	188 (BELT188B)
Blank (not represented on the print belt)	40					•
( )	41					•
â	42					•
ä	43					•
à	44					•
á	45					•
ã	46					•
å	47					•
ç	48					•
ñ	49					•
¢	4A			•	•	[
. (period)	4B	•	•	•	•	•
<	4C			•	•	•
(	4D		•	•	•	•
+	4E	•	•	•	•	•
|	4F			•	•	!
&	50	•	•	•	•	•
é	51					•
ê	52					•
ë	53					•
è	54					•
í	55					•
î	56					•
ï	57					•
ì	58					•
ß	59					•
!	5A			•	•	]
$	5B	•	•	•	•	•
*	5C	•	•	•	•	•
)	5D		•	•	•	•
;	5E			•	•	•

Character	Hexadecimal Equivalent	Print Belt 48 (BELT48)	48HN (BELT48HN)	64 (BELT64B BELT64C)	96 (BELT96)	188 (BELT188B)
¬	5F			•	•	∧
- (minus)	60	•	•	•	•	•
/	61	•	•	•	•	•
Â	62					•
Ä	63					•
Ã	64					•
Á	65					•
À	66					•
Å	67					•
Ç	68					•
Ñ	69					•
¦	6A				•	•
, (comma)	6B	•	•	•	•	•
%	6C	•		•	•	•
_ (underscore)	6D			•	•	•
>	6E			•	•	•
?	6F			•	•	•
Ø	70					•
É	71					•
Ê	72					•
Ë	73					•
È	74					•
Í	75					•
Î	76					•
Ï	77					•
ì	78					•
` (grave accent)	79			•	•	•
:	7A			•	•	•
#	7B	•		•	•	•
@	7C	•		•	•	•
' (single quote)	7D	•	•	•	•	•
=	7E		•	•	•	•
"	7F			•	•	•
Ø	80					•
a	81				•	•
b	82				•	•
c	83				•	•
d	84				•	•
e	85				•	•
f	86				•	•
g	87				•	•
h	88				•	•
i	89				•	•
«	8A					•
»	8B					•
ð	8C					•
≤	8D					•
Þ	8E					•
±	8F					•
°	90					•
j	91				•	•

Character	Hexadecimal Equivalent	Print Belt 48 (BELT48)	48HN (BELT48HN)	64 (BELT64B BELT64C)	96 (BELT96)	188 (BELT188B)
k	92				•	•
l	93				•	•
m	94				•	•
n	95				•	•
o	96				•	•
p	97				•	•
q	98				•	•
r	99				•	•
a	9A					•
º	9B					•
¤	9C				•	&
≤	9D					•
Æ	9E					•
⊕	9F				•	¤
µ	A0					•
~	A1				•	•
s	A2				•	•
t	A3				•	•
u	A4				•	•
v	A5				•	•
w	A6				•	•
x	A7				•	•
y	A8				•	•
z	A9				•	•
i	AA					•
¿	AB					•
Ð	AC					•
↑	AD					•
§	AE					•
®	AF					•
¢	B0					•
£	B1					•
¥	B2					•
₨	B3					•
ƒ	B4					•
§	B5					•
¶	B6					•
¼	B7					•
½	B8					•
¾	B9					•
¬	BA					•
|	BB					•
≠	BC					•
..	BD					•
'	BE					•
=	BF					•
{	C0				•	•
A	C1	•	•	•	•	•
B	C2	•	•	•	•	•
C	C3	•	•	•	•	•
D	C4	•	•	•	•	•

Character	Hexadecimal Equivalent	Print Belt 48 (BELT48)	48HN (BELT48HN)	64 (BELT64B BELT64C)	96 (BELT96)	188 (BELT188B)
E	C5	•	•	•	•	•
F	C6	•	•	•	•	•
G	C7	•	•	•	•	•
H	C8	•	•	•	•	•
I	C9	•	•	•	•	•
–	CA					•
ô	CB					•
ö	CC					•
ò	CD					•
ó	CE					•
õ	CF					•
}	D0				•	•
J	D1	•	•	•	•	•
K	D2	•	•	•	•	•
L	D3	•	•	•	•	•
M	D4	•	•	•	•	•
N	D5	•	•	•	•	•
O	D6	•	•	•	•	•
P	D7	•	•	•	•	•
Q	D8	•	•	•	•	•
R	D9	•	•	•	•	•
≥	DA					•
û	DB					•
ü	DC					•
ù	DD					•
ú	DE					•
ÿ	DF					•
\	E0			•	•	•
S	E2	•	•	•	•	•
T	E3	•	•	•	•	•
U	E4	•	•	•	•	•
V	E5	•	•	•	•	•
W	E6	•	•	•	•	•
X	E7	•	•	•	•	•
Y	E8	•	•	•	•	•
Z	E9	•	•	•	•	•
²	EA					•
Ô	EB					•
Ö	EC					•
Ò	ED					•
Ó	EE					•
Õ	EF					•
0	F0	•	•	•	•	•
1	F1	•	•	•	•	•
2	F2	•	•	•	•	•
3	F3	•	•	•	•	•
4	F4	•	•	•	•	•
5	F5	•	•	•	•	•
6	F6	•	•	•	•	•
7	F7	•	•	•	•	•

Character	Hexadecimal Equivalent	Print Belt 48 (BELT48)	48HN (BELT48HN)	64 (BELT64B BELT64C)	96 (BELT96)	188 (BELT188B)
8	F8	•	•	•	•	•
9	F9	•	•	•	•	•
³	FA					•
Ó	FB					•
Ú	FC					•
Ý	FD					•
Ü	FE					•

3262 Translation Tables

This section indicates all the available characters and their translated characters. The characters are shown in ascending sorting sequence; that is, the blank character is the lowest (it will be sorted before any other character) and the 9 character is the highest.

Another translation table, named #188E188, is supplied. This table turns off translation. The open set of parentheses, specified by ( ), indicate that the corresponding hexadecimal code does not have a character.

Table #96E48

If you have the BELT48 character print belt installed, use this table to have lowercase characters printed as uppercase characters instead of blanks.

From Character	From Code	To Character	To Code
( )	40	( )	40
( )	41	( )	40
( )	42	( )	40
( )	43	( )	40
( )	44	( )	40
( )	45	( )	40
( )	46	( )	40
( )	47	( )	40
( )	48	( )	40
( )	49	( )	40
¢	4A	( )	40
.	4B	.	4B
⟨	4C	( )	40
(	4D	( )	40
+	4E	+	4E
|	4F	( )	40
&	50	&	50
( )	51	( )	40
( )	52	( )	40
( )	53	( )	40
( )	54	( )	40
( )	55	( )	40
( )	56	( )	40
( )	57	( )	40
( )	58	( )	40
( )	59	( )	40
!	5A	( )	40
$	5B	$	5B
*	5C	*	5C
)	5D	( )	40
;	5E	( )	40
¬	5F	( )	40
–	60	–	60
/	61	/	61
( )	62	( )	40
( )	63	( )	40
( )	64	( )	40
( )	65	( )	40
( )	66	( )	40
( )	67	( )	40
( )	68	( )	40
( )	69	( )	40
¦	6A	( )	40
,	6B	,	6B
%	6C	%	6C
_	6D	( )	40
⟩	6E	( )	40
?	6F	( )	40
( )	70	( )	40
( )	71	( )	40
( )	72	( )	40
( )	73	( )	40
( )	74	( )	40
( )	75	( )	40
( )	76	( )	40
( )	77	( )	40
( )	78	( )	40
`	79	( )	40
:	7A	( )	40
#	7B	#	7B
@	7C	@	7C
'	7D	'	7D
=	7E	( )	40
"	7F	( )	40
( )	80	( )	40
a	81	A	C1
b	82	B	C2
c	83	C	C3
d	84	D	C4
e	85	E	C5
f	86	F	C6
g	87	G	C7
h	88	H	C8
i	89	I	C9
( )	8A	( )	40
( )	8B	( )	40
( )	8C	( )	40
( )	8D	( )	40
( )	8E	( )	40
( )	8F	( )	40
( )	90	( )	40
j	91	J	D1
k	92	K	D2
l	93	L	D3
m	94	M	D4
n	95	N	D5
o	96	O	D6
p	97	P	D7
q	98	Q	D8
r	99	R	D9
( )	9A	( )	40
( )	9B	( )	40
( )	9C	( )	40
( )	9D	( )	40
( )	9E	( )	40
( )	9F	( )	40

Table #96E48 (continued)

From Character	From Code	To Character	To Code
( )	A0	( )	40
( )	A1	( )	40
s	A2	S	E2
t	A3	T	E3
u	A4	U	E4
v	A5	V	E5
w	A6	W	E6
x	A7	X	E7
y	A8	Y	E8
z	A9	Z	E9
( )	AA	( )	40
( )	AB	( )	40
( )	AC	( )	40
( )	AD	( )	40
( )	AE	( )	40
( )	AF	( )	40
( )	B0	( )	40
( )	B1	( )	40
( )	B2	( )	40
( )	B3	( )	40
( )	B4	( )	40
( )	B5	( )	40
( )	B6	( )	40
( )	B7	( )	40
( )	B8	( )	40
( )	B9	( )	40
( )	BA	( )	40
( )	BB	( )	40
( )	BC	( )	40
( )	BD	( )	40
( )	BE	( )	40
( )	BF	( )	40
(	C0	( )	40
A	C1	A	C1
B	C2	B	C2
C	C3	C	C3
D	C4	D	C4
E	C5	E	C5
F	C6	F	C6
G	C7	G	C7
H	C8	H	C8
I	C9	I	C9
( )	CA	( )	40
( )	CB	( )	40
( )	CC	( )	40
( )	CD	( )	40
( )	CE	( )	40
( )	CF	( )	40

From Character	From Code	To Character	To Code
( )	D0	( )	40
J	D1	J	D1
K	D2	K	D2
L	D3	L	D3
M	D4	M	D4
N	D5	N	D5
O	D6	O	D6
P	D7	P	D7
Q	D8	Q	D8
R	D9	R	D9
( )	DA	( )	40
( )	DB	( )	40
( )	DC	( )	40
( )	DD	( )	40
( )	DE	( )	40
( )	DF	( )	40
\	E0	( )	40
( )	E1	( )	40
S	E2	S	E2
T	E3	T	E3
U	E4	U	E4
V	E5	V	E5
W	E6	W	E6
X	E7	X	E7
Y	E8	Y	E8
Z	E9	Z	E9
( )	EA	( )	40
( )	EB	( )	40
( )	EC	( )	40
( )	ED	( )	40
( )	EE	( )	40
( )	EF	( )	40
0	F0	0	F0
1	F1	1	F1
2	F2	2	F2
3	F3	3	F3
4	F4	4	F4
5	F5	5	F5
6	F6	6	F6
7	F7	7	F7
8	F8	8	F8
9	F9	9	F9
( )	FA	( )	40
( )	FB	( )	40
( )	FC	( )	40
( )	FD	( )	40
( )	FE	( )	40

Table #96E64

If you have the BELT64B or BELT64C character print belt installed, use this table to have lowercase characters printed as uppercase characters instead of blanks.

From		To	
Character	Code	Character	Code
( )	40	( )	40
( )	41	( )	40
( )	42	( )	40
( )	43	( )	40
( )	44	( )	40
( )	45	( )	40
( )	46	( )	40
( )	47	( )	40
( )	48	( )	40
( )	49	( )	40
¢	4A	¢	4A
.	4B	.	4B
<	4C	<	4C
(	4D	(	4D
+	4E	+	4E
|	4F	|	4F
&	50	&	50
( )	51	( )	40
( )	52	( )	40
( )	53	( )	40
( )	54	( )	40
( )	55	( )	40
( )	56	( )	40
( )	57	( )	40
( )	58	( )	40
( )	59	( )	40
!	5A	!	5A
$	5B	$	5B
*	5C	*	5C
)	5D	)	5D
;	5E	;	5E
¬	5F	¬	5F
–	60	–	60
/	61	/	61
( )	62	( )	40
( )	63	( )	40
( )	64	( )	40
( )	65	( )	40
( )	66	( )	40
( )	67	( )	40
( )	68	( )	40
( )	69	( )	40
¦	6A	( )	40
,	6B	,	6B
%	6C	%	6C
_	6D	_	6D
>	6E	>	6E
?	6F	?	6F
( )	70	( )	40
( )	71	( )	40
( )	72	( )	40
( )	73	( )	40
( )	74	( )	40
( )	75	( )	40
( )	76	( )	40
( )	77	( )	40
( )	78	( )	40
`	79	`	79
:	7A	:	7A
#	7B	#	7B
@	7C	@	7C
'	7D	'	7D
=	7E	=	7E
"	7F	"	7F
( )	80	( )	40
a	81	A	C1
b	82	B	C2
c	83	C	C3
d	84	D	C4
e	85	E	C5
f	86	F	C6
g	87	G	C7
h	88	H	C8
i	89	I	C9
( )	8A	( )	40
( )	8B	( )	40
( )	8C	( )	40
( )	8D	( )	40
( )	8E	( )	40
( )	8F	( )	40
( )	90	( )	40
j	91	J	D1
k	92	K	D2
l	93	L	D3
m	94	M	D4
n	95	N	D5
o	96	O	D6
p	97	P	D7
q	98	Q	D8
r	99	R	D9
( )	9A	( )	40
( )	9B	( )	40
( )	9C	( )	40
( )	9D	( )	40
( )	9E	( )	40
( )	9F	( )	40

Table #96E64 (continued)

From		To	
Character	Code	Character	Code
( )	A0	( )	40
( )	A1	( )	40
s	A2	S	E2
t	A3	T	E3
u	A4	U	E4
v	A5	V	E5
w	A6	W	E6
x	A7	X	E7
y	A8	Y	E8
z	A9	Z	E9
( )	AA	( )	40
( )	AB	( )	40
( )	AC	( )	40
( )	AD	( )	40
( )	AE	( )	40
( )	AF	( )	40
( )	B0	( )	40
( )	B1	( )	40
( )	B2	( )	40
( )	B3	( )	40
( )	B4	( )	40
( )	B5	( )	40
( )	B6	( )	40
( )	B7	( )	40
( )	B8	( )	40
( )	B9	( )	40
( )	BA	( )	40
( )	BB	( )	40
( )	BC	( )	40
( )	BD	( )	40
( )	BE	( )	40
( )	BF	( )	40
(	C0	( )	40
A	C1	A	C1
B	C2	B	C2
C	C3	C	C3
D	C4	D	C4
E	C5	E	C5
F	C6	F	C6
G	C7	G	C7
H	C8	H	C8
I	C9	I	C9
( )	CA	( )	40
( )	CB	( )	40
( )	CC	( )	40
( )	CD	( )	40
( )	CE	( )	40
( )	CF	( )	40

From		To	
Character	Code	Character	Code
( )	D0	( )	40
J	D1	J	D1
K	D2	K	D2
L	D3	L	D3
M	D4	M	D4
N	D5	N	D5
O	D6	O	D6
P	D7	P	D7
Q	D8	Q	D8
R	D9	R	D9
( )	DA	( )	40
( )	DB	( )	40
( )	DC	( )	40
( )	DD	( )	40
( )	DE	( )	40
( )	DF	( )	40
\	E0	\	E0
( )	E1	( )	40
S	E2	S	E2
T	E3	T	E3
U	E4	U	E4
V	E5	V	E5
W	E6	W	E6
X	E7	X	E7
Y	E8	Y	E8
Z	E9	Z	E9
( )	EA	( )	40
( )	EB	( )	40
( )	EC	( )	40
( )	ED	( )	40
( )	EE	( )	40
( )	EF	( )	40
0	F0	0	F0
1	F1	1	F1
2	F2	2	F2
3	F3	3	F3
4	F4	4	F4
5	F5	5	F5
6	F6	6	F6
7	F7	7	F7
8	F8	8	F8
9	F9	9	F9
( )	FA	( )	40
( )	FB	( )	40
( )	FC	( )	40
( )	FD	( )	40
( )	FE	( )	40

Table #188E48

If you have the BELT48 character print belt installed, use this table to have lowercase characters printed as uppercase characters instead of blanks. Also, international characters are printed as alphabetic characters instead of blanks.

From		To	
Character	Code	Character	Code
( )	40	( )	40
( )	41	( )	40
â	42	A	C1
ä	43	A	C1
à	44	A	C1
á	45	A	C1
ã	46	A	C1
å	47	A	C1
ç	48	C	C3
ñ	49	N	D5
[	4A	( )	40
.	4B	.	4B
<	4C	( )	40
(	4D	( )	40
+	4E	+	4E
!	4F	( )	40
&	50	&	50
é	51	E	C5
ê	52	E	C5
ë	53	E	C5
è	54	E	C5
í	55	I	C9
î	56	I	C9
ï	57	I	C9
ì	58	I	C9
ß	59	S	E2
]	5A	( )	40
$	5B	$	5B
*	5C	*	5C
)	5D	( )	40
;	5E	( )	40
^	5F	( )	40
–	60	–	60
/	61	/	61
Â	62	A	C1
Ä	63	A	C1
Ã	64	A	C1
Á	65	A	C1
À	66	A	C1
Å	67	A	C1
Ç	68	C	C3
Ñ	69	N	D5
¦	6A	( )	40
,	6B	,	6B
%	6C	%	6C
_	6D	( )	40
>	6E	( )	40
?	6F	( )	40

From		To	
Character	Code	Character	Code
ø	70	O	D6
É	71	E	C5
Ê	72	E	C5
Ë	73	E	C5
È	74	E	C5
Í	75	I	C9
Î	76	I	C9
Ï	77	I	C9
Ì	78	I	C9
`	79	( )	40
:	7A	( )	40
#	7B	#	7B
@	7C	@	7C
'	7D	'	7D
=	7E	( )	40
"	7F	( )	40
Ø	80	O	D6
a	81	A	C1
b	82	B	C2
c	83	C	C3
d	84	D	C4
e	85	E	C5
f	86	F	C6
g	87	G	C7
h	88	H	C8
i	89	I	C9
«	8A	( )	40
»	8B	( )	40
ð	8C	D	C4
≤	8D	( )	40
þ	8E	( )	40
±	8F	( )	40
°	90	( )	40
j	91	J	D1
k	92	K	D2
l	93	L	D3
m	94	M	D4
n	95	N	D5
o	96	O	D6
p	97	P	D7
q	98	Q	D8
r	99	R	D9
ª	9A	A	C1
º	9B	O	D6
æ	9C	( )	40
¸	9D	( )	40
Æ	9E	( )	40
¤	9F	( )	40

Table #188E48 (continued)

From Character	From Code	To Character	To Code
μ	A0	( )	40
~	A1	( )	40
s	A2	S	E2
t	A3	T	E3
u	A4	U	E4
v	A5	V	E5
w	A6	W	E6
x	A7	X	E7
y	A8	Y	E8
z	A9	Z	E9
¡	AA	( )	40
¿	AB	( )	40
Ð	AC	D	C4
†	AD	( )	40
Þ	AE	( )	40
®	AF	( )	40
¢	B0	( )	40
£	B1	( )	40
¥	B2	( )	40
Pts	B3	( )	40
ƒ	B4	( )	40
§	B5	( )	40
¶	B6	( )	40
¼	B7	( )	40
½	B8	( )	40
¾	B9	( )	40
¬	BA	( )	40
|	BB	( )	40
≠	BC	( )	40
..	BD	( )	40
,	BE	( )	40
=	BF	( )	40
{	C0	( )	40
A	C1	A	C1
B	C2	B	C2
C	C3	C	C3
D	C4	D	C4
E	C5	E	C5
F	C6	F	C6
G	C7	G	C7
H	C8	H	C8
I	C9	I	C9
—	CA	—	CA
ô	CB	O	D6
ö	CC	O	D6
ò	CD	O	D6
ó	CE	O	D6
õ	CF	O	D6
}	D0	( )	40
J	D1	J	D1
K	D2	K	D2
L	D3	L	D3
M	D4	M	D4
N	D5	N	D5
O	D6	O	D6
P	D7	P	D7
Q	D8	Q	D8
R	D9	R	D9
≥	DA	( )	40
û	DB	U	E4
ü	DC	U	E4
ù	DD	U	E4
ú	DE	U	E4
ÿ	DF	Y	E8
\	E0	( )	40
( )	E1	( )	40
S	E2	S	E2
T	E3	T	E3
U	E4	U	E4
V	E5	V	E5
W	E6	W	E6
X	E7	X	E7
Y	E8	Y	E8
Z	E9	Z	E9
²	EA	( )	40
ô	EB	O	D6
ö	EC	O	D6
ò	ED	O	D6
ó	EE	O	D6
õ	EF	O	D6
0	F0	0	F0
1	F1	1	F1
2	F2	2	F2
3	F3	3	F3
4	F4	4	F4
5	F5	5	F5
6	F6	6	F6
7	F7	7	F7
8	F8	8	F8
9	F9	9	F9
³	FA	( )	40
û	FB	U	E4
ü	FC	U	E4
ù	FD	U	E4
ú	FE	U	E4

Table #188E64

If you have the BELT64B or BELT64C character print belt installed, use this table to have lowercase characters printed as uppercase characters instead of blanks. Also, international characters are printed as alphabetic characters instead of blanks.

From		To	
Character	Code	Character	Code
( )	40	( )	40
( )	41	( )	40
â	42	A	C1
ä	43	A	C1
à	44	A	C1
á	45	A	C1
ã	46	A	C1
å	47	A	C1
ç	48	C	C3
ñ	49	N	D5
[	4A	( )	40
.	4B	.	4B
<	4C	<	4C
(	4D	(	4D
+	4E	+	4E
!	4F	!	4F
&	50	&	50
ê	51	E	C5
ê	52	E	C5
ë	53	E	C5
è	54	E	C5
í	55	I	C9
î	56	I	C9
ï	57	I	C9
ì	58	I	C9
β	59	S	E2
]	5A	( )	40
$	5B	$	5B
*	5C	*	5C
)	5D	)	5D
;	5E	;	5E
^	5F	( )	40
–	60	–	60
/	61	/	61
Â	62	A	C1
Ä	63	A	C1
À	64	A	C1
Á	65	A	C1
Å	66	A	C1
Å	67	A	C1
Ç	68	C	C3
Ñ	69	N	D5
¦	6A	( )	40
,	6B	,	6B
%	6C	%	6C
_	6D	–	6D
>	6E	>	6E
?	6F	?	6F
ø	70	O	D6
É	71	E	C5
Ê	72	E	C5
Ë	73	E	C5
È	74	E	C5
Í	75	I	C9
Î	76	I	C9
Ï	77	I	C9
Ì	78	I	C9
`	79	`	79
:	7A	:	7A
#	7B	#	7B
@	7C	@	7C
'	7D	'	7D
=	7E	=	7E
"	7F	"	7F
Ø	80	O	D6
a	81	A	C1
b	82	B	C2
c	83	C	C3
d	84	D	C4
e	85	E	C5
f	86	F	C6
g	87	G	C7
h	88	H	C8
i	89	I	C9
«	8A	( )	40
»	8B	( )	40
ð	8C	D	C4
≤	8D	( )	40
þ	8E	( )	40
±	8F	( )	40
°	90	( )	40
j	91	J	D1
k	92	K	D2
l	93	L	D3
m	94	M	D4
n	95	N	D5
o	96	O	D6
p	97	P	D7
q	98	Q	D8
r	99	R	D9
ª	9A	A	C1
º	9B	O	D6
æ	9C	( )	40
¸	9D	( )	40
Æ	9E	( )	40
¤	9F	( )	40

Table #188E64 (continued)

From Character	From Code	To Character	To Code
μ	A0	( )	40
~	A1	( )	40
s	A2	S	E2
t	A3	T	E3
u	A4	U	E4
v	A5	V	E5
w	A6	W	E6
x	A7	X	E7
y	A8	Y	E8
z	A9	Z	E9
¡	AA	( )	40
¿	AB	( )	40
Ð	AC	D	C4
†	AD	( )	40
Þ	AE	( )	40
®	AF	( )	40
¢	B0	( )	40
£	B1	( )	40
¥	B2	( )	40
Pts	B3	( )	40
ƒ	B4	( )	40
§	B5	( )	40
¶	B6	( )	40
¼	B7	( )	40
½	B8	( )	40
¾	B9	( )	40
¬	BA	( )	40
|	BB	( )	40
≠	BC	( )	40
..	BD	( )	40
,	BE	( )	40
=	BF	( )	40
{	C0	( )	40
A	C1	A	C1
B	C2	B	C2
C	C3	C	C3
D	C4	D	C4
E	C5	E	C5
F	C6	F	C6
G	C7	G	C7
H	C8	H	C8
I	C9	I	C9
–	CA	( )	40
ô	CB	O	D6
ö	CC	O	D6
ò	CD	O	D6
ó	CE	O	D6
õ	CF	O	D6
}	D0	( )	40
J	D1	J	D1
K	D2	K	D2
L	D3	L	D3
M	D4	M	D4
N	D5	N	D5
O	D6	O	D6
P	D7	P	D7
Q	D8	Q	D8
R	D9	R	D9
≥	DA	( )	40
û	DB	U	E4
ü	DC	U	E4
ù	DD	U	E4
ú	DE	U	E4
ÿ	DF	Y	E8
\	E0	\	E0
( )	E1	( )	E1
S	E2	S	E2
T	E3	T	E3
U	E4	U	E4
V	E5	V	E5
W	E6	W	E6
X	E7	X	E7
Y	E8	Y	E8
Z	E9	Z	E9
²	EA	( )	40
ô	EB	O	D6
ö	EC	O	D6
ò	ED	O	D6
ó	EE	O	D6
õ	EF	O	D6
0	F0	0	F0
1	F1	1	F1
2	F2	2	F2
3	F3	3	F3
4	F4	4	F4
5	F5	5	F5
6	F6	6	F6
7	F7	7	F7
8	F8	8	F8
9	F9	9	F9
³	FA	( )	40
Û	FB	U	E4
Ü	FC	U	E4
Ù	FD	U	E4
Ú	FE	U	E4

Table #188E96

If you have the BELT96 character print belt installed, use this table to have international characters printed as alphabetic characters instead of blanks.

From		To	
Character	Code	Character	Code
( )	40	( )	40
( )	41	( )	40
â	42	a	81
ä	43	a	81
à	44	a	81
á	45	a	81
ã	46	a	81
å	47	a	81
ç	48	c	83
ñ	49	n	95
[	4A	( )	40
.	4B	.	4B
⟨	4C	⟨	4C
(	4D	(	4D
+	4E	+	4E
!	4F	!	4F
&	50	&	50
é	51	e	85
ê	52	e	85
ë	53	e	85
è	54	e	85
í	55	i	89
î	56	i	89
ï	57	i	89
ì	58	i	89
ß	59	s	A2
]	5A	( )	40
$	5B	$	5B
*	5C	*	5C
)	5D	)	5D
;	5E	;	5E
∧	5F	( )	40
–	60	–	60
/	61	/	61
Â	62	A	C1
Ä	63	A	C1
Ã	64	A	C1
Á	65	A	C1
À	66	A	C1
Å	67	A	C1
Ç	68	C	C3
Ñ	69	N	D5
¦	6A	¦	6A
,	6B	,	6B
%	6C	%	6C
_	6D	_	6D
⟩	6E	⟩	6E
?	6F	?	6F

From		To	
Character	Code	Character	Code
ø	70	o	96
É	71	E	C5
Ê	72	E	C5
Ë	73	E	C5
È	74	E	C5
Í	75	I	C9
Î	76	I	C9
Ï	77	I	C9
Ì	78	I	C9
`	79	`	79
:	7A	:	7A
#	7B	#	7B
@	7C	@	7C
'	7D	'	7D
=	7E	=	7E
"	7F	"	7F
Ø	80	0	D6
a	81	a	81
b	82	b	82
c	83	c	83
d	84	d	84
e	85	e	85
f	86	f	86
g	87	g	87
h	88	h	88
i	89	i	89
«	8A	( )	40
»	8B	( )	40
ð	8C	d	84
≤	8D	( )	40
þ	8E	( )	40
±	8F	( )	40
.	90	( )	40
j	91	j	91
k	92	k	92
l	93	l	93
m	94	m	94
n	95	n	95
o	96	o	96
p	97	p	97
q	98	q	98
r	99	r	99
a	9A	a	81
º	9B	o	96
æ	9C	( )	40
¸	9D	( )	40
Æ	9E	( )	40
¤	9F	( )	40

Table #188E96 (continued)

From		To	
Character	Code	Character	Code
μ	A0	( )	40
~	A1	~	A1
s	A2	s	A2
t	A3	t	A3
u	A4	u	A4
v	A5	v	A5
w	A6	w	A6
x	A7	x	A7
y	A8	y	A8
z	A9	z	A9
¡	AA	( )	40
¿	AB	( )	40
Ð	AC	D	C4
†	AD	( )	40
Þ	AE	( )	40
®	AF	( )	40
¢	B0	( )	40
£	B1	( )	40
¥	B2	( )	40
Pts	B3	( )	40
ƒ	B4	( )	40
§	B5	( )	40
¶	B6	( )	40
¼	B7	( )	40
½	B8	( )	40
¾	B9	( )	40
¬	BA	( )	40
|	BB	( )	40
≠	BC	( )	40
··	BD	( )	40
'	BE	( )	40
=	BF	( )	40
(	C0	(	C0
A	C1	A	C1
B	C2	B	C2
C	C3	C	C3
D	C4	D	C4
E	C5	E	C5
F	C6	F	C6
G	C7	G	C7
H	C8	H	C8
I	C9	I	C9
-	CA	( )	40
ô	CB	o	96
ö	CC	o	96
õ	CD	o	96
ó	CE	o	96
ò	CF	o	96

From		To	
Character	Code	Character	Code
}	D0	}	D0
J	D1	J	D1
K	D2	K	D2
L	D3	L	D3
M	D4	M	D4
N	D5	N	D5
O	D6	O	D6
P	D7	P	D7
Q	D8	Q	D8
R	D9	R	D9
≥	DA	( )	40
û	DB	u	A4
ü	DC	u	A4
ù	DD	u	A4
ú	DE	u	A4
ÿ	DF	y	A8
\	E0	\	E0
( )	E1	( )	E1
S	E2	S	E2
T	E3	T	E3
U	E4	U	E4
V	E5	V	E5
W	E6	W	E6
X	E7	X	E7
Y	E8	Y	E8
Z	E9	Z	E9
²	EA	( )	40
ô	EB	O	D6
ö	EC	O	D6
õ	ED	O	D6
ó	EE	O	D6
ò	EF	O	D6
0	F0	0	F0
1	F1	1	F1
2	F2	2	F2
3	F3	3	F3
4	F4	4	F4
5	F5	5	F5
6	F6	6	F6
7	F7	7	F7
8	F8	8	F8
9	F9	9	F9
³	FA	( )	40
û	FB	U	E4
ü	FC	U	E4
ù	FD	U	E4
ú	FE	U	E4

Appendix F. EBCDIC and ASCII Code Tables

The character sets for EBCDIC (extended binary coded decimal interchange code) and ASCII (American National Standard Code for Information Interchange) are shown in the following tables. Use the set that your system supports.

EBCDIC

For example, hex C1 (binary 1100 0001) is the letter A.

		Main Storage Bit Positions 0, 1, 2, 3															
Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4, 5, 6, 7	Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	DS		SP	&	–						{	}	\	0
0001	1	SOH	DC1	SOS		RSP		/		a	j	~		A	J	NSP	1
0010	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	3	ETX	DC3	WUS	IR					c	l	t		C	L	T	3
0100	4	SEL	ENP RES	INP BYP	PP					d	m	u		D	M	U	4
0101	5	HT	NL	LF	TRN					e	n	v		E	N	V	5
0110	6		BS	ETB	NBS					f	o	w		F	O	W	6
0111	7	DEL	POC	ESC	EOT					g	p	x		G	P	X	7
1000	8	GE	CAN		SBS					h	q	y		H	Q	Y	8
1001	•9	SPS	EM		IT				`	i	r	z		I	R	Z	9
1010	A	RPT	UBS	SM/SW	RFF	¢	!	¦	:					SHY			I
1011	B	VT	CU1	FMT	CU3	.	$	,	#								
1100	C	FF	IFS		DC4	<	*	%	@								
1101	D	CR	IGS	ENQ	NAK	(	)	_	'								
1110	E	SO	IRS	ACK		+	;	>	=								
1111	F	SI	ITB/IUS	BEL	SUB	|	¬	?	"								E0

Duplicate Assignment

ASCII

For example, hex 41 (binary 0100 0001) is the letter A.

		Main Storage Bit Positions 0, 1, 2, 3															
Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4, 5, 6, 7	Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	SP	0	@	P	`	p								
0001	1	SOH	DC1	!	1	A	Q	a	q								
0010	2	STX	DC2	"	2	B	R	b	r								
0011	3	ETX	DC3	#	3	C	S	c	s								
0100	4	EOT	DC4	$	4	D	T	d	t								
0101	5	ENQ	NAK	%	5	E	U	e	u								
0110	6	ACK	SYN	&	6	F	V	f	v								
0111	7	BEL	ETB	'	7	G	W	g	w								
1000	8	BS	CAN	(	8	H	X	h	x								
1001	9	HT	EM	)	9	I	Y	i	y								
1010	A	LF	SUB	*	:	J	Z	j	z								
1011	B	VT	ESC	+	;	K	[	k	{								
1100	C	FF	FS	,	<	L	\	l	|							
1101	D	CR	GS	-	=	M	]	m	}								
1110	E	SO	RS	.	>	N	^	n	~								
1111	F	SI	US	/	?	O	_	o	DEL								

Appendix G. Using the IDDUXLAT Procedure

Introduction

The IDDUXLAT procedure translates existing RPG program source members or Text Management System (TMS) data definitions into new IDDU field, format, and file definitions. This appendix contains the following information about the use of IDDUXLAT:

- Considerations for running IDDUXLAT, and rules about the way IDDUXLAT interprets and translates the source specifications
- An example of the use of IDDUXLAT. This example includes a listing of the source specifications used as input to IDDUXLAT, as well as the printed output received from IDDUXLAT.

For the syntax format and parameter descriptions of the IDDUXLAT procedure, see Chapter 4 of this manual. For more information about IDDU, read *Getting Started With Interactive Data Definition Utility*, SC21-8003, or the online information for IDDU (select option 11 from the IDDU system help menu).

Considerations and Rules

What You Should Consider Before You Run IDDUXLAT

The following items should be considered before you run the IDDUXLAT procedure:

- Use the IDDUXLAT procedure to translate compilable, error-free source members.
- The IDDUXLAT procedure does not change the source specifications in any way.
- The IDDUXLAT procedure can create definitions only for input or update disk files (except record address files and table files); all other file types, including work station and communications files, are ignored and cannot be translated.

- In some instances, a file may be described by specifications in several different source members. If the specifications describing that file differ in any way, IDDUXLAT will create new and separate definitions for each variation. Thus, many definitions (each with a unique name) might be created for a single file when a single definition would suffice.

  Therefore, you should avoid the possibility of creating excessive and duplicate, though uniquely named, file definitions. Carefully examine the contents of your source specifications, and translate each variation of the particular file description specifications only once.

- The IDDUXLAT procedure can create a maximum of 373 unique definitions for each source member that is processed. If a source member you want to translate contains more than 373 file, record format, and field descriptions, you should consider splitting up the source specifications.

  If a single member contains all the F and I specifications for a file or a set of files, only that member needs to be translated.

- The amount of time required to translate the source specifications and create definitions depends on the number of descriptions contained in the specifications. If there are an extremely large number of descriptions to be translated, the IDDUXLAT procedure might degrade overall system performance when the procedure is run in command mode. Therefore, if a source member contains more than 150 descriptions, consider running IDDUXLAT in batch mode by submitting the procedure to the job queue.

- Record formats are not specifically named by the source specifications. The IDDUXLAT procedure, therefore, generates format definition names using the specified definition prefix name followed by a 4-digit sequence number.

- The IDDUXLAT procedure uses the file name specified in the source specifications as the name of the created file definition. If a file definition already exists by that name, IDDUXLAT generates a name using the specified definition prefix name followed by a 6-digit sequence number.

- When translating an RPG auto report source member, the /COPY function is ignored.

- IDDUXLAT requires that TMS data definitions be contained in the library named #DATADEF. If the TMS data definitions you want to translate are not contained in #DATADEF, use the LIBRLIBR procedure to copy them to #DATADEF.

  When translating a TMS data definition, the IDDUXLAT procedure uses the data definition name as the name of the first format definition that it creates; thereafter, all format names are generated using the specified definition prefix name followed by a 4-digit sequence number. IDDUXLAT uses the description fields in the I specifications as the text of the short comments and as the text of the column headings for the corresponding field definitions.

- All long comments for definitions created by IDDUXLAT contain:

  – An indication that the definition was created by the IDDUXLAT procedure, as well as the parameters used

  – The date and time the definition was created

  – For field and file definitions, the original name of the field or file as specified in the source specifications

  – The name of the library that contains the source member

  – The name of the source member containing the specifications

  – The program ID found in the H specification (if available)

  – The names of any other definitions that use that definition (for example, the name of the file and format definitions that use a field definition)

  The following is an example of a long comment contained in a field definition created by IDDUXLAT:

```
FIELD  - CUSNRX
Created by IDDUXLAT DICTRPG ,ACTREC  ,        ,RPGLIB  ,RPG ,FI,PRINT
Date - 100384   Time - 1037
Original Name - CUSNRX
Library - RPGLIB     Source Member - ACTREC
RPG Program ID - ACTREC
In File - DMASTER
In Format - FI0004
```

- If the IDDUXLAT procedure finds an error in the source specifications, or is unable to translate the specifications, warning and terminating messages are printed.

  A warning message is printed if IDDUXLAT can continue processing the source member. If IDDUXLAT is unable to translate the member, no changes are made to the output data dictionary, any unprocessed specifications in the member are ignored, a terminating message is placed in the printed output, and the next requested source member is processed.

Rules that Apply to the Source Specifications

The following describes how the IDDUXLAT procedure interprets the source specifications.

Rules for Interpreting the H (Control) Specifications

- All H specifications containing an * (asterisk) or a / (slash) in column 7 are ignored. A warning message is printed if an RPG auto report /COPY command is found.
- If an H specification does not contain an * or a / in column 7, the program ID in columns 75 through 80 is saved for use in the long comment for all definitions created by IDDUXLAT.
- If no H specification is found, or if the program ID entry in the first H specification is blank or invalid, no program ID is saved. The definitions created by IDDUXLAT are not affected and translation continues.

Rules for Interpreting the F (File) Specifications

- If translating a TMS data definition, "* DEF-" in columns 7 through 12 and "-" in column 43 are interpreted as a special TMS F specification. These entries are interpreted as a record format name, if columns 44 through 51 match a subsequent file specification.

  Columns 13 through 20 must contain a data definition name, and columns 44 through 51 must contain a file name.

- The following must be true:

  - The first position of the file name entry (columns 7 through 14) must contain an alphabetic character (A through Z, #, $, or @).
  - The file type entry (column 15) must be either I (input) or U (update).
  - The file designation entry (column 16) must not be R (record address file) or T (table file).
  - The first four characters (columns 40 through 43) of the device entry must be DISK.

  Any F specification not meeting the above is ignored; a warning message is printed.

- RPG auto report /COPY commands are ignored; a warning message is printed if a /COPY command is found.
- The following entries are checked for adherence to proper syntax:

  - The file name specified in columns 7 through 14. If a file name was used in a previous F specification, the current F specification is ignored.
  - The record length entry must be in the range 1 through 4096.
  - If translating a TMS data definition, the file's brief description (columns 75 through 98) is saved and is used in the short comment for the corresponding file definition.

Rules for Interpreting the I (Input) Specifications

I specifications are categorized in the following manner:

- Comment, compiler listing directive, and auto report specifications (an * or a / is in column 7)
- File and record type identification specifications. This category of I specifications includes:
  - File name I specifications
  - Sequence I specifications
  - AND line specifications (record identification code extension)
  - OR line specifications (record identification code extension)
  - OR line specifications (joint format description)
- Field description I specifications
- Data structure statement I specifications
- Data structure field description I specifications

The following rules apply to the I specifications:

- All comment, compiler listing directive, and auto report I specifications are ignored. A warning message is printed if an auto report /COPY command is found. Also, a message is displayed that gives you the option to continue processing or cancel the IDDUXLAT procedure.
- Conventions for file and record type I specifications are:
  - A file name I specification is interpreted as starting a new sequence of specifications associated with a specified file name. If that file name was not specified in a previous F specification, all I specifications associated with the file name are ignored.
  - Each subsequent sequence I specification indicates the end of a record format description and the beginning of a new record format description for the same file. A separate format definition is created for each record format description.
  - The specification following a file name or sequence I specification can be an AND or OR line or a file description I specification. A maximum of 20 OR lines can follow a file name or sequence I specification. Record identification indicators (columns 21 through 27, 28 through 34, and 35 through 41) are used to identify record format descriptions in a multiple-format file. A maximum of 70 record identification codes can be associated with any one file name or sequence specification.
  - Field record relation indicators (columns 63 and 64) are used to associate field description I specifications with a particular record format. These variations in record formats are called joint format descriptions. IDDUXLAT creates a separate format definition for each joint format description.

- Conventions for field description I specifications are:
  - The data type is specified in column 43.
  - The field location FROM position (columns 44 through 47) and TO position (columns 48 through 51) indicate the position of the field in the record format. The FROM and TO positions must be less than or equal to the record length of the file. The FROM position must be less than or equal to the TO position.

    The field length must be in the range 1 through 4096 for character data, 1 through 15 for zoned decimal data, 1 through 8 for packed decimal data, and 2 or 4 for binary data.
  - For numeric data types, the number of decimal positions (column 62) must be less than or equal to the field length. For character data, the data type entry (column 43) must be blank and the decimal positions entry must be blank.
  - The field name is in columns 53-58. This entry also might contain a data structure statement or a data structure field name.
  - When an OR relation is used between field names, the field record relation entry in columns 63 through 64 is used to associate a particular field description with a particular record format description.
  - If translating a TMS data definition, the field description can appear in columns 75 through 90; the field description becomes the column heading(s) for the corresponding field definition.

    If an H appears in column 93, the field contains hexadecimal data.

    An edit code can appear in column 94. The following are allowed edit codes:

    C Decimal point character is a comma

    D Decimal point character is a period

    Y Date field

    Z Leading zeros are not shown
- Conventions associated with data structure statement and data structure field description I specifications are:
  - A data structure description is made up of a data structure I specification possibly followed by a one or more data structure field description I specifications. The IDDUXLAT procedure recognizes a data structure only if a data structure name appears.
  - Data structures can be used to redefine fields described by field description I specifications. A field is redefined in this manner by using the data structure name as the field name.
  - If a data structure redefines a character or a zoned decimal numeric field, the field descriptions in the data structure are used as regular field descriptions.

General Rules for Interpreting the Source Specifications

- All imbedded diagnostic message lines (?? in columns 12 and 13) are ignored by the IDDUXLAT procedure.
- The source is considered ended when the last specification is translated, or when an ** or a /* is found in columns 1 through 3.
- F specifications without corresponding I specifications are ignored. I specifications must follow their corresponding F specification.
- File names must not be duplicated in the F specifications; however, a file name can be the same as a field name or a data structure name. An F specification with a duplicate file name is ignored.
- A field name can duplicate a data structure name if 1) the field does not appear in the data structure and 2) the field length is greater than or equal to the total length of all the fields in the data structure.
- Field names can be duplicated if their lengths, data types, and decimal positions are also the same. If these field attributes are not the same, a warning message is printed. If duplicate field names appear within the same format, a new field definition name is created for all but the first field. New field names are generated using the definition prefix name specified in the IDDUXLAT procedure followed by a 4-digit sequence number.
- Data structure names must not be duplicated, and must not be the same as a data structure field name.
- Overlapping and structured fields are not supported by the IDDUXLAT procedure. If either is encountered, the following occurs:
  - Non-overlapping fields occupying lower record format positions (lesser FROM position values) are used before fields occupying higher record format positions (greater FROM position values).
  - If two or more fields have the same FROM position, the first field with the shortest length is used.
  - Filler fields are used to define areas of a record format not specifically described by field description I specifications.
- If a record ID code is defined for a record position that is not within a field defined by an I specification, the record ID code is not translated and is not placed in the corresponding format definition.

Example

In the following example, the specifications in an RPG source member named ACTREC, contained in a library named RPGLIB, are translated into IDDU definitions. The IDDU definitions are stored in an IDDU data dictionary named DICTRPG.

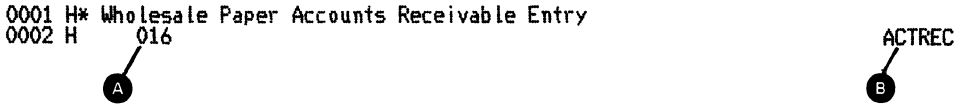
ACTREC contains file description specifications for several disk files, as well as a work station file. For the purposes of this example, you will see only the information pertaining to the disk file named DMASTER. You will be shown the source specifications for DMASTER, the parameters used in the IDDUXLAT procedure, and portions of the listing printed by the IDDUXLAT procedure.

Input to IDDUXLAT: RPG Source Specifications

The following describes the RPG source specifications pertaining to the disk file named DMASTER and to the definitions to be created from those specifications.

H Specifications

The following are the H (control) specifications contained in ACTREC:



The first H specification is ignored because it contains an * (comment statement) in column 7.

The second H specification indicates (A) the amount of main storage to be available to run the compiled program, and (B) the name assigned to the program. The program name is used in the long comment for all definitions created by IDDUXLAT.

F Specifications

The following are several of the F (file) specifications contained in ACTREC:

```
0009 FSCREEN  CP  F     180            WORKSTN                          ACTREC
0010 F                                                KNUM       4     ACTREC
0011 F                                                KSAVDS SCRDS     ACTREC
0012 F                                                KID    XD        ACTREC
0013 F                                                KIND       10    ACTREC
                    :
                    :
                    :
0055 FDFILE01 IC  F 222 222R 6AI     4 DISK                 150       ACTREC
0056 FDFILE02 IC  F  64  64R 2AI     4 DISK                           ACTREC
0057 FDFILE03 UC  F  64  64R 6AI     3 DISK                           ACTREC
0058 FDMASTER UC  F  53  53R           DISK                           ACTREC
     (A)      (B)                     (C)
```

The F specifications in program line numbers 0009 through 0013 describe a work station file; these specifications will be ignored by the IDDUXLAT procedure.

The F specification in program line number 0058 describes the file named DMASTER. The entries in this F specification indicate that (A) DMASTER is (B) an update chained (C) disk file.

I Specifications

The following are the I (input) specifications contained in ACTREC that pertain to the file named DMASTER:

```
0069 IDMASTER NS  13   2 CH   3 CO                                    ACTREC
0070 I       OR   14   2 CH   3 CB                                    ACTREC
0071 I                                      1   3 CODEX               ACTREC
0072 I                 (A)                  4   90CUSNRX              ACTREC
0073 I                                      10  150INVNRX     13 ┐    ACTREC
0074 I                                  P   17  200DATEX      13 ├(B) ACTREC
0075 I                                  P   10  130DATEX      14 │    ACTREC
0076 I                                      15  200INVNRX     14 ┘    ACTREC
0077 I                                      21  21 SALCDX             ACTREC
0078 I                                  P   22  252INVAMX             ACTREC
0079 I                                  P   26  292CDSALX             ACTREC
0080 I                                  P   30  332CNSAMX             ACTREC
0081 I                                  P   34  372INCSTX             ACTREC
0082 I                                      38  390SLSNRX             ACTREC
0083 I                                      41  430PRECNX             ACTREC
0084 I                                      44  460NXRCNX             ACTREC
0085 I       NS  15   2 CO   3 CO                                    ACTREC
0086 I       OR  16   2 CO   3 CB                                    ACTREC
0087 I                                      1   3 CODEX               ACTREC
0088 I                                      4   90CUSNRX              ACTREC
0089 I                                      10  150INVNRX     15      ACTREC
0090 I                                  P   17  200DATEX      15      ACTREC
0091 I                                  P   10  130DATEX      16      ACTREC
0092 I                                      15  200INVNRX     16      ACTREC
0093 I                                  P   21  242AMTRCX             ACTREC
0094 I                                  P   25  282CDSAMX             ACTREC
0095 I                                  P   29  322TTLCRX             ACTREC
0096 I                                      41  430PRECNX             ACTREC
0097 I                                      44  460NXRCNX             ACTREC
0098 I       NS  17   2 CP   3 CO                                    ACTREC
0099 I       OR  18   2 CP   3 CB                                    ACTREC
0100 I                                      1   3 CODEX               ACTREC
0101 I                                      4   90CUSNRX              ACTREC
0102 I                                      10  150INVNRX     17      ACTREC
0103 I                                  P   17  200DATEX      17      ACTREC
0104 I                                  P   10  130DATEX      18      ACTREC
0105 I                                      15  200INVNRX     18      ACTREC
0106 I                                  P   21  242TDCRAX             ACTREC
0107 I                                      25  30 ADJNRX             ACTREC
0108 I                                      41  430PRECNX             ACTREC
0109 I                                      44  460NXRCNX             ACTREC
```

The I specifications for DMASTER use a combination of (A) record identifying indicators and (B) field record relation indicators to identify a total of six record format descriptions. Consequently, six format definitions will be created for DMASTER.

The IDDUXLAT Parameters This Example Uses

The following IDDUXLAT procedure command is entered:

```
IDDUXLAT DICTRPG,ACTREC,,RPGLIB,RPG,FI,PRINT
```

DICTRPG specifies that the definitions created by IDDUXLAT are stored in the data dictionary named DICTRPG.

ACTREC specifies that the translated specifications are contained in the source member named ACTREC.

RPGLIB specifies that ACTREC is contained in the library named RPGLIB.

RPG specifies that ACTREC contains RPG source specifications.

FI specifies that if a field name is the same as the name of an existing definition, IDDUXLAT generates a field definition name in the form FInnnn, where nnnn is a 4-digit sequence number. Generated format definition names are also in the form FInnnn. If a file name is the same as the name of an existing file definition, the generated file definition name is in the form FInnnnnn.

PRINT specifies that all possible printed output is requested.

Output from IDDUXLAT: The PRINT Option

The following describes the printed output received from the IDDUXLAT procedure.

Section 1: The Specified IDDUXLAT Parameters

Section 1 of the printed output lists the parameters used in the IDDUXLAT procedure command:

```
**************************** S/36 IDDUXLAT SECTION 1 **************************
ACTREC  -RPGLIB                                  10/02/84   15:24   Page-   1


                RPG II F&I Specification Translation Aid

                     Data Dictionary Name ------ DICTRPG
                     Source Library Name ------- RPGLIB
                     Source Member Name -------- ACTREC
                     TEXT/Subtype Entry -------- RPG
                     Field Name Prefix --------- FI
                     Print Option -------------- PRINT
```

Section 2: The RPG Source Member Listing and Messages

Section 2 of the printed output lists the source specifications translated by the IDDUXLAT procedure. The following is a portion of Section 2 as printed for this example:

```
**************************** S/36 IDDUXLAT SECTION 2 ***************************

ACTREC  -RPGLIB                                  10/02/84   15:24   Page-   2

RPG II Source Translated:

   000001   0001 H* Wholesale Paper Accounts Receivable Entry
   000002   0002 H     016                                                    ACTREC
   000003   0003 F*************************************************************ACTREC
   000004   0004 F*  THIS VERSION OF ACTREC CONTAINS THE FILE DESCRIPTION STATEMENTS
   000005   0005 F*  FOR ALL DATA BASES USED IN WHOLESALE PAPER TESTING.
   000006   0006 F*  4         TERMINALS SUPPORTED, MAXIMUM                   ACTREC
   000007   0007 F*  08/06/84    DATE LAST CHANGED BY LAG
   000008   0008 F*************************************************************ACTREC
   000009   0009 FSCREEN  CP  F     180            WORKSTN                    ACTREC
   ******   F-Spec is not relevant to this translation - Ignored
   000010   0010 F                                          KNUM        4     ACTREC
   ******   Invalid File name on F-Spec - Ignored
   000011   0011 F                                          KSAVDS SCRDS      ACTREC
   ******   Invalid File name on F-Spec - Ignored
   000012   0012 F                                          KID    XD         ACTREC
   ******   Invalid File name on F-Spec - Ignored
   000013   0013 F                                          KIND       10     ACTREC
   ******   Invalid File name on F-Spec - Ignored
```

The IDDUXLAT procedure recognized that the F specifications in program line numbers 0009 through 0013 describe a work station file. Because the IDDUXLAT procedure translates the F specifications for disk files only, several printed messages indicate that these specifications were not relevant and were ignored.

The F and I specifications for the disk file named DMASTER were translated without error.

Section 3: The IDDU Definitions Created by IDDUXLAT

Section 3 of the printed output briefly describes each of the format definitions created by the IDDUXLAT procedure. The following shows the description of one of the six format definitions created for DMASTER.

```
**************************** S/36 IDDUXLAT SECTION 3 ***************************

ACTREC  -RPGLIB                                  10/02/84   15:24   Page-  10

Data Dictionary Definitions Generated Cont:

(A) File - DMASTER
    (B) Format - FI0004

          Fields  Start  Length  Dec   Type  Description
                   (D)     (E)    (F)  (G)
      (C) CODEX      1       3     0     C
          CUSNRX     4       6     0     Z
          INVNRX    10       6     0     Z
          *         16       1           C   Filler Field
          DATEX     17       4     0     P
          SALCDX    21       1           C
          INVAMX    22       4     2     P
          CDSALX    26       4     2     P
          CNSAMX    30       4     2     P
          INCSTX    34       4     2     P
          SLSNRX    38       2     0     Z
          *         40       1           C   Filler Field
          PRECNX    41       3     0     Z
          NXRCNX    44       3     0     Z
          *         47       7           C   Filler Field

    Format - FI0004   placed in dictionary
```

Each description identifies:

(A) The name of the file definition that uses the format definition. In this example, the file definition name (DMASTER) is the same as the file name used in the F specification.

(B) The name of the format definition created by IDDUXLAT. In this example, IDDUXLAT used the specified definition prefix name (FI) and added a 4-digit sequence number. FI0004 was the fourth created format definition. (Although not shown in this example, IDDUXLAT previously created three format definitions for the other disk files used by ACTREC.)

(C) A list of the field definitions created by IDDUXLAT that are used by the format definition. In this example, IDDUXLAT used the field names specified in the I specifications as the names of the field definitions.

An * (asterisk) in the *Fields* column indicates that a filler field was created for an area in the record format that was not specifically described by an I specification. Field definitions are not created for filler fields.

(D) The starting position of each field definition used by the format definition.

(E) The length of each field described by a field definition.

Ⓕ The number of decimal positions in each field, if numeric.

Ⓖ The data type of each field. The following abbreviations are used:

C Character data

B Binary numeric data

H Hexadecimal data

P Packed decimal numeric data

Z Zoned decimal numeric data

Section 4: Statistics and Messages

The following shows Section 4 of the printed output:

```
************************** S/36 IDDUXLAT SECTION 4 **************************
ACTREC  -RPGLIB                                  10/02/84   15:24   Page-  17

  Ⓐ Translation Was Successful

       Source Member errors:


         Ⓑ Number of Warnings --------    7

Definitions Created                File   Format  Field
     Added to Dictionary -------     4       10     23 ─Ⓒ
     Existing Definitions Used -     0        0     42 ─Ⓓ

                *****END OF TRANSLATION*****
```

Ⓐ IDDUXLAT successfully created all possible definitions.

Ⓑ A total of seven warning errors were found and printed. In addition to the five errors shown previously, two others not shown in this example were also found.

Ⓒ A total of four new file definitions, 10 new format definitions, and 23 new field definitions were created.

Ⓓ Some fields appeared several times within the source specifications. In some cases, a field had exactly the same characteristics (the same data type, field length, and number of decimal positions) each time it appeared. If IDDUXLAT had already created a field definition for that field, IDDUXLAT reused the field definition. In this example, field definitions were reused a total of 42 times.

Glossary

#LIBRARY. The library, provided with the system, that contains the System Support Program Product. See *system library.*

abnormal termination. A system failure or operator action that causes a job to end unsuccessfully.

access. To go to or reach; to get at.

access level. The level of authority an operator has in order to use a secured file, library, folder, or folder member.

access method. The way that records in files are referred to by the system. The reference can be consecutive (records are referred to one after another in the order in which they appear in the file), or it can be random (the individual records can be referred to in any order).

acquire. To assign a display station or session to a program.

acquired session. A session that has been started by a System/36 program using an acquire operation, or in BASIC, using an OPEN statement.

action item. A piece of mail that has a due date but has not yet been answered.

activity queue. A list of messages and batch jobs that are to be sent or submitted at a specific date and time.

adapter. See *communications adapter.*

address. (1) A name, label, or number that identifies a location in storage, a device in a network, or any other data source. (2) In Personal Services/36, an 8-byte code required for sign-on and the distribution of mail.

addressing. (1) In data communications, the way that the sending or control station selects the station to which it is sending data. (2) A means of identifying storage locations.

advanced peer-to-peer networking (APPN). A communications feature that routes data in a network between two or more APPC systems that are not directly attached. See also *node* and *network.*

advanced program-to-program communications (APPC). Communications support that allows System/36 to communicate with other systems having the same support. APPC is the way that System/36 puts the IBM SNA LU-6.2 protocol into effect.

alarm. An audible signal at a display station or printer that is used to get the operator's attention.

alert. A record sent to another system to communicate a problem or an impending problem. On System/36, the problem management portion of the Communications and Systems Management feature used to generate and send alerts.

alert indicator. The indicator that specifies whether or not the System/36 should send an alert to a host system when an IBM-supplied or user-defined message is issued from a message member.

align. To bring into or be in line with another or with others. For example, to align numbers on the decimal point.

allocate. To assign a resource, such as a disk file or a diskette file, to perform a specific task.

alphabetic character. Any one of the letters A through Z (uppercase and lowercase). Some program products extend the alphabet to include the special characters #, $, and @.

alphameric. Consisting of letters, numbers, and often other symbols, such as punctuation marks and mathematical symbols.

alphanumeric. See *alphameric.*

alternative cylinder. A disk cylinder that is made available by the computer in place of a cylinder that cannot be used.

alternative index. An index that is built after a physical file is created and that provides a different order for reading or writing records in the file. Contrast with *primary index.*

alternative line. A secondary switched line to which a remote controller can be attached if the primary communications line is not available.

alternative sector. A disk sector that is made available by the system in place of a sector that cannot be used. See *sector.*

alternative system console. A command display station that can be designated as the system console.

American National Standard Code for Information Interchange (ASCII). The code developed by ANSI for information interchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters.

American National Standards Institute (ANSI). An organization sponsored by the Computer and Business Equipment Manufacturers Association for establishing voluntary industry standards.

ANSI. See *American National Standards Institute (ANSI).*

APAR. See *authorized program analysis report (APAR).*

APPC. See *advanced program-to-program communications (APPC).*

application. (1) A particular business task, such as inventory control or accounts receivable. (2) A group of related programs that apply to a particular business area, such as the Inventory Control or the Accounts Receivable application.

application program. A program used to perform an application or part of an application.

APPN. See *advanced peer-to-peer networking (APPN).*

archive. To copy a folder member onto disk, tape, or diskette.

archived member. A folder member that has been saved on diskette, disk, or tape.

argument. An expression that is passed to a function or subroutine for evaluation.

arithmetic expression. A statement containing any combination of data items joined together by one or more arithmetic operators in such a way that the statement can be evaluated as a single numeric value.

arithmetic operator. A symbol used to represent a mathematical operation, such as + or -, used to indicate addition or subtraction.

ascending key sequence. The arrangement of data in order from the lowest value of the key field to the highest value of the key field. Contrast with *descending key sequence.*

ascending sequence. The arrangement of data in order from the lowest value to the highest value, according to the rules for comparing data. Contrast with *descending sequence.*

ASCII. See *American National Standard Code for Information Interchange (ASCII).*

assembler. A program that converts assembler language statements to machine instructions.

assign/free area. An area of main storage that contains control information for all system activity and for each job that is active.

assignment. The process of giving values to variables.

asynchronous transmission. In data communications, a method of transmission in which the bits included in a character or block of characters occur during a specific time interval. However, the start of each character or block of characters can occur at any time during this interval. Contrast with *synchronous transmission.*

attribute. A characteristic. For example, an attribute for a displayed field could be blinking.

audit trail. Information that allows the history of things such as a customer account or item record to be traced. The more recent information can be stored in the computer.

audit window. A field in the status line that displays the name of a nondisplayed character or text instruction when the cursor is positioned below the character or instruction.

authority. The right to communicate with or use a resource.

authorization list. A list of user identifications and access levels that is used to secure folders and folder members.

authorize. To allow a user to communicate with or use a resource.

authorized program analysis report (APAR). A request for correction of a defect in a current release of an IBM-supplied program.

auto report. An RPG option that simplifies the defining of formats for printed reports and that allows the previously written statements to be included in new programs.

autoanswer. In data communications, the ability of a station to receive a call over a switched line without operator action. Contrast with *manual answer.*

autocall. In data communications, the ability of a station to place a call over a switched line without operator action. Contrast with *manual call.*

autocall unit. A common carrier device that allows System/36 to automatically call a remote location.

autoduplication. An option of DFU that allows information from a previous record to be duplicated in the current record.

automatic response severity level. The value that indicates whether messages should be automatically responded to by the System Support Program Product.

autowriter. A System Support Program Product option that causes the spool writer program to be loaded without operator action whenever output exists in the spool file. See also *spool writer.*

back up. To copy information, usually onto diskette or tape, for safekeeping.

backspace. To move the cursor one character position backward.

backup copy. A copy, usually of a file, library member, or folder, that is kept in case the original is unintentionally changed or destroyed.

badge security. A System Support Program Product option that helps prevent the unauthorized use of a display station by checking the data from a magnetic stripe on a badge before allowing an operator to sign on.

BASIC (beginner's all-purpose symbolic instruction code). A programming language designed for interactive systems and originally developed at Dartmouth College to encourage people to use computers for simple problem-solving operations.

basic data exchange. A file format for exchanging data on diskettes between systems or devices.

basic ideographic character set. A character set defined by IBM that contains 3226 Kanji and 481 additional characters. The additional characters include Katakana, Hiragana, the alphabet (A through Z and a through z), numbers (0 through 9), Roman numerals (I through X), Greek, Cyrillic, and special symbols. Contrast with *extended ideographic character set*; see also *ideographic character set.*

BASIC procedure. A set of BASIC commands, BASIC statements, input data, and/or comments that cause a specific operation or set of operations to be performed in BASIC.

batch. Pertaining to activity involving little or no operator action. Contrast with *interactive.*

batch BSC. The System Support Program Product support that provides data communications with BSC computers and devices via the RPG T specification or the assembler $DTFB macroinstruction.

batch compilation. A method of compiling programs without the continual attention of an operator.

batch processing. A processing method in which a program or programs process records with little or no operator action. Contrast with *interactive processing.*

beginning of tape. A reflective marking near the beginning of a tape reel that indicates where the system can begin recording data.

BGU/36. See *Business Graphics Utilities/36 (BGU/36).*

binary. (1) Pertaining to a system of numbers to the base two; the binary digits are 0 and 1. (2) Involving a choice of two conditions, such as on-off or yes-no.

binary synchronous communications (BSC). A form of communications line control that uses transmission control characters to control the transfer of data over a communications line. Compare with *synchronous data link control (SDLC).*

bit. Either of the binary digits 0 or 1. See also *byte.*

block. (1) A group of records that is recorded or processed as a unit. Same as *physical record.* (2) Ten sectors (2560 bytes) of disk storage. (3) In data communications, a group of records that is recorded, processed, or sent as a unit.

bps. Bits per second.

branching. Performing a statement other than the next one in sequence.

BSC. See *binary synchronous communications (BSC).*

BSCEL (binary synchronous communications equivalence link) subsystem. The SSP-ICF subsystem that provides BSC communications with another System/36 and many other BSC computers and devices.

buffer. (1) A temporary storage unit, especially one that accepts information at one rate and delivers it at another rate. (2) An area of storage, temporarily reserved for performing input or output, into which data is read or from which data is written.

Business Graphics Utilities/36 (BGU/36). A program product that can be used to design, display, print, and plot graphics.

busy printer. A printer whose spool writer is actively printing a spool file entry; or a printer whose spool writer is stopped but has spool file entries that are not being used by the COPYPRT procedure, not in a held status, and not in the process of being deferred for printing until the spool file program has finished intercepting them.

byte. The amount of storage required to represent one character; a byte is 8 bits.

C. Celsius.

C & SM. See *Communications and Systems Management (C & SM).*

cable thru. A standard function or special feature that allows multiple work stations to be attached to a particular line.

cache. A fixed user area of main storage that contains recently accessed disk data.

cache page. The smallest amount of contiguous disk data that can be held in a cache.

calendar. A list or schedule of appointments, reminders, and programs.

calendar item. An appointment, reminder, or program scheduled in a calendar.

calendar view. A way of showing calendar items. Appointments can be viewed for a day, a week, or weeks combined from several calendars. The entire calendar (including reminders and programs in addition to appointments) can also be viewed.

call. (1) To activate a program or procedure at its entry point. Compare with *load.* (2) In data communications, the action necessary in making a connection between two stations on a switched line.

cancel. To end a task before it is completed.

carrier. A continuous frequency that can be modulated with a second (information-carrying) signal.

CCITT. Consultative Committee on International Telegraphy and Telephone.

CCP. See *communications control program (CCP).*

CCP subsystem. The SSP-ICF subsystem that provides data communications with a System/3 Model 15D.

CGU. See *character generator utility (CGU).*

change management. The part of the Communications and Systems Management feature that allows a host system operator to send (via DSX) programming changes and new programs to System/36, and to start procedures on System/36.

character. A letter, digit, or other symbol.

character generator utility (CGU). A program that is used to create, maintain, and display ideographic characters.

character key. A keyboard key that allows the user to enter the character shown on the key. Compare with *command key* and *function key.*

character set. A group of characters used for a specific reason; for example, the set of characters a printer can print.

character string. A sequence of consecutive characters.

characters per inch (CPI). The number of characters printed within an inch horizontally across a page.

chart. A display screen, printed page, or plotted page that contains multiple graphs.

chart member. A member that contains all the graph format member names and data member names defined for multiple charts.

chart utility. The part of BGU/36 that allows you to design, display, print, and plot charts and maintain chart members.

check. (1) An error condition. (2) To look for a condition.

child. Pertaining to a secured resource, either a file or library, that uses the user list of a parent resource. A child resource can have only one parent resource. Contrast with *parent.*

CICS subsystem. The SSP-ICF subsystem that allows binary synchronous communications with CICS/VS.

CICS/VS. Customer Information Control System, which operates on a host system such as a System/370, or a 30XX or 43XX processor.

clocking. In data communications, a method of controlling the number of data bits sent on a communications line in a given time.

close. To end the processing of a file.

closed user group. A group of DTEs that can access only one another. DTEs outside of the group can neither access nor be accessed by members of the group.

COBOL (common business-oriented language). A high-level programming language, similar to English, that is used primarily for commercial data processing.

code. (1) Instructions for the computer. (2) To write instructions for the computer. Same as *program*. (3) A representation of a condition, such as an error code.

collating sequence. The sequence in which characters are ordered within the computer for sorting, combining, or comparing.

color palette. The range of colors defined by hue, lightness, and saturation to be used when a graph is displayed on a graphics-capable display station.

column separator. A symbol on each side of a position of a field on a display. This symbol does not occupy a position on the display.

command. A request to the system to perform an operation or a procedure.

command display. A display that allows an operator to display and send messages, and use control commands and procedure commands to start and control jobs. Contrast with *standby display*. See also *console display* and *subconsole display*.

command display station. A display station from which an operator can start and control jobs. A command display station can become an alternative system console, can be designated as a subconsole, and can also be used as a data display station. See also *alternative system console*, *data display station*, and *subconsole*.

command file. In the MSRJE utility, a disk file, procedure member, or source member that can contain MSRJE utility control statements and records to be transmitted to the host system. Contrast with *data file*.

command key. A keyboard key that is used to request specific programmed actions. Compare with *character key* and *function key*.

command line. The blank line on a display where commands or option numbers can be entered.

command processor. The part of the System Support Program Product that processes control commands and that passes procedure commands and operation control language statements to the initiator.

comment. Words or statements in a program or on a display that serve as documentation rather than as instructions, choices, or prompts.

common carrier. In data communications, any government-regulated company that provides communication services to the general public.

communications. See *data communications*.

communications adapter. A hardware feature that enables a computer or device to become a part of a data communications network.

Communications and Systems Management (C & SM). A feature of the System Support Program Product that contains the remote management support (also referred to as DHCF), the change management support (referred to as DSNX), and the problem management support (referred to as alerts).

communications control program (CCP). An IBM System/3 Model 15 program that allows communications between System/3 and the SSP-ICF CCP subsystem.

communications file. A file that describes an advanced program-to-program communications (APPC) subsystem session between a System/36 program and a remote device, another program, or another system.

communications file definition. The format in the communications file that contains the APPC subsystem session description.

communications line. The line over which data communications takes place; for example, a telephone line.

communications link. See *data link*.

communications queue. A list that keeps track of mail to be sent to users on remote systems.

communications queue definition. A table that specifies three characteristics of a communications queue: the number of items sent in a transmission; the interval between retrying failed transmissions; and the length of time that an item that cannot be transmitted is is left on the queue before it is removed.

communications routing table. A table of remote location and session group pairs used for sending mail.

communications subsystem. See *subsystem*.

compile. To translate a program written in a high-level programming language into a machine language program.

compiler. A program that compiles.

compress. (1) To move files, libraries, or folders together on disk to create one continuous area of unused space. (2) To replace repetitive characters in a file or folder with control characters so that the file or folder takes up less space when saved on diskette.

compression. In data communications, a technique for removing strings of duplicate characters and for removing trailing blanks before transmitting data.

computer graphics. The use of a computer to produce pictorial representations of relationships, such as charts, and two- or three-dimensional images, by means of dots, lines, curves, and so forth.

computer output reduction (COR). A 3812 Printer capability that allows output normally printed on 14-inch paper to print on 8-1/2 x 11 inch paper.

concatenate. (1) To link together. (2) To join two character strings.

condense. To move library members together in a library to create one continuous area of unused space in the library.

condition. An expression in a program or procedure that can be evaluated to a value of either true or false when the program or procedure is running.

conditional expression. A logical statement that describes the relationship (such as greater than or equal) of two items.

configuration. The group of machines, devices, and programs that make up a data processing system. See also *system configuration.*

configuration member. (1) A library member that describes the devices, programming, and characteristics of the system. (2) In data communications, a member that defines the attributes of a communications subsystem or line.

configure. (1) To describe (to the system) the devices, optional features, and program products installed on a system. (2) To describe to SSP-ICF both the communications facilities connected to System/36 and the attributes of the subsystem and remote system.

consecutive processing. The processing of records in the order in which they exist in a file. Same as *sequential processing.* See also *random processing.*

console. A device used for communication between an operator and the system.

console display. A display that can be requested only at the system console. From a console display an operator can display, send, and reply to messages and use all control commands.

constant. A data item with a value that does not change. Contrast with *variable.*

contiguous. Being in actual contact.

continuation line. A line of a source statement into which characters are entered when the source statement cannot be contained on the previous line or lines.

control command. A command used by an operator to control the system or a work station. A control command does not run a procedure and cannot be used in a procedure.

control panel. A panel that contains lights and keys used to observe and operate the status of the operations within the system.

control station. The primary or controlling computer on a multipoint line. The control station controls the sending and receiving of data.

control storage. Storage in the computer that contains the programs used to control input and output operations and the use of main storage. Contrast with *main storage.*

control storage initial program load. The loading of control storage programs from disk or diskette to control storage.

control storage processor. The hardware that performs control storage instructions to handle data transfer and main storage, and input/output assignments.

controlled cancel. The system action that ends the job step being run and saves any new data already created. The job that is running can continue with the next job step.

controller. Circuitry or a device used to coordinate and control the operation of one or more devices.

COR. See *computer output reduction.*

counter. A register or storage location used to accumulate the number of occurrences of an event.

coupler. A device that connects a modem to a telephone network.

CPI. See *characters per inch (CPI).*

cradle. The part of a telephone that holds the handset.

creation date. The program date at the time a file is created. See also *program date*, *session date*, and *system date*.

current library. The first library searched for any required members. The current library can be specified during sign-on or while running programs and procedures.

cursor. A movable symbol on a display, used to indicate to the operator where to type the next character.

cylinder. All disk or diskette tracks that can be read or written without moving the disk drive or diskette drive read/write mechanism.

data circuit-terminating equipment (DCE). The equipment installed at the user's location that provides all the functions required to establish, maintain, and terminate a connection, and the signal conversion and coding between the data terminal equipment (DTE) and the line.

data communications. The transmission of data between computers and/or remote devices (usually over a long distance).

data definition. Information that describes the contents and characteristics of a field, format (record), or file. A data definition can include such things as field names, lengths, and data types. See also *field definition*, *file definition*, and *format definition*.

data dictionary. A folder that contains field, format, and file definitions.

data display station. A display station from which an operator can only enter data. A data display station is acquired and controlled by a program. Contrast with *command display station*.

Data Encryption Subroutine. A feature of the System Support Program Product that codes and decodes data for security purposes. This subroutine is only used by the SSP-ICF Finance subsystem.

data entry facility. A function of Query/36 that allows a user to add, change, and mark records to be deleted in a file. The file must be linked to a file definition created with IDDU.

data file. In the MSRJE utility, a disk file, procedure member, or source member that can contain only records to be transmitted to the host system. Contrast with *command file*.

data file utility (DFU). The part of the Utilities Program Product that is used to create, maintain, display, and print disk files.

data link. The equipment and rules (protocols) used for sending and receiving data.

data link escape (DLE) character. In BSC, a transmission control character usually used in transparent text mode to indicate that the next character is a transmission control character.

data management. See *disk data management*.

data member. See *graph data member*.

data merge. See *data/text merge*.

data mode. In data communications, a time during which BSC is sending or receiving characters on the communications line.

data stream. All information (data and control information) transmitted over a data link.

data terminal equipment (DTE). The data processing unit that uses communications lines.

data type. A category that identifies the mathematical qualities and internal representation of data.

data/text merge. The process of combining data from a file (such as names and addresses) with the text of a document.

DCE. See *data circuit-terminating equipment (DCE)*.

DDFF. See *Distributed Disk File Facility (DDFF)*.

DDM. See *Distributed Data Management (DDM)*.

DDSA. See *digital data service adapter (DDSA)*.

deactivate. To make ineffective. For example, to deactivate security.

debug. To detect, locate, and remove errors from a program.

decimal. (1) Pertaining to a system of numbers to the base ten; decimal digits range from 0 through 9. (2) A proper fraction in which the denominator is a power of 10.

dedicated system. A system intentionally allocated to a single job or task.

default. See *default value*.

default printer. A printer that accepts all the printed output from a display station that is assigned to it.

default prompt. A field name from a D-specification used to prompt for the field's contents.

default value. A value stored in the system that is used when no other value is specified.

define-the-file (DTF). A control block containing information that is passed between data management routines and users of the data management routines.

delete. To remove. For example, to delete a file.

delete character. A character that identifies a record to be removed from a file.

delete-capable file. A file from which records can be logically removed without compressing the file.

delimiter. A character or sequence of characters that marks the beginning or end of a unit of data.

demodulate. To set a modulated signal to its original state.

descending key sequence. The arrangement of data in order from the highest value of the key field to the lowest value of the key field. Contrast with *ascending key sequence.*

descending sequence. The arrangement of data in order from the highest value to the lowest value, according to the rules for comparing data. Contrast with *ascending sequence.*

detail record. A record that contains the daily activities or transactions of a business. For example, the items on a customer order are typically stored in detail records. Contrast with *header record.*

Development Support Utility (DSU). A program product that can be used to create, edit, remove, view, or print procedure members and source members.

device code. A two-character code used during system configuration to identify the models of display stations and printers.

DFU. See *data file utility (DFU).*

DHCF. See *Distributed Host Command Facility (DHCF).*

diagnosed-source file. A library member containing source statements and associated error messages.

diagnosed-source member. See *diagnosed-source file.*

diagnostic. Pertaining to the detection and isolation of an error.

diagnostic diskette. A diskette that contains tests to check that the system is operating properly.

diagnostic program. A computer program that recognizes, locates, and explains either a fault in equipment or a mistake in a computer program.

digit. Any of the numerals from 0 through 9.

digital data service adapter (DDSA). In data communications, a device used when transmitting data using a nonswitched digital data system. Compare with *modulator-demodulator (modem).*

direct file. A disk file in which records are referenced by the relative record number. Contrast with *indexed file* and *sequential file.*

directory. See *network resource directory (NRD).*

disable. In interactive communications, to end a subsystem and free the area of main storage used by that subsystem. Contrast with *enable.*

DISC. Disconnect.

disconnect (DISC) character. In data communications, the part of the BSC transmission control sequence for ending the connection on a switched line.

disconnect time-out. An indication that the BSC station you were communicating with has been inactive for a specified length of time and, therefore, has been disconnected.

disconnected mode. In SDLC, a response from a secondary station indicating that it is disconnected and wants to be online.

disk. A storage device made of one or more flat, circular plates with magnetic surfaces on which information can be stored.

disk data management. The System Support Program Product support that processes a request to read or write data.

disk drive. The mechanism used to read and write information on disk.

disk file. A set of related records on disk that is treated as a unit. See also *record file* and *stream file.*

diskette. A thin, flexible magnetic plate that is permanently sealed in a protective cover. It can be used to store information copied from the disk or to exchange information with other computers.

diskette drive. The mechanism used to read and write information on diskettes.

diskette magazine drive. A diskette drive that holds up to two magazines plus three individual diskettes.

diskette 1. A diskette that contains information on only one side.

diskette 2D. A diskette that contains information on both sides, and with two times the amount of information stored in the same space as a diskette 1. Therefore, a diskette 2D holds approximately four times the amount of information as a diskette 1.

display. (1) A visual presentation of information on a display screen. (2) To show information on the display screen.

display area. For ideographic support, an 18-by-18 matrix on the character definition display that is used to display the character currently being defined or updated.

display format. Data that defines (or describes) a display.

display layout sheet. A form used to plan the location of data on the display.

display screen. The part of the display station on which information is displayed.

display station. A device that includes a keyboard from which an operator can send information to the system and a display screen on which an operator can see the information sent to or the information received from the system.

Display Station Pass-Through (DSPT). A communications feature that allows a user to sign on to one System/36 from another System/36 and access that remote system's resources.

DisplayWrite/36 (DW/36). A program product that creates, revises, views, and prints documents that are produced in an office environment.

Displaywriter user. See *independent work station user.*

disposition. In file processing, the process of specifying whether a file is new, old, or shared, and how the file is to be shared.

Distributed Data Management (DDM). A feature of the System Support Program Product that allows an application program to work on files that reside on a remote system.

Distributed Disk File Facility (DDFF). A feature of the System Support Program Product that allows a System/3 or System/34 with DDFF to access disk files on System/36.

Distributed Host Command Facility (DHCF). Another name for the remote management support offered by the Communications and Systems Management feature. This support allows HCF host system users to operate System/36s in an HCF network.

Distributed Systems Executive (DSX). A program product available for IBM host systems (System/370, 43XX, and 30XX) that allows the host system to get, send, and remove files, programs, formats, and procedures in a network of computers.

Distributed Systems Node Executive (DSNX). Another name for the change management support offered by the Communications and Systems Management feature. This support processes changes sent by a DSX host system.

distribution list. A list of users to receive a particular piece of mail. This list can be within a group.

DLE. See *data link escape (DLE) character.*

document. One or more lines of text that can be named and stored as a member in a folder.

document folder. A folder that is used to store documents.

DSNX. See *Distributed Systems Node Executive (DSNX).*

DSPT. See *Display Station Pass-Through (DSPT).*

DSU. See *Development Support Utility (DSU).*

DSX. See *Distributed Systems Executive (DSX).*

DTE. See *data terminal equipment (DTE).*

DTF. See *define-the-file (DTF).*

Dual Cluster feature. A feature that provides eight cable connections and allows the attachment of up to eight work stations to a 5251 Model 2 or 12 Display Station. See *Cluster feature.*

dump. (1) To copy the contents of all or part of storage, usually to an output device. (2) Data that has been dumped.

duplex. Pertains to communications in which data can be sent and received at the same time. Same as full duplex. Contrast with *half duplex.*

DW/36. See *DisplayWrite/36 (DW/36).*

EBCDIC. See *extended binary-coded decimal interchange code (EBCDIC).*

EBCDIC character. Any one of the symbols included in the 8-bit EBCDIC set.

edit. (1) To modify the form or format of data; for example, to insert or remove characters such as for dates or decimal points. (2) To check the accuracy of information that has been entered, and to indicate if an error is found. (3) To make changes to a document by adding, changing, or removing text.

EDIT display. The display used to make changes to a member by adding, changing, or removing statements.

EIA. Electronic Industries Association.

eight-line communications adapter/attachment (ELCA). A feature that allows up to eight communication lines to be connected to a 5360 System Unit.

ELCA. See *eight-line communications adapter/attachment (ELCA).*

embedded blanks. Blanks that are surrounded by any other characters.

emulation. Imitation; for example, the imitation of a computer or device.

enable. In interactive communications, to load and start a subsystem. Contrast with *disable.*

end of extent. The end of the area on a disk or diskette reserved for a file.

end of tape. A reflective marking near the end of a tape reel that indicates where the system must stop recording data.

end-of-number character. A character that indicates the end of the telephone number to the autocall unit.

end-of-text (ETX) character. In binary synchronous communications, the transmission control character used to end a logical set of records that began with the start-of-text character.

end-of-transmission (EOT) character. In binary synchronous communications, the transmission control character usually used to end communications.

end-of-transmission-block (ETB) character. In binary synchronous communications, the transmission control character used to end a block of records that began with the start-of-text character.

ENQ. See *enquiry (ENQ) character.*

enquiry (ENQ) character. In binary synchronous communications, the transmission control character usually used to request a response from the remote system or device.

enter. To type in information from a keyboard and press the Enter key in order to send the information to the computer.

enter/update mode. The mode that is used to enter new statements into a source or procedure member, or to change statements that already exist in a source or procedure member.

EOT. See *end-of-transmission (EOT) character.*

error code. See *system reference code.*

error recording analysis procedure (ERAP). An IBM-supplied program that processes and presents recorded errors related to the devices (disk, for example) of the system.

ETB. See *end-of-transmission-block (ETB) character.*

ETX. See *end-of-text (ETX) character.*

evoke. To start a program or procedure so that it can communicate with your program.

exchange file. A file format for exchanging data on diskette or tape between systems or devices that support that medium. See also *basic data exchange.*

exchange station ID. In SDLC, a control field command and/or response for passing station IDs between the primary station and a secondary station.

expiration date. The date after which a diskette file is no longer protected from being automatically erased by the system.

exponent. A number, indicating to which power another number (the base) is to be raised.

exponent (of an E-format number). An integer constant specifying the power of ten by which the base (mantissa) of the decimal floating-point number is to be multiplied.

exponentiation. The operation in which a value is raised to a power.

expression. A representation of a value. For example, variables and constants appearing alone or in combination with operators.

extendable disk file. A file that the system can increase in size whenever more space is needed.

extended binary-coded decimal interchange code (EBCDIC). A set of 256 eight-bit characters.

extended character file. An area on disk that contains the extended ideographic character set.

extended ideographic character set. An ideographic character set, residing in auxiliary storage, that contains 3483 IBM-supplied ideographic characters and up to 4370 user-defined ideographic characters. Contrast with *basic ideographic character set*; see also *ideographic character set.*

extent. A continuous space on disk or diskette that is occupied by, or reserved for, a particular file, library, or folder.

external indicators. Indicators that can be set by another program before a program is run or changed while a program is running. The external indicators are U1 through U8.

feature. A programming or hardware option, usually available at an extra cost. For example, Communications is a feature of the System Support Program Product.

field. One or more characters of related information (such as a name or an amount).

field definition. Information that describes the characteristics of data in a field. A field definition is contained in a data dictionary.

file. A set of related records treated as a unit.

file definition. (1) In RPG, file description and input specifications that describe the records and fields in a file. (2) In IDDU, information that describes the contents and characteristics of a file. A file definition is contained in a data dictionary.

file name. The name used by a program to identify a file. See also *label.*

fill pattern. The shading used inside a bar and pie slice, and below the lines of a surface graph.

Finance subsystem. The SSP-ICF subsystem that allows System/36 to communicate with the 3601 and 4701 Finance Controllers and the 3694 Document Processor.

first-level message. A message that is issued immediately when an error occurs. See also *second-level message.*

fixed-format menu. A menu that is formatted as two 12-item columns. Compare with *free-format menu.*

folder. A named area on disk that contains documents, profiles, mail, or data definitions. Compare with *library.*

folder directory. An area, in a folder, that contains information about each member in the folder; for example, the member name and the location.

folder member. A named collection of statements in a folder. A document is an example of a folder member.

font. An assortment of characters of a given size and style; for example, 10 point Courier.

font ID. A number that identifies the print wheel for certain printers.

format. (1) A defined arrangement of such things as characters, fields, and lines, usually used for displays, printouts, files, or documents. (2) To arrange such things as characters, fields, and lines. (3) In BASIC, a representation of the correct form of a command or statement. (4) In IDDU, a group of related fields, such as a record, in a file.

format definition. Information that describes the contents and characteristics of data within a group of related fields, such as a record in a file. A format definition is contained in a data dictionary.

format member. A load member that contains display formats generated from S and D specifications in a program.

format 1. An area in the disk volume table of contents that contains information about a file; for example, the address and length of the file.

formatted diskette. A diskette on which control information has been written but which may or may not contain any data.

FORTRAN (formula translation). A high-level programming language used primarily for scientific, engineering, and mathematical applications.

free-format menu. A menu for which the programmer defines the format of lines 3 through 20. Contrast with *fixed-format menu.*

full duplex. Same as *duplex.*

full-screen editor. A program that allows you to edit an entire screen of data or text at a time.

function. The action for which a thing exists.

function key. A keyboard key that requests an action but does not display or print a character. The cursor movement and Help keys are examples of function keys. Compare with *command key* and *character key.*

Gaiji. A character in the extended ideographic character set.

GDDM. See *Graphical Data Display Manager (GDDM).*

GDIF. See *graph data input file (GDIF).*

general user. A person, such as an office principal (manager or professional), secretary, or clerk, who is enrolled in and who can sign on to and use Personal Services/36 directly. Contrast with *indirect user.*

generic. Relating to or characteristic of a whole group or class.

global. Pertains to information available to more than one program or subroutine.

graph. Displayed, printed, or plotted output that compares two or more sets of variable data. The types of graphs are bar, line, pie, surface, and text.

graph data input file (GDIF). A file that contains all the data values and labels needed to generate a graph. The file is copied to a graph data member by the BGUDATA procedure.

graph data member. A source member that contains all the actual graph data values.

graph format member. See *format member.*

graph utility. The part of BGU/36 that allows you to design, display, print, and plot graphs, produce a graph object file, and maintain format and data members.

graphic. (1) A picture. (2) See *computer graphics.*

Graphical Data Display Manager (GDDM). A program product that processes both text and graphics for output on a display, printer, or plotter.

half duplex. Pertains to communications in which data can be sent in only one direction at a time. Contrast with *duplex.*

half-index down. The printing of text one-half line down. See also *subscript.*

half-index up. The printing of text one-half line up. See also *superscript.*

handset. The part of a telephone used for talking and listening.

hard copy. A printed copy. Contrast with *online.*

hardware. The equipment, as opposed to the programming, of a system.

HDLC. See *high-level data link control (HDLC).*

header label. A special set of records on a diskette or tape that describes the contents of the diskette or tape.

header record. A record that contains information, such as customer name and customer address, that is common to following detail records. Contrast with *detail record.*

heading. A title of a section, identifying a topic, placed above the section to introduce or categorize the information that follows.

Help key. A function key that, when pressed, displays online information or some part of the system help support.

help support. See *system help support.*

help text. The part of the system help support that offers additional information about displays and messages.

hex. See *hexadecimal.*

hexadecimal. Pertaining to a system of numbers to the base sixteen; hexadecimal digits range from 0 (zero) through 9 (nine) and A (ten) through F (fifteen).

high-level data link control (HDLC). Control of data links by use of a specified series of bits rather than by the control characters of the ISO Standard 7-bit character set for information processing interchange.

history file. A file that contains a log of system actions and operator responses.

horizontal. Parallel to the horizon.

host system. The primary or controlling computer in a communications network. See also *control station.*

I/O. See *input/output (I/O).*

IBM PC. An IBM personal computer; for example, an IBM Personal Computer AT.

ID. Identification.

IDDU. See *interactive data definition utility (IDDU).*

identifier. (1) A sequence of bits or characters that identifies a program, device, or system to another program, device, or system. (2) In COBOL, a data name that is unique or is made unique by the correct combination of qualifiers, subscripts, or indexes. (3) In Personal Services/36, a name that identifies the type of member in a group. The identifier can be a calendar, a user ID, or another group.

ideographic. Pertaining to 2-byte characters consisting of pictograms, symbolic characters, and other types of symbols.

ideographic character set. The combination of the basic and extended ideographic character sets; see also *basic ideographic character set* and *extended ideographic character set*.

ideographic session. A display station operating session during which ideographic data is used for system communication with the operator.

ideographic sort utility. A program that sorts ideographic data.

ideographic SSP. A version of the System Support Program Product that includes formats for help displays in both Katakana 1-byte and Kanji 2-byte ideographic characters. Compare with *Kanji-preferred SSP*.

ideographic support. The hardware and programming elements that allow processing of ideographic data.

idle printer. A printer whose spool writer is started but the spool writer has only spool file entries that are being used by the COPYPRT procedure, are in a held status, or are in the process of being deferred for printing until the spool file program has finished intercepting them.

IF expressions. Expressions within a procedure that are used to test for a condition.

IGC. See *ideographic*.

IMS/IRSS (Information Management System/Intelligent Remote Station Support) subsystem. The SSP-ICF subsystem that provides synchronous communications with IMS/VS. IMS/VS operates on a host system such as a System/370, or a 30XX or 43XX processor.

independent work station. A work station that can operate independently of a host system, but which can also communicate with a host system to use Personal Services/36. An example of an independent work station is a Displaywriter.

independent work station user. A person who uses the Electronic Document Distribution licensed program to communicate with Personal Services/36.

index. (1) A table containing the key value and location of each record in an indexed file. (2) A computer storage position or register, the contents of which identify a particular element in a set of elements.

index key. The field within a record that identifies that record in an indexed file.

indexed file. A file in which the key and the position of each record are recorded in a separate portion of the file called the index. Contrast with *direct file* and *sequential file*.

indicator. An internal switch that communicates a condition between parts of a program or procedure.

indirect user. A person enrolled as a Personal Services/36 user who is authorized to handle mail but has no mail log. Contrast with *general user*.

informational message. A message that provides information to the operator, but does not require a response.

initial program load (IPL). The process of loading the system programs and preparing the system to run jobs.

initialize. To prepare for use. For example, to initialize a diskette.

initiate. To start.

input. Data to be processed.

input stream. The sequence of operation control statements and data given to the system from an input device.

input/output (I/O). Pertaining to either input or output, or both.

inquiry. (1) A request for information in storage. (2) A request that puts a display station into inquiry mode. (3) In data communications, a request for information from another system.

inquiry mode. A mode during which the job currently running from a display station is interrupted so that other work can be done. The operator puts the display station in inquiry mode by pressing the Attn key.

inquiry program. (1) A program that allows an operator to get information from a disk file. (2) A program that runs while the system is in inquiry mode.

installation. The location where a system is installed.

instruction. A statement that specifies an operation to be performed by the computer and the locations in storage of all data involved in that operation.

integer. A positive or negative whole number; that is, an optional sign followed by a number that does not contain a decimal point.

interactive. Pertaining to activity involving requests and replies as, for example, between an operator and a program or between two programs. Contrast with *batch.*

Interactive Communications Feature (SSP-ICF). A feature of the System Support Program Product that allows a program to interactively communicate with another program or system.

interactive data definition utility (IDDU). The part of the System Support Program Product used to define the characteristics of data and the contents of files.

interactive processing. A processing method in which each operator action causes a response from the program or the system. Contrast with *batch processing.*

interrupt. (1) To temporarily stop a process. (2) In data communications, to take an action at a receiving station that causes the sending station to end a transmission.

Intra subsystem. An SSP-ICF subsystem that enables programs to communicate with other programs on the same system without the use of communication lines.

intrinsic. Belonging to the essential nature of a thing.

inverse. A square array that results from a mathematical operation on a square array such that the two arrays can be multiplied together to obtain a square array with a determinant of one.

invite. To ask for input data from either a display station or an SSP-ICF session.

IPL. See *initial program load (IPL).*

ISO. International Standards Organization.

job. (1) A unit of work to be done by a system. (2) One or more related procedures or programs grouped into a procedure.

job file. A disk file that exists until the job that uses it ends.

job queue. A list of jobs waiting to be processed by the system.

job region. The main storage space reserved by the System Support Program Product for use by a job.

job step. A unit of work represented by a single program or a procedure that contains a single program. A job consists of one or more job steps.

job stream. One or more library source members or procedure members saved on diskette or tape.

justify. To adjust text to be even with the top, bottom, left, or right margin.

K-byte. 1024 bytes.

Kanji. (1) The ideographic character set used by the Japanese to represent their native language. (2) A single character in the ideographic character set.

Kanji-preferred SSP. A version of the System Support Program Product that includes formats for help displays in Kanji 2-byte ideographic characters. Compare with *ideographic SSP.*

Katakana. A native Japanese character set that is used primarily to write foreign words phonetically.

key. One or more characters used to identify the record and establish the record's order within an indexed file.

keyboard. A group of numeric keys, alphabetic keys, and function keys used for entering information at a display station and into the system.

Keylock feature. A security feature in which a lock and key can be used to restrict the use of the display station.

keyword. A symbol that identifies a parameter.

label. (1) The name in the disk or diskette volume table of contents or on a tape that identifies a file. See also *file name.* (2) The name that identifies a statement.

LAN. See *local area network (LAN).*

left-adjust. To place or move an entry in a field so that the leftmost character of the field is in the leftmost position. Contrast with *right-adjust.*

library. (1) A named area on disk that can contain programs and related information (not files). A library consists of different sections, called library members. Compare with *folder.* (2) The set of publications for a system.

library control sector. In a library directory, the first sector, which contains a record of the used and available space in the library.

library directory. An area, in a library, that contains information about each member in the library; for example, the member name and the location.

library member. A named collection of records or statements in a library. The types of library members are *load member, procedure member, source member,* and *subroutine member.*

library member subtype. A specific classification of a library member type. For example, a source member can be identified as a COBOL source member or a DFU source member.

licensed application program. A set of licensed programs used to perform a particular data processing task, such as a distribution management application or a construction management application.

licensed program. An IBM-written program that performs functions related to processing user data.

lines per inch (LPI). The number of characters printed within an inch vertically down the page.

link. In data communications, the connection between two systems.

link level. A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the full-duplex link connecting the subscriber's machine to the network node. LAP and LAPB are the link access protocols recommended by the CCITT.

link protocol. See *link level.*

link-editing. To combine, by the overlay linkage editor, a number of load members and/or subroutine members into one program.

linkage editor. See *overlay linkage editor.*

literal. A symbol or a quantity in a source program that is itself data, rather than a reference to data.

load. (1) To move data or programs into storage. (2) To place a diskette into a diskette drive or a diskette magazine into a diskette magazine drive. (3) To insert paper into a printer. (4) To mount a tape or insert a tape cartridge into a tape drive.

load member. A library member that contains information in machine language, a form that the system can use directly. Contrast with *source member.*

load module. A program in a form that can be loaded into main storage and run. The load module is the output of the overlay linkage editor.

local. Pertaining to a device, file, or system that is accessed directly from your system, without the use of a communications line. Contrast with *remote.*

local area network (LAN). The physical connection among devices located on the same premises for information transfer.

local data area. A 512-byte area on disk that can be used to pass information between jobs and job steps during a session. A separate local data area exists for each command display station.

location name. In interactive communications, the identifying name associated with a particular system or device.

location password. A string of hexadecimal characters that allows the system to verify the identity of a remote location.

location profile. A profile in the user identification file that contains information about a remote system that is allowed to access resources on your system.

log. (1) To record; for example, to log all messages on the system printer. (2) See *mail log.*

logical unit (LU). The part of a system or device in an SNA network that allows a user or program to use the communications network.

loop. A sequence of instructions that is performed repeatedly until an ending condition is reached.

LPI. See *lines per inch (LPI).*

LU. See *logical unit (LU).*

macro. See *macroinstruction.*

macroinstruction. A single instruction that represents a set of instructions.

magazine. A container that holds up to 10 diskettes.

magnetic ink. An ink that contains particles of a magnetic substance whose presence can be detected by magnetic sensors.

magnetic ink character recognition. The identification of characters through the use of magnetic ink.

magnetic stripe reader. A device, attached to a display station, that reads data from a magnetic stripe on a badge before allowing an operator to sign on.

magnetic tape. See *tape.*

magnetic tape unit. A device for reading or writing data from or on magnetic tape.

mail. Any correspondence (online or hard copy) that is sent between users.

mail folder. A folder used to store documents sent and received as mail.

mail log. A record of all the mail sent or received by a user.

mail queue. See *communications queue.*

main storage. The part of the processing unit where programs are run. Contrast with *control storage.*

main storage processor. Hardware that performs the machine language instructions in main storage.

mandatory entry field. A field in which an operator must enter at least one character.

mandatory fill field. A field for which an operator must enter nothing or must fill in completely.

manual answer. In data communications, a line type requiring operator actions to receive a call over a switched line. Contrast with *autoanswer.*

manual call. In data communications, a line type requiring operator actions to place a call over a switched line. Contrast with *autocall.*

margin. The space between the text area and the top, bottom, or side edges of the display or paper.

mask. A pattern of characters that controls the keeping, deleting, or testing of portions of another pattern of characters.

master configuration record. Information, stored on disk, that describes system devices, programming, and characteristics.

master file. A collection of permanent information, such as a customer address file.

master security officer. A person who is designated to control all of the security tasks that are provided with the System Support Program Product. A master security officer can, for example, deactivate password, badge, or resource security, or add, change, or remove security information about any system operator. Contrast with *security officer.*

megabyte. One million bytes.

member. See *library member.*

menu. A displayed list of items from which an operator can make a selection.

menu security. A System Support Program Product option that restricts an operator to selecting items from a particular menu.

merge. To combine two or more ordered files into one similarly ordered file.

message. (1) Information sent to one or more users or display stations from a program or another user. A message can be either displayed or printed. (2) An indication of the condition of the system sent by the system. (3) For IMS/IRSS, a unit of data sent over the communications line.

message identification. A field in the display or printout of a message that directs the user to the description of the message in a message guide or a reference manual. This field consists of up to four alphabetic characters, followed by a dash, followed by the message identification code.

message identification code (MIC). A four-digit number that identifies a record in a message member. This number can be part of the message identification.

message member. A library member that defines the text of each message and its associated message identification code.

MIC. See *message identification code (MIC).*

migrate. To convert files created by one program or utility to files that can be used by another program or utility.

MLCA. See *multiline communications adapter/attachment (MLCA).*

mm. Millimeter.

mnemonic. An identifier or symbol, using characters intended to assist memory, that is associated with a command, instruction, or statement.

mode. A method of operation. For an example, see *enter/update mode.*

modem. See *modulator-demodulator (modem).*

modulation. Changing the frequency or size of one signal by using the frequency or size of another signal.

modulator-demodulator (modem). A device that converts data from the computer to a signal that can be transmitted on a communications line, and converts the signal received to data for the computer.

module. (1) One part of a program, which usually performs a specific task (such as disk input/output). (2) See *load module.* (3) See *object module.*

modulus 10/modulus 11 checking. Formulas used to calculate the check digit for a self-check field.

monitor. Programming or hardware that observes, supervises, controls, or verifies the operation of a system.

MRT procedure. See *multiple requester terminal (MRT) procedure.*

MRT program. See *multiple requester terminal (MRT) program.*

MSRJE. See *Multiple Session Remote Job Entry (MSRJE).*

multiline communications adapter/attachment (MLCA). A feature that allows up to four communication lines to be connected to System/36.

multinational character set. An option that makes an expanded set of 188 characters available to countries with supported language groups.

multiple. More than one.

multiple requester terminal (MRT) procedure. A procedure that calls a multiple requester terminal program.

multiple requester terminal (MRT) program. A program that can process requests from more than one display station or SSP-ICF session at the same time using a single copy of the program. Contrast with *single requester terminal (SRT) program.*

Multiple Session Remote Job Entry (MSRJE). A feature of the System Support Program Product that allows one or more remote job entry sessions to operate on a host system (such as a System/370, or a 30XX or 43XX processor) at the same time.

multipoint. In data communications, pertains to a network that allows two or more stations to communicate with a single system on one line.

multiprogramming. The processing of two or more programs at the same time.

multivolume file. A diskette file that occupies more than one diskette.

NAK. See *negative acknowledgment character (NAK).*

negative acknowledgment character (NAK). In binary synchronous communications, a transmission control character sent as a negative response to data received.

NEP. See *never-ending program (NEP).*

nest. To incorporate a structure or structures of some kind into a structure of the same kind. For example, to nest one loop (the nested loop) within another loop (the nesting loop); to nest one subroutine (the nested subroutine) within another subroutine (the nesting subroutine).

nested procedure. A procedure that is called by another procedure. See also *procedure level.*

network. A collection of data processing products connected by communications lines for information exchange between stations.

network resource directory (NRD). An area on disk that lists the files on remote systems that can be accessed using Distributed Data Management (DDM).

never-ending program (NEP). A long-running program that does not share system resources, except for shared files and the spool file.

node. (1) An addressable location in a communications network that provides host processing services. (2) A point where packets are received, stored, and forwarded to another node (or DTE) according to a routing method the network has defined.

node identification. A string of characters that identifies a node to the system.

non-return-to-zero inverted (NRZI). On System/36, a method of data transmission where the signal is changed to transmit a 0 bit. For the 1 bit the signal stays the same. This ensures that the signal does not stay the same for an extended period of time.

noncontiguous. Not being in actual contact.

nondisplay. A field attribute that prevents the displaying of data.

nonlabeled tape. A tape that has no labels. Tape marks are used to indicate the end of the volume and the end of each data file.

nonrequesting terminal program. A program that is not associated with a requesting display station.

nonstandard labeled tape. A tape that has labels but does not follow the IBM standard labeling conventions.

nonswappable storage. The storage containing programs or data that must remain in storage.

nonswitched line. A connection between computers or devices that does not have to be established by dialing. Contrast with *switched line.*

NRZI. See *non-return-to-zero inverted (NRZI).*

nucleus. That portion of main storage that is used by the System Support Program Product.

null. See *null character.*

null character. The character hex 00, used to represent the absence of a displayed or printed character.

null character string. Two consecutive single quotation marks that specify a character constant of no characters.

null record. In binary synchronous communications, a record that contains no data; only the data link control characters STX ETX.

number list. A list of telephone numbers to be called using a communications program and the X.21 feature.

numeric. Pertaining to any of the digits 0 through 9.

object module. A set of instructions in machine language. The object module is produced by a compiler from a subroutine or source program and can be input to the overlay linkage editor.

OCL. See *operation control language (OCL).*

office products. A group of IBM-supplied programs that work together to help an office operate more efficiently. The office products are DisplayWrite/36 (DW/36), Personal Services/36, and Query/36. The interactive data definition utility (IDDU) can be used to define files used by DW/36 and Query/36.

office profile. A profile that contains information about a user.

OFFICE/36. The group of office products: DisplayWrite/36 (DW/36), Personal Services/36, and Query/36.

offline. Neither controlled directly by, nor communicating with, the computer, or both. Contrast with *online.*

online. Being controlled directly by, or directly communicating with, the computer, or both. Contrast with *offline.*

online information. Information, read on the display screen, that explains displays, messages, and programs. For some programs, the online information is similar to a printed manual and may contain a table of contents, guide information, practice exercises, help text, a glossary, and an index.

open. To prepare a file for processing.

operand. A quantity of data that is operated on, or the address in a computer instruction of data to be operated on.

operating system. A collection of system programs that controls the overall operation of a computer system.

operation. A defined action, such as adding or comparing, performed on one or more data items.

operation control language (OCL). A language used to identify a job and its processing requirements to the System Support Program Product.

operator. (1) A person who operates a device. (2) A symbol that represents an operation to be done.

option. An item (usually numbered) in a list that a user selects to perform a task.

optional network facilities. Facilities a packet switching data network user may request when establishing a virtual circuit. See also *reverse charging, closed user group,* and *throughput class negotiation.*

optional SSP. Displays and programs included in the System Support Program Product that can optionally be loaded during system configuration.

output. The result of processing data.

overflow indicator. An indicator that signifies that the last line on a page has been printed or skipped.

overflow line. The line specified as the last line to be printed on a page.

overlay. (1) To write over (and therefore destroy) an existing file. (2) A program segment that is loaded into main storage and replaces all or part of a previously loaded program segment.

overlay linkage editor. The part of the System Support Program Product that combines object programs to produce code that can be run and allows the user to determine overlays for programs.

overlay region. A continuous area of main storage in which segments can be loaded independently of other regions.

override. (1) A parameter or value that replaces a previous parameter or value. (2) To replace a parameter or value.

override user ID. A user identification that is used to sign on to the system if the user identification file is destroyed.

overstrike. To enter a character in a space currently occupied by another character.

overview. A brief general survey; a summary.

packed decimal format. A format in which each byte (except the rightmost byte) within a field represents two numeric digits. The rightmost byte contains one digit and the sign. For example, the decimal value 123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format.*

packed key. An index key in packed decimal format.

packet. A data transmission information unit. It has a header on the front that indicates the destination of the packet. Commonly used data field lengths in packets are 128 or 256 bytes.

packet level. A part of Recommendation X.25 that defines the protocol for establishing logical connections between two DTEs and for transferring data on these connections.

packet procedures. See *packet level.*

packet switching. The act of transferring and routing packets from source to destination based on information contained in their headers.

packet switching data network (PSDN). A communications network that uses packet switching as a means of transmitting data.

pad. To fill unused positions in a field with dummy data, usually zeros or blanks.

page. A 2048-byte segment of main storage.

palette. See *color palette.*

parameter. A value supplied to a procedure or program that either is used as input or controls the actions of the procedure or program.

parent. Pertaining to a secured resource, either a file or library, whose user list is shared with one or more other files or libraries. Contrast with *child.*

password. A string of characters that, when entered along with a user ID, allows an operator to sign on to the system.

password security. A System Support Program Product option that helps prevent the unauthorized use of a display station, by checking the password entered by each operator at sign-on.

patch. To change directly the contents of a file or library member.

path information unit (PIU). A transmission unit in an SNA network. A PIU has a transmission header and can contain a basic information unit (BIU) that holds data.

PC. A personal computer.

PC Support/36. A group of programs that can be used to transfer data from a System/36 to an IBM personal computer, to use disk storage on System/36 as IBM personal computer disk storage, and to use printers attached to System/36 as a IBM personal computer printer.

PCE. See *procedure control expression (PCE).*

Peer subsystem. The SSP-ICF subsystem that allows System/36 to communicate with another System/36 or System/34 using SNA/SDLC.

pending. Waiting, as in an operation is pending.

permanent virtual circuit (PVC). A virtual circuit that has a logical channel permanently assigned to it at each DTE. The usual call establishment protocol is therefore not required.

personal computer utility (PCU). The part of PC Support/36 that creates, deletes, and copies files to and from virtual disks on the System/36.

Personal Services/36. A program product that can be used to send and receive mail, schedule appointments on calendars, maintain directories of names and addresses, and work with groups of users or calendars.

phone list. A list of telephone numbers to be called using a communications program and the autocall or X.25 feature.

physical connection. See *physical level (X.25)*

physical file. A file that contains data records.

physical level (X.25). A standard that defines the electrical, physical, functional, and procedural methods used to control the physical link running between the DTE and the DCE.

physical record. (1) A group of records that is recorded or processed as a unit. Same as *block.* (2) A unit of data that is moved into or out of the computer.

pitch. A unit of width of typewriter type, based on the number of characters that can be set in a linear inch. For example, 10-pitch type has 10 characters per inch.

plot. To draw or diagram. To connect point-by-point coordinate values.

plotter. A device for drawing a graph or chart.

point-to-point line. A communications line that connects a single remote station to a computer.

poll. To execute a polling sequence.

polling. A method for determining whether each of the stations on a communications line has data to send.

port. A part of the system unit or remote controller to which cables for display stations and printers are attached.

position. The location of a character in a series, as in a record, a displayed message, or a computer printout.

positional parameter. A parameter that must appear in a specified location, relative to other positional parameters.

post. (1) To add information in a record to keep that record current. (2) To note the occurrence of an event.

primary index. The index that is built when a file is created. Contrast with *alternative index*.

print band. An interchangeable metal band that contains the print characters used by some printers.

print belt. See *print band*.

print entries. See *spool file entries*.

print file. In MSRJE, a file created by the host system that is printed on your system.

print image. A character set that corresponds to the characters on a print band.

print intercept routine. The spooling routine that causes printer output to be placed in a spool file rather than being printed.

print wheel. An interchangeable print element used in certain printers.

printer load balancing. A System/36 function that balances spooled output among a group of printers.

printout. Information from the computer that is produced by a printer.

priority. The relative ranking of items. For example, a job with high priority will be run before one with regular or low priority.

problem determination. The process of identifying why the system is not working. Often this process identifies programs, equipment, data communications facilities, or user errors as the source of the problem.

problem management. The part of the Communications and Systems Management feature that allows System/36 to generate and send alerts to a host or peer system using APPC or APPN.

procedure. A set of related operation control language statements (and, possibly, utility control statements and procedure control expressions) that cause a specific program or set of programs to be performed.

procedure command. A command that runs a procedure.

procedure control expression (PCE). A set of statements and expressions that control how a procedure runs.

procedure level. The relative position of a procedure within nested procedures. For example, if procedure A calls procedure B, and procedure B in turn calls procedure C, then procedure C is a third-level procedure.

procedure member. A library member that contains the statements (such as operation control language statements) necessary to perform a program or set of programs.

processing. The performance of operations and calculations on data.

processing unit. The part of the system unit that performs instructions and contains main storage.

profile. Data that describes the significant features of a user, program, device, or remote location.

program. (1) A sequence of instructions for a computer. See *source program* and *load module*. (2) To write a sequence of instructions for a computer. Same as *code*.

program date. The date associated with a program (job step). See also *creation date*, *session date*, and *system date*.

program generation. The compilation of a WSU program.

program product. A licensed program for which a fee is charged.

program status register. A register that contains conditions that can be tested by branch or jump instructions.

program temporary fix (PTF). A temporary solution to or bypass of a defect in a current release of a licensed program.

Programming Request for Price Quotation (PRPQ). A program created especially for a particular group of customers or an application. Documentation for the program is provided only to those customers who order the PRPQ.

prompt. A displayed request for information or operator action.

protected field. A displayed field in which operators cannot enter data.

protocol. A set of rules governing the communication and transfer of data between two or more devices in a communications system.

PRPQ. See *Programming Request for Price Quotation (PRPQ).*

PSDN. See *packet switching data network (PSDN).*

PTF. See *program temporary fix (PTF).*

PTF backup library. A library that contains a copy of the load modules replaced by a PTF. The library is created by the PTF procedure with the APPLY parameter specified. When the REMOVE parameter is specified, the PTFs are removed and the original load modules are replaced.

PTF library. A library that contains the PTFs to be applied by the PTF procedure. The library is created by the PTF procedure with the COPY parameter specified.

public data network. A communications common carrier network that provides data communications services over switched or nonswitched lines.

purge. To remove an entry from a queue.

query. A request for information from a file based on specific conditions; for example, a request for a list of all customers in a customer master file whose balance is greater than $1000.

Query/36. A program product that produces files and reports based on those files. The files must be linked to file definitions created with IDDU.

queue. A line or list formed by items waiting to be processed.

random processing. The processing of records in an order other than the order that they exist in a file. See also *consecutive processing* and *sequential processing.*

receive time-out. In data communications, the result of no data being received in a given period of time.

Recommendation X.25. A document, CCITT Recommendation X.25, that outlines standards for the connection of processing equipment to a packet switching data network.

record. A collection of fields that is treated as a unit.

record file. A file on disk in which the data is read and written in records. Contrast with *stream file.*

record separator. In binary synchronous communications, a character used to indicate the end of one record and the beginning of another.

record type. The classification of records in a file.

recovery procedure. (1) An action performed by the operator when an error message appears on the display screen. Usually, this action permits the program to continue or permits the operator to run the next job. (2) The method of returning the system to the point where a major system error occurred and running the recent critical jobs again.

region size. The amount of main storage available for a program to run. See also *job region* and *step region.*

register. A storage area, in a computer, usually intended for some special reason, capable of storing a specified amount of data such as a bit or an address.

relative file. Same as *direct file.*

relative record number. A number that specifies the location of a record in relation to the beginning of the file.

release. (1) To allow a spool job that is being held to run. (2) A distribution of new function for an existing program product.

release update. The process of updating programming support by installing a new release of the System Support Program Product and program products.

reminder. A calendar item that includes a date, but no start time or end time.

remote. Pertaining to a device, file, or system that is accessed by your system through a communications line. Contrast with *local.*

remote controller. A device, attached to a communications line, that controls the operation of one or more remote display stations and printers.

remote job entry (RJE). Sending job instructions and possibly data to a remote system requesting it to run a job.

Remote Operation/Support Facility (ROSF). An implementation that allows an operator at a remote support group to use a remote display station (and an optional remote printer) to provide operational and technical assistance.

remotely started session. A session started by an incoming procedure start request from the remote system. Contrast with *acquired session.*

reorganize. To move folder members together at the front of the folder to reduce as much as possible the number of folder extents.

requester. A display station or interactive communications session that requests a program to be run.

reset. To return a device or circuit to a clear state.

resident file. A file that exists on disk until it is specifically deleted or changed to a scratch file.

resource. Any part of the system required by a job or task, including main storage, input and output devices, the processing unit, and files, libraries, and folders.

resource security. A System Support Program Product option that restricts the use of information in files, libraries, folders, and folder members to specified users.

resource security file. A security file that contains information that restricts access to files, libraries, and folders.

restore. Return to an original value or image. For example, to restore a library from diskette.

return code. In data communications, a value generated by the system or subsystem that is returned to a program to indicate the results of an operation issued by that program.

reverse charging. A packet switching data network optional facility. It enables the DTE to request that the cost of a communications session it initiates be charged to the DTE that is called. See also *optional network facilities.*

reverse-interrupt character (RVI). In binary synchronous communications, a request by the receiving station to the sending station to stop sending and begin receiving a message.

revise. To change. See also *edit.*

right-adjust. To place or move an entry in a field so that the rightmost character of the field is in the rightmost position. Contrast with *left-adjust.*

RJE. See *remote job entry (RJE).*

ROSF. See *Remote Operation/Support Facility (ROSF).*

routine. A set of statements in a program that causes the system to perform an operation or a series of related operations.

RPG. A programming language specifically designed for writing application programs that meet common business data processing requirements.

run. To cause a program, utility, or other machine function to be performed.

RVI. See *reverse-interrupt character (RVI).*

RWS. Remote work station.

scratch file. A file, usually used as a work file, that exists until the program that uses it ends.

screen design aid (SDA). The part of the Utilities Program Product that helps the user design, create, and maintain displays and menus. Additionally, SDA can generate specifications for RPG and WSU work station programs.

scroll. See *roll.*

SDA. See *screen design aid (SDA).*

SDLC. See *synchronous data link control (SDLC).*

second-level message. A message that supplies additional information about an error condition when the Help key is pressed for a first-level message. See also *first-level message.*

sector. (1) An area on a disk track or a diskette track reserved to record information. (2) The smallest amount of information that can be written to or read from a disk or diskette during a single read or write operation.

security. The protection of data, system operations, and devices from accidental or intentional ruin, damage, or exposure. See also *system security.*

security officer. A person who is designated to control many of the system security tasks that are provided with the System Support Program Product. A security officer can, for example, add, change, or remove security information about system console operators, subconsole operators, and display station operators. A security officer cannot, however, deactivate password, badge, or resource security. Contrast with *master security officer.*

segment. A part of a program that can be run without the entire program being in main storage.

seion. A Japanese syllable.

seion kana control field. A control field that the ideographic sort program sorts into seion kana sequence and that recognizes reading field sounds.

separator character. In data communications, the character that is used with some autocall units to separate the digits to be dialed.

separator page. A printed page used to show the end of output for one job and the start of output for another job.

sequential file. A file in which records occur in the order in which they were entered. Contrast with *direct file* and *indexed file.*

sequential processing. The processing of records in the order in which they exist in a file. Same as *consecutive processing.* See also *random processing.*

service aids. The group of procedures and programs that is used to determine and correct system problems.

service log. A file that contains information about problems that have occurred with the system. The file can be added to using the SERVLOG procedure.

session. (1) The logical connection by which a System/36 program or device can communicate with a program or device at a remote location. (2) The length of time that starts when an operator signs on the system and ends when the operator signs off the system.

session date. The date associated with a session. See also *creation date, program date,* and *system date.*

session group. In APPC, a number of communications sessions to be managed as a unit.

session library. The library specified, or assigned as a default, when signing on or while running a program.

SEU. See *source entry utility (SEU).*

severity code. A code that indicates how serious a compiling error or an operating error is.

severity level. See *automatic response severity level.*

shared folder facility. A function of PC Support/36 that allows multiple System/36 and personal computer users to have concurrent access to a folder from a personal computer.

shift. To adjust line contents by moving existing information to the left or right of its original position.

shift-in control character. A character (hex 0F) that indicates the end of a string of ideographic characters. Contrast with *shift-out control character*

shift-out control character. A character (hex 0E) that indicates the start of a string of ideographic characters. Contrast with *shift-in control character*

sign off. To end a session at a display station.

sign on. (Verb) To begin a session at a display station.

sign-on. (Noun) The action an operator uses at a display station in order to begin working at the display station.

single line communications adapter/attachment (SLCA). In data communications, a feature that allows a single communications line to be connected to System/36.

single requester terminal (SRT) program. A program that can process requests from only one display station or SSP-ICF session from each copy of the program. Contrast with *multiple requester terminal (MRT) program.*

SLCA. See *single line communications adapter/attachment (SLCA).*

SMF. See *system measurement facility (SMF).*

SNA. See *systems network architecture (SNA).*

SNA Upline Facility (SNUF). The SSP-ICF subsystem that allows System/36 to communicate with CICS/VS and IMS/VS application programs on a host system. Also, using this subsystem, DHCF communicates with HCF and DSNX communicates with DSX.

SNBU. See *switched network backup (SNBU).*

SNUF. See *SNA Upline Facility (SNUF).*

software. Programs, languages, and/or routines that control the operations of a computer in solving a given problem.

sort utility. The part of the System Support Program Product used to arrange records (or their relative record numbers) in a sequence determined by data contained in one or more fields within the records.

source entry utility (SEU). The part of the Utilities Program Product used by the operator to enter and update source and procedure members.

source member. A library member that contains information in the form in which it was entered, such as RPG specifications. Contrast with *load member.*

source program. A set of instructions that are written in a programming language and that must be translated to machine language before the program can be run.

source statement. A statement written in a programming language.

special character. A character other than an alphabetic or numeric character. For example; *, +, and % are special characters.

specification sheets. Forms on which a program is coded and described.

split key. A key, for an indexed file, defined from more than one field within each record.

spool file. A disk file that contains output that has been saved for later printing.

spool file entries. Output in the spool file waiting to be printed.

spool intercept buffer. An area of main storage containing printer data that is being written in the spool file.

spool writer. The part of the System Support Program Product that prints output that has been saved in the spool file.

spooling. The part of the System Support Program Product that saves output on disk for later printing.

SRT program. See *single requester terminal (SRT) program.*

SSP. See *System Support Program Product (SSP).*

SSP-ICF. See *Interactive Communications Feature (SSP-ICF).*

standard label tape. A tape that follows the IBM standard labeling conventions.

standby display. A display that allows an operator to enter data only. When a standby display appears, the display station can be acquired by a program. Contrast with *command display.*

statement. An instruction in a program or procedure.

station. A computer or device that can send or receive data.

status. A condition. For example, the status of a printer, a job, or a communications line.

step region. The main storage space reserved by the System Support Program Product for use by a program.

storage index. A table in main storage that contains the address of the lowest key on each track in the file index.

storage map. A compiler printout that shows the names and storage locations of variables and statement numbers in the object program.

storage usage map. An overlay linkage editor printout that shows the names and storage locations of routines that make up the load member.

stream file. A file on disk in which data is read and written in consecutive fields. Contrast with *record file.*

subconsole. A display station that controls a printer or printers.

subconsole display. A display that can be requested only from a command display that appears on a subconsole. From a subconsole display an operator can display and send messages, and enter all control commands except those that can be entered only at the system console. See also *console display.*

subdirectory. A user-defined area of a folder that contains the names, descriptions, member types, and security information for other directories (subdirectories) and folder members. Subdirectories are part of shared folder facility. See also *shared folder facility.*

subroutine. A group of instructions that can be called by another program or subroutine.

subroutine member. A library member that contains information that must be combined with one or more members before being run by the system.

subscript. The function that allows text and numbers to be printed one-half line below the normal printing line. For example, the number 2 in the chemical formula for water, H_2O, is a subscript.

subsystem. The part of communications that handles the requirements of the remote system, isolating most system-dependent considerations from the application program.

subtype. See *library member subtype.*

superscript. The function that allows text and numbers to be printed one-half line above the normal printing line. For example, a footnote is called out in text with a superscript number.

swapping. The process of temporarily removing an active job from main storage, saving it on disk, and processing another job in the area of main storage formerly occupied by the first job.

switched line. In data communications, a connection between computers or devices that is established by dialing. Contrast with *nonswitched line.*

switched network backup (SNBU). In data communications, a technique that provides a switched line connection when a nonswitched line fails.

switched virtual circuit. A virtual circuit that is requested from the network through a virtual call. It is released when the virtual circuit is cleared.

SYN. See *synchronization (SYN) character.*

synchronization (SYN) character. In binary synchronous communications, the transmission control character that provides a signal to the receiving station for timing.

synchronous. Occurring in a regular or predictable sequence.

synchronous data link control (SDLC). A form of communications line control that uses commands to control the transfer of data over a communications line. Compare with *binary synchronous communications (BSC).*

synchronous transmission. In data communications, a method of transmission in which the sending and receiving of characters is controlled by timing signals. Contrast with *asynchronous transmission.*

syntax. The rules for the construction of a command or statement.

system. The computer and its associated devices and programs.

system configuration. A process that specifies the machines, devices, and programs that form a particular data processing system.

system console. A display station from which an operator can keep track of and control system operation.

system date. The date assigned by the system operator during the initial program load procedure. See also *creation date*, *program date*, and *session date.*

system dump. A dump of all active programs (and their associated data) recorded after an error stops the system. Contrast with *task dump.*

system help support. The part of the System Support Program Product that uses menus, prompts, and descriptive text to aid an operator.

system library. The library, provided with the system, that contains the System Support Program Product and is named #LIBRARY.

system list device. The device that receives output for most System Support Program Product utility programs and service aids.

system log device. The device or devices designated by the LOG OCL statement to record messages and OCL statements.

system measurement facility (SMF). System Support Program Product routines that, in conjunction with control storage routines, observe system and device activity, observe SSP work area usage, and record this data in a disk file.

system printer. The printer that is used for any printed output that is not specifically directed to another printer.

system program. An IBM-supplied program that is installed on the system. The System Support Program Product (SSP) is an example.

system reference code. A four-character code that contains information for a service representative. This code either is provided as part of a message or is displayed on the control panel.

system security. A system function that restricts the use of files, libraries, folders, folder members, and display stations to certain users.

system service display station. A display station that can use all the procedures, programs, and commands needed to service the system.

System Support Program Product (SSP). A group of licensed programs that manage the running of other programs and the operation of associated devices, such as the display station and printer. The SSP also contains utility programs that perform common tasks, such as copying information from diskette to disk.

system unit. The part of the system that contains the processing unit, the control panel, the disk drive and the disk, and either a diskette drive or a diskette magazine drive.

systems network architecture (SNA). A set of rules for controlling the transfer of information in a data communications network.

table. A collection of data in which each item is uniquely identified by a label, by its position relative to the other items, or by some other means.

tape. A thin, flexible magnetic strip on which data can be stored. It can be used to store information copied from the disk.

tape cartridge. A case containing a reel of magnetic tape arranged for insertion into a tape drive.

tape drive. A mechanism used to read and write information on magnetic tapes.

tape mark. A mark on the tape that indicates the beginning or end of a file or tape.

tape reel. A round device on which magnetic tape is wound.

tape volume. A single reel of magnetic tape.

task. A unit of work (such as a user program) for the main storage processor.

task dump. A dump of a program that failed (and its associated data). Contrast with *system dump*.

task work area. An area on disk containing control information and work areas related to a specific task.

terminal. In data communications, a device, usually equipped with a keyboard and a display device, capable of sending and receiving information over a communications line.

terminator. The part of the System Support Program Product that performs the action necessary to end a job or program.

text. The displayed or printed information of a document.

text folder. See *document folder*.

throughput class negotiation. A packet switching data network optional facility. Allows a DTE to negotiate the speed at which its packets travel through the packet switching data network.

token-ring network. The local area network (LAN) designed to run on the IBM Calling System.

trace. To record data that provides a history of events that occur in the system.

trace file. A file that contains a record of the events that occur in the system.

track. A circular path on the surface of a disk or diskette on which information is magnetically recorded and from which recorded information is read.

trailer. Control information that WSU adds to the end of each record in a transaction file.

transaction. (1) An item of business. The handling of customer orders and customer billing are examples of transactions. (2) In interactive communications, the communication between the application program and a specific item (usually another application program) at the remote system.

transaction file. A file containing data, such as customer orders, that is usually used only with a master file.

transmission control characters. In data communications, special characters that are included in a message to control communication over a data link. For example, the sending station and the receiving station use transmission control characters to exchange information; the receiving station uses transmission control characters to indicate errors in data it receives.

transparent data. Data that can contain any hexadecimal value.

tributary station. In data communications, a secondary device on a multipoint line.

truncate. To shorten a field or statement to a specified length.

unlink. To remove an association between a file on disk and a file definition in a data dictionary. Contrast with *link*.

update authority. The right to add, change, or remove items in a file, library, or folder.

uppercase. Pertaining to capital letters.

UPSI switch. See *user program status indicator (UPSI) switch*.

user area. The parts of main storage and disk that are available to the user.

user ID. See *user identification (user ID)*.

user identification (user ID). A string of characters that identifies a user to the system.

user identification file (user ID file). A file containing information about which operators can use certain system functions, which menu is displayed when an operator signs on to the system, and which library is assigned to an operator when the operator signs on to the system.

user identification record (user ID record). A record in the directory that gives a user's name, address, and telephone number.

user list. A list, containing the user identification and access levels, of all operators who are allowed to use a specified file or library.

user profile. A profile in the user identification file that contains information about someone who is allowed to sign on to the system.

user program status indicator (UPSI) switch. One of a set of eight switches that can be set by and passed between application programs and procedures.

Utilities Program Product. A program product that contains the data file utility (DFU), the source entry utility (SEU), the work station utility (WSU), and the screen design aid (SDA).

utility control statement. A statement that gives a utility program information about the way the program is to perform or the output it is to produce.

utility program. (1) A program provided to perform a task that is required by many of the programs using the system; for example, a program that copies information from diskette to disk. (2) A program of the System Support Program Product that performs a common task.

valid. (1) Allowed. (2) True, in conforming to an appropriate standard or authority.

variable. A name used to represent a data item whose value can change while the program is running. Contrast with *constant.*

verify. To confirm the correctness of something.

vertical. Perpendicular to a base line.

view. To show a graph or chart on a display.

virtual call facility. A user facility in which a call setup procedure and a call clearing procedure will determine a period of communication between two data terminal equipments (DTEs) in which user's data will be transferred in the network in the packet mode of operation. All the user's data is delivered from the network in the same order in which it is received by the network.

virtual circuit. A logical connection established between two DTEs. It can be permanent, that is, defined when you subscribe to your network port, or it can be dynamically established when creating a switched virtual circuit.

virtual disk. An area of main storage created by PC Support/36 to contain data from an IBM personal computer.

virtual diskette. An area of main storage created by the File Support Utility to contain data from an IBM personal computer.

voice-grade telephone line. A telephone line that is normally used for transmission of voice communications. The line requires a modem for data communications.

volume label. An area on a standard label tape used to identify the tape volume and its owner. This area is the first 80 bytes and contains VOL1 in the first four positions.

volume table of contents (VTOC). An area on a disk or diskette that describes the location, size, and other characteristics of each file, library, and folder on the disk or diskette.

VTOC. See *volume table of contents (VTOC).*

WACK. See *wait-before-transmitting-acknowledgment character (WACK).*

wait-before-transmitting-acknowledgment character (WACK). In BSC, the transmission control character indicating that the station is temporarily not ready to receive data.

weak external reference. An external reference that does not have to be resolved during linkage editing. If it is not resolved, it appears as though its value were resolved to zero.

window. In data communications, the number of data packets a DTE or DCE can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing or flow control of packets.

work area. The area in main storage that a BASIC program occupies during compilation. The work area consists of the file specification, code, free, and symbol areas, and compiler tables.

work file. A file that is used for temporary storage of data being processed.

work station. A device that lets people transmit information to or receive information from a computer; for example, a display station or printer.

work station address. (1) A number used in a configuration member to identify a work station attached to a port. (2) The address to which the switches on a work station are set, or the internal default address.

work station data management. The part of the System Support Program Product that enables a program to present data on a display screen by providing a string of data fields and a format name.

work station ID. A two-character identifier assigned to each display station and printer on your system.

work station utility (WSU). The part of the Utilities Program Product that helps you to write programs for data entry, editing, and inquiry.

World Trade. (1) Pertains to the distinction between the US and the rest of the world. (2) Pertains to the combination of:

- IBM World Trade Americas/Far East Corporation
- IBM World Trade Europe/Middle East/Africa Corporation

WSU. See *work station utility (WSU).*

X.21. In data communications, a specification of the CCITT that defines the connection of data terminal equipment to an X.21 (public data) network.

X.21 feature. The feature that allows System/36 to be connected to an X.21 network.

X.21 short hold mode. An option specified during system configuration that allows a circuit switched line to be disconnected when the line is not active.

X.25. In data communications, a specification of the CCITT that defines the interface to an X.25 (packet switching) network.

X.25 feature. The feature that allows System/36 to be connected to an X.25 network.

X.75. A standard that defines ways of interconnecting two X.25 networks.

zoned decimal format. A format for representing numbers in which the digit is contained in bits 4 through 7 and the sign is contained in bits 0 through 3 of the rightmost byte; bits 0 through 3 of all other bytes contain 1s (hex F). For example, in zoned decimal format, the decimal value of +123 is represented as 1111 0001 1111 0010 1111 0011. Contrast with *packed decimal format.*

zoned decimal item. A numeric data item that is represented internally in zoned decimal format.

zoned field. A field that contains data in the zoned decimal format.

1024-byte format. A format for diskette 2D diskettes with 1024 bytes per sector and 8 sectors per track.

1255 Magnetic Character Reader. A device that reads documents printed with magnetic ink characters.

128-byte format. A format for diskette 1 diskettes with 128 bytes per sector and 26 sectors per track.

256-byte format. A format for diskette 2D diskettes with 256 bytes per sector and 26 sectors per track.

3270 BSC Support subsystem. The subsystem that provides program-to-program communications with IMS/VS, CICS/VS, TSO, VM, or system application programs using 3270 BSC protocols, and provides support for the BSC portion of the 3270 Device Emulation feature.

3270 Device Emulation. A feature of the System Support Program Product that allows a System/36 local or remote device to appear as a 3270 device to another system.

3270 SNA Support subsystem. The subsystem that provides support for the SNA portion of the 3270 Device Emulation feature.

3278 Device Emulation. A feature of the System Support Program Product that allows a System/36 local or remote device to appear as a 3278 device to another system.

512-byte format. A format for diskette 1 diskettes with 512 bytes per sector and 8 sectors per track.

Index

Special Characters

+ (continuation character) 2-7
$ARSP utility program
changing alert indicators A-3
changing automatic response values A-3
$BICR utility program
copying basic data exchange files A-5
listing basic data exchange files A-4
$BMENU utility program (creates menus) A-7
$BUILD utility program (corrects disk files) A-9
$COPY utility program
additional functions A-32
copying, organizing disk files A-10
listing files A-14
restoring disk files A-18
restoring network resource directory A-23
saving disk files A-24
saving network resource directory A-31
$CPPE utility program (displays error messages) A-37
$CZUT utility program (specifies alert messages) A-38
$DCOPY utility program (copies diagnostic diskettes) A-38
$DDST utility program (sorts keys of indexed files) A-39
$DELET utility program (deletes files or libraries) A-40
$DPGP utility program (prints graphics file) A-42
$DUPRD utility program (copies diskettes) A-43
$FBLD utility program
creates alternative indexes A-46
creates new disk files A-44
$FEFIX utility program C-23
$FREE utility program (compresses disk space) A-47
$HELP utility program A-47
$HIST utility program (history file list, copy, and erase) A-48
$IDSET utility program (defines remote IDs for communications) A-49
$IEDS utility program (disables communications subsystems) A-50
$IENBL utility program (enables communications subsystems) A-51
$INIT utility program (prepares diskettes) A-52
$LABEL utility program (lists file and library names) A-53
$MAINT utility program
changing
library size (ALOCLIBR procedure) A-60
condensing library space A-63
copying
job streams A-74
library members to files A-66
members to libraries from files A-72
$MAINT utility program *(continued)*
copying library members A-64
creating
libraries A-56
creating library members A-57
listing library members A-75
from diskette or tape A-77
removing library members A-78
renaming library members A-64
restoring libraries A-80
saving libraries A-79
$MGBLD utility program (creates message members) A-82
$MMSP utility program (stops monitoring communications lines) A-83
$MMST utility program (starts monitoring communications lines) A-83
$PACK utility program (compresses disk space) A-84
$PNLM utility program (defines phone lists) A-84
$POST utility program
copying 5260 data files A-85
listing diskette files A-86
$PRCED utility program (edits user ID file) A-87
$PRCLT utility program (lists user ID file) A-87
$PRLST utility program (lists user ID file) A-88
$PRPWD utility program (changes user password) A-88
$PRUED utility program (edits user ID file) A-89
$PRUID utility program (password and badge security) A-90
$PRURS utility program (restores user ID file) A-91
$PRUSV utility program (saves user ID file) A-93
$RENAM utility program (renames disk files, libraries, and folders) A-94
$RREDT utility program (edits resource security file) A-95
$RRESC utility program (resource security) A-96
$RRLST utility program (lists resource security file) A-97
$RRSAV utility program (saves resource security file) A-98
$RRSTR utility program (restores resource security file) A-99
$RRTED utility program (edits resource security file) A-101
$RRTLT utility program (lists resource security file) A-101
$SETCF utility program
changing communications information A-102
changing Print key information A-105
defining display station environment A-104
$SETCP utility program (defines communications environment) A-106

$SFGR utility program (generates display formats) A-108
$SINCT utility program (restores network resource directory) A-110
$SINDL utility program (removes network resource directory) A-110
$SINR utility program (edits network resource directory) A-110
$SVCASRV utility program A-111
$TCOPY utility program
copying tape files A-112
$TINIT utility program (initializing tapes) A-114
$TMSERV utility program
copying folder members A-115
copying folders A-116
listing folder members A-117
moving folders A-119
reorganizing folders A-118
restoring folder members A-120
restoring folders A-121
$UASC utility program (displays spool file entries) A-122
$UASF utility program (copies spool file entries) A-123
$XNLM utility program (defines X.21 call lists) A-124
$XNSH utility program (defines short hold mode line configuration) A-124
$XREST utility program
restoring extended character file A-125
$XSAVE utility program
saving extended character file A-126
*** statement 3-9**
/ (equal to) conditional expression 3-50
/* statement 5-123
// * (informational message statement) 3-57
// ** (system console message statement) 3-59
> (greater than) conditional expression 3-52
restrictions 3-52
?C'value'? substitution expression 3-15
?CD? substitution expression 3-16
using with ERR procedure 4-176
?CLIB? substitution expression 3-18
?Cn? substitution expression 3-15
?DATE? substitution expression 3-18
?F'A,name'? substitution expression 3-19
?F'S,name'? substitution expression 3-18
?L'position,length'? substitution expression 3-19
?M'mic,position,length'? substitution expression 3-21
?MENU? substitution expression 3-22
?n? substitution expression 3-11
?n'value'? substitution expression 3-11
?nF'value'? substitution expression 3-12
?nR? substitution expression 3-13
?nR'mic'? substitution expression 3-14
?nT'value'? substitution expression 3-12
?PRINTER? substitution expression 3-22
?PROC? substitution expression 3-23
?R? substitution expression 3-13
?R'mic'? substitution expression 3-14
?SLIB? substitution expression 3-23
?SYSLIST? substitution expression 3-23
?TIME? substitution expression 3-24
?USER? substitution expression 3-24
?VOLID? substitution expression 3-25
?WS? substitution expression 3-26
#GCFR utility program (requests an X.21 feature) A-127
#LIBRARY
See also system library
restoring using RESTLIBR procedure 4-386
saving using SAVELIBR procedure 4-431
#STRTUP1 procedure 4-4
#STRTUP2 procedure 4-6
#188E188 translation table 5-60
#188E48 translation table E-12
#188E64 translation table E-14
#188E96 translation table E-16
#96E48 translation table E-8
#96E64 translation table E-10
= (equal to) conditional expression 3-50

A

ABEND OCL statement 5-7
acquiring
See allocating
activating subconsole support 6-2
ACTIVE conditional expression 3-29
active library
See current library
active message member, assigning 5-73
active printer groups, displaying 4-34
activity
printed output 5-83
stopping all system 6-48
system
measuring 4-473
restarting 6-36
tracing system
using TRACE procedure C-32
ADD utility control statement, $SFGR utility program A-108
adding
disk files to diskette files 4-416
disk files to tape files 4-416
display formats using FORMAT procedure 4-181
entries to system service log C-30
exchange file from disk to tape file using TAPECOPY 4-500
exchange file from tape to disk file using TAPECOPY 4-500

adding *(continued)*
folder members to diskette files using ARCHIVE procedure 4-22
folder members to tape files using ARCHIVE procedure 4-22
library members
to a file 4-193
to a new library 4-62
to an existing library 4-537
values to parameters 3-60
address, task control block, displaying 6-40
addressing characters
changing using ALTERCOM procedure 4-11
list of 4-17
alert indicators, changing 4-457
ALERT procedure 4-7
aligning, printer forms in printer 5-83
all parameters, passing to another procedure 5-64
ALL report file 4-474
ALLOCATE OCL statement 5-8
ALLOCATE utility control statement, $MAINT utility program A-56, A-60
allocating
communications sessions 5-106
disk files
to jobs 5-38
to programs 5-32
disk space for scratch and job files 5-104
diskette drive to a procedure 5-8
diskette files 5-43
display station to program 5-120
folders, to change the size of 4-8
libraries
to a job 5-67
to change the size of 4-9
message members to programs and procedures 5-73
printers to programs 5-83
region size for a program 5-103
tape drive to a procedure 5-8
tape files 5-48
allowing
See also releasing, starting
duplicate keys 5-40
jobs to be started 6-36
jobs to run 6-36
jobs to run from job queue 6-36
SSP-ICF sessions 6-36
system service display station to be used 6-36
ALOCFLDR procedure 4-8
ALOCLIBR procedure 4-9
alter
disk cache 4-72
ALTERBSC procedure 4-10
ALTERCOM procedure 4-11
alternative index, creating 4-59
alternative sectors
See BUILD procedure
alternative system console, transferring 6-15
ALTERSDL procedure 4-18
answer
automatic 4-13
manual 4-13
tone, defining 4-461
answering
See replying
APAR files
creating using APAR procedure C-2
listing using DUMP procedure C-7
APAR procedure C-2
apostrophes ('), indicating procedure parameter data 5-64
APPC
remote location status, displaying 6-40
starting session groups using STRTGRP procedure 4-493
stopping session groups using STOPGRP procedure 4-491
subsystems, specifying using SESSION OCL statement 5-106
APPLYPTF procedure
See PTF procedure
APPNINFO procedure 4-19
ARCHIVE procedure 4-22
ARCHIVE utility control statement, $TMSERV utility program
copying folder members A-115
$ARSP utility program (specifies automatic message responses) A-3
changing alert indicators A-3
changing automatic response values A-3
ASCII characters
for addressing and polling 4-18
table of F-1
ASM procedure 4-27
ASMLOAD procedure 4-29
ASMSAVE procedure 4-30
Assembler programs, compiling 4-27
assembling
See compiling
ASSIGN command 6-2
assigning
communications line to a program 5-21
message members to programs and procedures 5-73
phone list to a program 5-21
printer as the system printer 6-2
printer group printer IDs 4-34
values to parameters 3-11
forcing 3-12
prompting 3-13, 3-14
temporary 3-12
using EVALUATE statement 3-60

assumed values, assigning 3-11
forcing 3-12
prompting 3-13, 3-14
temporary 3-12
ATTR OCL statement 5-11
attributes
library member, description of 4-282
procedures
data 2-10
MRT 2-10
OCL statement logging 2-10
parameters 2-10
AUTO advance, changing 5-9
AUTO procedure 4-30
auto report program
compiling 4-31
cross-reference 4-32
auto response
See automatic response
AUTOC procedure 4-31
autocall
creating connection lists using DEFINX21 procedure 4-141
creating phone lists using DEFINEPN procedure 4-140
end of number characters, allowing 4-461
LISTDONE conditional expression 3-41
separator character, allowing 4-461
using COMM OCL statement 5-21
using SESSION OCL statement 5-106
automatic advance, diskette drive, changing 5-9
automatic answer 4-13
automatic response
severity level, specifying
using NOHALT OCL statment 5-79
using NOHALT procedure 4-305
specifying responses for using RESPONSE procedure 4-374

B

backing up
See saving
backup line, secondary, specifying 4-11
BACKUP procedure 4-34
badge security
adding or removing 4-441
defining numbers for 4-443
balancing, printer load 4-34
BALPRINT procedure 4-34
basic data exchange
copying files to or from diskette 4-542
description of 4-548
basic data exchange *(continued)*
for library members A-68
listing
using LISTFILE procedure 4-267
using the $BICR utility program A-4
preparing diskette for using INIT procedure 4-234
BASIC procedure 4-39
BASIC programs
converting BASICS procedure 4-42
cross-referencing using BASICS procedure 4-42
developing using BASIC procedure 4-39
running using BASICP procedure 4-40
running using BASICR procedure 4-41
BASICP procedure 4-40
BASICR procedure 4-41
BASICS procedure 4-42
BASLOAD procedure 4-44
BASSAVE procedure 4-45
batch
See job queue
belts, printer
creating image members for 5-61
for 3262 Printer E-1
specifying
using IMAGE OCL statement 5-59
using SET procedure 4-454
BGUATTR procedure 4-46
BGUCHART procedure 4-47
BGUDATA procedure 4-50
BGUGRAPH procedure 4-51
BGULOAD procedure 4-54
BGUSAVE procedure 4-55
$BICR utility program (copies basic data exchange files) A-5
binary synchronous communications
See BSC
blank compression, specifying 4-11
BLDFILE procedure 4-56
BLDINDEX procedure 4-59
BLDLIBR procedure 4-62
BLDMENU procedure 4-66
block, task control, displaying status of 6-40
blocking, changing
index 5-40
record 5-40
blocks
changing, folder size 4-8
changing, library size 4-9
converting to records and sectors B-3
file size, maximum 4-56, 5-34
BLOCKS conditional expression 3-30
$BMENU utility program (creates menus) A-7
books, related xiv

border, Print key, specifying
using PRINTKEY procedure 4-354
using SET procedure 4-454
using WORKSTN OCL statement 5-120
braces in syntax formats 1-3
brackets in syntax formats 1-3
branching in procedures 3-67
BSC
changing parameters
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
line monitoring
starting using STARTM procedure 4-489
stopping using STOPM procedure 4-492
subsystems, specifying using SESSION OCL statement 5-106
tracing using TRACE procedure C-32
$BUILD utility program (corrects disk files) A-9
BUILD procedure 4-70
building
disk files
using BLDFILE procedure 4-56
using COPYDATA procedure 4-111
using FILE OCL statement 5-32
libraries using BLDLIBR procedure 4-62
menus
using BLDMENU procedure 4-66
using SDA procedure 4-437
bypassing duplicate key checking 5-40
bytes
converting to blocks B-3
converting to sectors B-3

C

C & SM
problem management 4-7
procedures
ALERT 4-7
SETALERT 4-457
?C'value'? substitution expression 3-15
?Cn? substitution expression 3-15
CACHE procedure 4-72
calculating parameter values 3-60
calendars
batch printing 4-307
changing using OFCCAL procedure 4-309
creating using OFCCAL procedure 4-309
call list
See phone list
call lists, creating using DEFINX21 procedure 4-141
calling procedures
example of 2-8
using EVOKE OCL statement 5-30
using INCLUDE OCL statement 5-63
CANCEL command 6-5
CANCEL OCL statement 5-16
CANCEL statement 3-60
canceling
display station sessions 6-5
jobs
currently running 6-5
from job queue 6-5
preventing from using ATTR OCL statement 5-11
with a task dump 6-7
print entries 6-5
procedures using CANCEL statement 3-60
spool file entries using CANCEL OCL statement 5-16
canceling an X.21 facility using REQUESTX procedure 4-373
CATALOG procedure 4-73
caution
See precautions
?CD? substitution expression 3-16
using with ERR procedure 4-176
CEND statement, $MAINT utility program A-69
CGU procedure 4-91
CGULOAD procedure 4-92
CGUSAVE procedure 4-93
CHANGE command 6-9
CHANGE OCL statement 5-18
changing
alert indicators 4-457
allowing duplicate keys
using RESTORE procedure 4-392
AUTO/NOAUTO defaults 5-9
automatic response severity level 5-79
BASIC programs 4-39
BASIC source member to BASIC subroutine member 4-42
calendars using OFCCAL procedure 4-309
characters per inch
See also font
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
COBOL display formats using COBSDA procedure 4-103
COBOL programs
using COBOLONL procedure 4-102
using COBSEU procedure 4-104
communications line parameters 4-11
communications queues and routes using OFCMAINT procedure 4-320
data definitions using IDDUDFN procedure 4-223

changing *(continued)*
data definitions using IDDURBLD procedure 4-227
data dictionaries using IDDUDCT procedure 4-222
date
using DATE OCL statement 5-25
using DATE procedure 4-137
using SET procedure 4-454
date format using SET procedure 4-454
defaults using OFCDFLT procedure 4-313
directory size
using ALOCLIBR procedure 4-9
using RESTLIBR procedure 4-386
disk cache 4-72
disk files
using DFU (data file utility) 4-150
using ENTER procedure 4-172
using UPDATE procedure 4-550
disk files using QRY procedure 4-360
disk storage, using PATCH procedure C-17
diskette storage, using PATCH procedure C-17
display formats
using DSU procedure 4-163
using FORMAT procedure 4-181
using SDA procedure 4-437
using SEU procedure 4-465
using the $SFGR utility program A-108
documents using TEXTDOC procedure 4-512
DW/36 user profiles using TEXTPROF procedure 4-532
enrollment using OFCUSER procedure 4-324
error reporting in an IBM Token-Ring Network using TRNMGR procedure 4-549
file size
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
first parameter prompted for 5-98
folders using TEXTFLDR procedure 4-530
forms number
order printed in, using START command 6-37
using CHANGE command 6-9
using CHANGE OCL statement 5-18
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SET procedure 4-454
FORTRAN display formats using FORTSDA procedure 4-191
FORTRAN programs
using FORTONL procedure 4-187
using FORTSEU procedure 4-192
horizontal print density
See also font
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
index blocking 5-40

changing *(continued)*
job step date
using DATE OCL statement 5-25
using DATE procedure 4-137
key length or position
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
length of procedure parameters 5-99
library member name
using CHNGEMEM procedure 4-94
library member reference number
using CHNGEMEM procedure 4-94
library member subtype
using CHNGEMEM procedure 4-94
library or directory size
using ALOCLIBR procedure 4-9
using RESTLIBR procedure 4-386
library, sign-on, using SET procedure 4-454
lines per inch
See also font
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
lines per page
See also font
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SET procedure 4-454
local data area 5-70
menus
using DSU procedure 4-163
using SDA procedure 4-437
using SEU procedure 4-465
message members
using CREATE procedure 4-132
using DSU procedure 4-163
using SEU procedure 4-465
MRT number 5-11
network resource directory using EDITNRD procedure 4-167
never-ending program indicator 5-11
NOAUTO/AUTO defaults 5-9
number of copies to be printed
using CHANGE command 6-9
using CHANGE OCL statement 5-18
number of multiple requester terminals 5-11
pagesize of disk cache 4-72
paper drawer to be used
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
print belt image
using IMAGE OCL statement 5-59
using SET procedure 4-454

changing *(continued)*
print entry defer status 6-9
print image translation tables 5-59
Print key border, header, or printer
using PRINTKEY procedure 4-354
using SET procedure 4-454
using WORKSTN OCL statement 5-120
printed output
reducing size using FORMS OCL statement 5-55
reducing size using PRINT procedure 4-350
reducing size using PRINTER OCL statement 5-83
rotating using FORMS OCL statement 5-55
rotating using PRINTER OCL statement 5-83
printed output position 6-9
printer paper drawer to be used
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
printer to be used
using CHANGE command 6-9
using CHANGE OCL statement 5-18
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
printer, sign-on, using SET procedure 4-454
priority of spool writer 6-9
priority, processing
using ATTR OCL statement 5-11
using PRTY command 6-29
procedure parameter prompting sequence 5-98
procedures, using DSU procedure 4-163
procedures, using SEU procedure 4-465
program date
using DATE OCL statement 5-25
using DATE procedure 4-137
programs, using DSU procedure 4-163
programs, using SEU procedure 4-465
queries using QRY procedure 4-360
record blocking 5-40
reduction size of printed output
See also font
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
region size
using REGION OCL statement 5-103
using SET procedure 4-454
resource security file using SECEDIT procedure 4-443
rotating
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
rotation of printed output
using FORMS OCL statement 5-55

changing *(continued)*
rotation of printed output *(continued)*
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
RPG display formats using RPGSDA procedure 4-414
RPG programs
using RPGONL procedure 4-411
using RPGSEU procedure 4-414
separator pages, number of 6-9
session date
using DATE OCL statement 5-25
using DATE procedure 4-137
using SET procedure 4-454
severity level, automatic response 5-79
size of disk cache 4-72
source members, using DSU procedure 4-163
source members, using SEU procedure 4-465
space for ideographic character
using PRINTER OCL statement 5-83
subconsole support 6-2
supplemental dictionaries using TEXTDCT procedure 4-511
switch settings
using SWITCH OCL statement 5-112
using SWITCH procedure 4-496
system list device
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
system printer 6-2
user identification file using SECEDIT procedure 4-443
user profiles, DW/36, using TEXTPROF procedure 4-532
vertical print density
See also font
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
work station IDs 6-2
changing an x.25 configuration 4-297
characters
addressing, BSC 4-17
ASCII, table of F-1
EBCDIC, table of F-1
lowercase to uppercase, printing 5-59
maximum
JOBQ OCL statement 5-66
MSG OCL statement 5-76
procedure continuation 2-7
parameters, allowed in 5-64
polling, BSC, table of 4-17
printed per inch, changing
See also font

characters *(continued)*
printed per inch, changing *(continued)*
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
printed per line, changing
See also font
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
testing value of 3-50, 3-52
characters per inch
See font
chart of SSP utilities and OCL statements called by SSP procedures 4-2
chart of system tasks 1-7
check, program, specifying device for task dump 5-7
checking
See testing
checking, bypassing duplicate key 5-40
CHGXLATE procedure 4-93
CHNGEMEM procedure 4-94
clearing
menu from the display 6-22
the local data area 5-71
?CLIB? substitution expression 3-18
clocking, modem, specifying
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
CNFIGICF procedure 4-96
CNFIGSSP procedure 4-97
CNFIGX25 procedure 4-97
COBLOAD procedure 4-98
COBOL procedure 4-98
COBOL programs
compiling using COBOLC procedure 4-99
creating or changing
using COBOLONL procedure 4-102
using COBSEU procedure 4-104
creating or changing display formats using COBSDA procedure 4-103
COBOLC procedure 4-99
COBOLCG procedure 4-101
COBOLG procedure 4-102
COBOLONL procedure 4-102
COBOLP procedure 4-102
COBSAVE procedure 4-103
COBSDA procedure 4-103
COBSEU procedure 4-104
code, return, substitution expression 3-16
coding
procedures 2-1
rules, OCL statements 5-4
tips
procedure control expressions 2-23
coding *(continued)*
tips *(continued)*
procedures 2-22
collecting
See also gathering
free disk space using COMPRESS procedure 4-105
free folder space using CONDENSE procedure 4-109
free library space using CONDENSE procedure 4-109
information from a disk file using COPYDATA procedure 4-111
collecting diagnostic information C-2
combining substitution expressions 3-27
COMM OCL statement 5-21
command keys, help support 4-200
command syntax formats, description of 1-3
command, procedure 1-1
commands
See also control commands, procedures
preventing from processing 6-48
stopping 6-48
commas in syntax formats 1-3
comment (*) statement 3-9
comments
in OCL statements 5-5
in procedures 3-9
communications
allowing SSP-ICF sessions 6-36
autocall
creating a call list for 4-141
creating a phone list for 4-140
end of number characters, allowing 4-461
LISTDONE conditional expression 3-41
separator characters, allowing 4-461
call list, creating using DEFINX21 procedure 4-141
changing parameters
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
debugging subsystems using ICFDEBUG procedure 4-221
disabling subsystems using DISABLE procedure 4-157
display station, sending messages to 5-76, 6-24
displaying status of 6-40
enabling subsystems using ENABLE procedure 4-170
end of number characters, allowing 4-461
error reporting in an IBM Token-Ring Network 4-549
EVOKED conditional expression 3-38
finance subsystem, loading using LOAD3601 procedure 4-293
line speed, changing
using SETCOMM procedure 4-461
line status, displaying 6-40

communications *(continued)*
line type, changing
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
line, assigning to a program 5-21
line, defining
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
line, dropping or holding after signoff
using OFF command 6-27
using OFF OCL statement 5-81
line, placing online or offline 5-116, 6-53
loading finance subsystem using LOAD3601 procedure 4-293
monitoring
using STARTM procedure 4-489
using STOPM procedure 4-492
multiple session remote job entry 4-302
multipoint line, specifying
using ALTERCOM procedure 4-13
using SETCOMM procedure 4-461
nonswitched line, specifying
using ALTERCOM procedure 4-13
using SETCOMM procedure 4-461
operator, sending messages to 5-76, 6-24
personal computers, sending and receiving messages 6-24
phone list, creating using DEFINEPN procedure 4-140
point-to-point line, specifying
using ALTERCOM procedure 4-13
using SETCOMM procedure 4-461
remote IDs, defining using DEFINEID procedure 4-139
remote job entry 4-302
separator characters, allowing 4-461
SESSION OCL statements 5-106
setting up subsystems using CNFIGICF procedure 4-96
start monitoring line 4-489
starting APPC session groups using STRTGRP procedure 4-493
starting SSP-ICF sessions 6-36
status, displaying 6-40
stop monitoring line 4-492
stopping APPC session groups using STOPGRP procedure 4-491
stopping SSP-ICF sessions 6-48
switched line, specifying
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
testing for subsystems using ENABLED expression 3-37
X.21
canceling a facility 4-373
requesting a facility 4-373

communications *(continued)*
X.25 line, specifying
using SETCOMM procedure 4-461
Communications and Systems Management
problem management 4-7
procedures
ALERT 4-7
SETALERT 4-457
COMPILE OCL statement 5-23
compiling
Assembler programs 4-27
auto report RPG programs 4-31
COBOL programs
using COBOLC procedure 4-99
using COBOLONL procedure 4-102
display formats using FORMAT procedure 4-181
FORTRAN programs
using FORTONL procedure 4-187
using FORTRANC procedure 4-188
message members using CREATE procedure 4-132
RPG programs
using AUTOC procedure 4-31
using RPGC procedure 4-407
WSU program 4-554
COMPRESS procedure 4-105
COMPRESS utility control statement
$FREE utility program A-47
$MAINT utility program A-63
compressing
data when saving files on diskette using SAVE procedure 4-416
disk space using COMPRESS procedure 4-105
folder space using CONDENSE procedure 4-109
library space using CONDENSE procedure 4-109
compression, blanks, specifying 4-11
computer output reduction 4-350
CONDENSE procedure 4-109
condensing
disk space using COMPRESS procedure 4-105
folder space using CONDENSE procedure 4-109
library space using CONDENSE procedure 4-109
conditional expressions
/ (equal to) 3-50
> (greater than) 3-52
= (equal to) 3-50
ACTIVE 3-29
BLOCKS 3-30
CONSOLE 3-30
DATAF1 3-31
DATAI1 3-32
DATAT 3-34
DSPLY 3-36
ENABLED 3-37
EVOKED 3-38
INQUIRY 3-39
JOBQ 3-40

conditional expressions *(continued)*
LISTDONE 3-41
LOAD 3-42
MRT 3-43
PROC 3-44
restrictions 3-50, 3-52
SECURITY 3-45
SOURCE 3-46
SUBR 3-47
SWITCH 3-48
VOLID 3-53, 3-54
conditional procedure processing 3-28
configuration, communications, testing for using ENABLED expression 3-37
configuring
an X.25 network 4-97
communications subsystems using CNFIGICF procedure 4-96
the system using CNFIGSSP procedure 4-97
connection list
See phone list
considerations
See also precaution
file names 5-33
file record length 4-56
file size 4-56
library names 4-62
parameter coding 2-6
console
See system console
CONSOLE command 6-15
CONSOLE conditional expression 3-30
console, messages, replying to 6-32
console, system, restrictions 3-10
contents of procedures 2-1
continuing
diskette files 5-9
OCL statements 5-4
statements in procedures 2-7
maximum characters 2-7
continuous carrier, defining using SETCOMM procedure 4-461
control
block, task, displaying status of 6-40
procedure, expressions 3-1
unit, placing online or offline 5-116, 6-53
control commands 6-1
ASSIGN 6-2
CANCEL 6-5
CHANGE 6-9
CONSOLE 6-15
HOLD 6-18
INFOMSG 6-20
JOBQ 6-21
MENU 6-22
MODE 6-23
control commands 6-1 *(continued)*
MSG 6-24
OFF 6-27
POWER 6-28
PRTY 6-29
RELEASE 6-30
REPLY 6-32
RESTART 6-34
START 6-36
STATUS 6-40
STATUSF 6-45
STOP 6-48
TIME 6-52
VARY 6-53
control expressions, procedure, guide 3-2
conversion tables, multinational character set D-10
converting
blocks to records B-3
decimal to hexadecimal B-1
hexadecimal to decimal B-1
multinational characters D-1
table of D-10
records to blocks B-3
sectors to blocks B-3
TMS documents to DW/36 documents using TEXTCONV procedure 4-511
copies, printed output
changing using CHANGE command 6-9
changing using CHANGE OCL statement 5-18
specifying using PRINTER OCL statement 5-83
$COPY utility program
copying, organizing disk files A-10
listing files A-14
restoring disk files A-18
restoring network resource directory A-23
saving disk files A-24
saving network resource directory A-31
COPY and CEND statements, $MAINT utility program A-69
COPY utility control statement, $MAINT utility program
changing library or directory size A-60
copying from library to library A-64
copying job stream files A-74
copying library members
from a file A-72
to a file A-66
creating libraries and copying members from a file A-56
listing libraries A-75, A-77
COPYADD utility control statement, $COPY utility program A-24
COPYALL utility control statement, $COPY utility program A-18, A-24

COPYDATA procedure 4-111
COPYDIAG procedure 4-120
COPYFILE utility control statement, $COPY utility program
copying or organizing disk files A-10
DELETE parameter A-32
listing files A-14
OUTPUT parameter A-32
restoring disk files A-18
restoring network resource directory A-23
saving disk files A-24
saving network resource directory A-31
copying
APAR information C-2
diagnostic information C-2
disk files
basic data exchange format 4-542
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
using SAVE procedure 4-416
WSU transaction using WSUTXEX procedure 4-561
diskettes, using COPYI1 procedure 4-121
documents using TEXTDOC procedure 4-512
exchange file from tape to disk file using TAPECOPY procedure 4-500
exchange files from disk to tape file using TAPECOPY procedure 4-500
folder members
using ARCHIVE procedure 4-22
using RETRIEVE procedure 4-401
folders using SAVEFLDR procedure 4-428
history file using HISTORY procedure 4-209
libraries
basic data exchange format A-66
from a file using TOLIBR procedure 4-537
record mode format A-66
restoring using RESTLIBR procedure 4-386
saving using SAVELIBR procedure 4-431
to a file using FROMLIBR procedure 4-193
to another library using LIBRLIBR procedure 4-255
using the $MAINT utility program A-64
library members 4-62
basic data exchange format A-66
from a file using TOLIBR procedure 4-537
record mode format A-66
to a file using FROMLIBR procedure 4-193
to another library using LIBRLIBR procedure 4-255
using the $MAINT utility program A-64
network resource directory
using RESTNRD procedure 4-389
using SAVENRD procedure 4-434
resource security file
using SECREST procedure 4-448
using SECSAVE procedure 4-451
copying *(continued)*
special E-format diskette files using POST procedure 4-345
spool file entries using COPYPRT procedure 4-126
user identification file
using SECREST procedure 4-448
using SECSAVE procedure 4-451
COPYI1 procedure 4-121
COPYI1 utility control statement, $DUPRD utility program A-43
COPYLIBR utility control statement, $MAINT utility program
restoring libraries A-80
saving libraries A-79
COPYPRT Files, File Format for 4-128
COPYPRT procedure 4-126
correcting disk file data using BUILD procedure 4-70
CPI
See characters per inch
See font
$CPPE utility program (displays error messages) A-37
CREATE procedure 4-132
CREATE utility control statement, $SFGR utility program A-108
creating
alternative index files 4-59
calendars using OFCCAL procedure 4-309
COBOL display formats using COBSDA procedure 4-103
COBOL programs
using COBOLONL procedure 4-102
using COBSEU procedure 4-104
connection list using DEFINX21 procedure 4-141
data definitions from RPG specs using IDDUXLAT procedure 4-227
data definitions using IDDUDFN procedure 4-223
data dictionaries using IDDUDCT procedure 4-222
direct disk files using BLDFILE procedure 4-56
disk cache 4-72
disk files
using BLDFILE procedure 4-56
using DFU (data file utility) 4-150
using ENTER procedure 4-172
using FILE OCL statement 5-32
using QRY procedure 4-360
display formats
using DSU procedure 4-163
using FORMAT procedure 4-181
using SDA procedure 4-437
using SEU procedure 4-465
using the $SFGR utility program A-108
documents using TEXTDOC procedure 4-512
DW/36 user profiles using TEXTPROF procedure 4-532
folders using TEXTFLDR procedure 4-530
FORTRAN display formats using FORTSDA procedure 4-191

creating *(continued)*
FORTRAN programs
using FORTONL procedure 4-187
using FORTSEU procedure 4-192
indexed disk files using BLDFILE procedure 4-56
libraries using BLDLIBR procedure 4-62
library members
using DSU procedure 4-163
using SEU procedure 4-465
using the $MAINT utility program A-57
menus
using BLDMENU procedure 4-66
using DSU procedure 4-163
using SDA procedure 4-437
using SEU procedure 4-465
using the $BMENU utility program A-7
message members
using CREATE procedure 4-132
using DSU procedure 4-163
using SEU procedure 4-465
network resource directory using EDITNRD procedure 4-167
phone list
using DEFINEPN procedure 4-140
using DEFINX21 procedure 4-141
print belt members 5-61
procedures 2-1
using DSU procedure 4-163
using SEU procedure 4-465
using the $MAINT utility program A-57
programs, using DSU procedure 4-163
programs, using SEU procedure 4-465
queries using QRY procedure 4-360
reports using QRY procedure 4-360
resource security file, using SECDEF procedure 4-441
RPG display formats using RPGSDA procedure 4-414
RPG display station programs, using SDA procedure 4-437
RPG programs
using RPGONL procedure 4-411
using RPGSEU procedure 4-414
sequential disk files using BLDFILE procedure 4-56
source members
using DSU procedure 4-163
using SEU procedure 4-465
using the $MAINT utility program A-57
unique names 2-24
user identification file, using SECDEF procedure 4-441
user profiles, DW/36, using TEXTPROF procedure 4-532
WSU programs
using SDA procedure 4-437
using WSU procedure 4-554
creating *(continued)*
WSU transaction file, using WSUTXCR procedure 4-559
creation date for disk files 5-37
cross-referencing
auto report RPG program 4-32
RPG programs
using RPGC procedure 4-408
using RPGX procedure 4-415
CRT
See display station
current library
changing
using LIBRARY OCL statement 5-67
using MENU command 6-22
using MENU OCL statement 5-75
using SLIB procedure 4-471
substitution expression 3-18
currently running jobs, status of 6-40, 6-45
$CZUT utility program (specifies alert messages) A-38

D

danger
See precaution
data
attribute, procedures 2-10
prompting for 5-97
data area, local
changing 5-70
clearing 5-71
data areas, listing, using DUMP procedure C-7
data collection file
deleted from disk 4-475
name 4-475
data definitions
changing using IDDUDFN procedure 4-223
changing using IDDURBLD procedure 4-227
creating from RPG specs using IDDUXLAT procedure 4-227
creating using IDDUDFN procedure 4-223
linking using IDDULINK procedure 4-224
printing using IDDUPRT procedure 4-226
data dictionaries
changing using IDDUDCT procedure 4-222
creating using IDDUDCT procedure 4-222
data exchange, basic, for libraries A-68
data file utility procedures
DFU 4-150
ENTER 4-172
INQUIRY 4-238
LIST 4-259
UPDATE 4-550

data, procedure parameter 5-64
DATAF1 conditional expression 3-31
DATAI1 conditional expression 3-32
DATAT conditional expression 3-34
date
creation for disk files 5-37
displaying 6-52
job step, changing
using DATE OCL statement 5-25
using DATE procedure 4-137
program, changing
using DATE OCL statement 5-25
using DATE procedure 4-137
session, changing
using DATE OCL statement 5-25
using DATE procedure 4-137
using SET procedure 4-454
date format, changing using SET procedure 4-454
DATE OCL statement 5-25
DATE procedure 4-137
?DATE? substitution expression 3-18
days, retention, diskette files 5-45
days, retention, tape files 5-51
$DDST utility program (sorts keys of indexed files) A-39
DEALLOC OCL statement 5-27
deallocating the diskette drive 5-27
DEBUG OCL statement 5-28
debugging
communications subsystems using ICFDEBUG procedure 4-221
procedures 2-24
using DEBUG OCL statement 5-28
using LOG OCL statement 5-72
using LOG procedure 4-294
programs
using TRACE procedure C-32
programs using SETDUMP procedure C-31
RPG programs using RPGONL procedure 4-411
system
using the SERVICE procedure C-29
using TRACE procedure C-32
decimal converting to and from hexadecimal B-1
defaults
assigning 3-10
forcing 3-12
prompting 3-13, 3-14
temporary 3-12
changing using OFCDFLT procedure 4-313
procedure parameter 2-5
defer status, printed output, changing
using CHANGE command 6-9
using PRINTER OCL statement 5-83
deferring printed output 5-83
DEFINEID procedure 4-139
DEFINEID utility control statement, $IDSET utility program A-49
DEFINEPN procedure 4-140
defining
message file using MSGFILE procedure 4-299
definitions
job step 2-1
procedure 1-1, 2-1
procedure command 1-1
utility programs 1-2
DEFINLOC procedure 4-140
DEFINX21 procedure 4-141
DEFINX25 procedure 4-142
DEFSUBD procedure 4-143
$DELET utility program (deletes files or libraries) A-40
DELETE procedure 4-144
DELETE utility control statement
$MAINT utility program A-78
$SFGR utility program A-108
delete-capable files
creating
using BLDFILE procedure 4-57
using FILE OCL statement 5-40
removing records from
using COPYDATA procedure 4-117
using SAVE procedure 4-416
deleting
See also removing
disk cache 4-72
DETAIL report file 4-474
developing
BASIC programs 4-39
COBOL programs 4-102
FORTRAN programs 4-187
RPG programs 4-411
device emulation, 3270
using EM3270 procedure 4-168
using ES3270 procedure 4-178
device error information, listing C-16
device for task dump, specifying 5-7
DFA procedure C-5
DFU (data file utility) procedures
DFU 4-150
ENTER 4-172
INQUIRY 4-238
LIST 4-259
UPDATE 4-550
DFU procedure 4-150
DFULOAD procedure 4-151
DFUSAVE procedure 4-152
diagnostic information, collecting C-2
DICTLOAD procedure 4-153
DICTSAVE procedure 4-155
direct disk files, creating, using BLDFILE procedure 4-56

directions
coding OCL statements 5-4
reading statement formats 1-3
directory
changing the size of 4-9
library, sample listing of 4-277
library, specifying size of 4-62
to procedure control expressions 3-2
to using the system support 1-7
to utilities and statements called by SSP procedures 4-2
DISABLE procedure 4-157
DISABLE utility control statement, $IEDS utility program A-50
disk
changing using PATCH procedure C-17
copying folder members from using RETRIEVE procedure 4-401
copying folder members to using ARCHIVE procedure 4-22
copying folders to using SAVEFLDR procedure 4-428
copying libraries to
record mode format A-66
using FROMLIBR procedure 4-193
gathering free space
in folders using CONDENSE procedure 4-109
in libraries using CONDENSE procedure 4-109
on disk using COMPRESS procedure 4-105
restoring
folder members from 4-401
saving
folder members on 4-22
folders on 4-428
disk files
adding exchange files from disk to tape file using TAPECOPY procedure 4-500
adding exchange files from tape to disk file using TAPECOPY procedure 4-500
allocating
to entire jobs 5-32
to programs 5-32
allowing duplicate keys
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
alternative index, creating 4-59
blocking, index, changing 5-40
blocking, record, changing 5-40
bypassing duplicate key checking 5-40
changing
name of using RENAME procedure 4-372
using DFU (data file utility) 4-150
using ENTER procedure 4-172
using UPDATE procedure 4-550
changing size of
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
disk files *(continued)*
compressing space using COMPRESS procedure 4-105
copying
basic data exchange diskette files 4-542
libraries to using FROMLIBR procedure 4-193
library members to 4-537
special E-format diskettes using POST procedure 4-345
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
using SAVE procedure 4-416
WSU transaction files using WSUTXEX procedure 4-561
copying exchange files from disk to tape file using TAPECOPY procedure 4-500
copying exchange files from tape to disk file using TAPECOPY procedure 4-500
creating
basic data exchange diskette files 4-542
from special E-format diskettes using POST procedure 4-345
using BLDFILE procedure 4-56
using DFU (data file utility) 4-150
using ENTER procedure 4-172
using FILE OCL statement 5-32
using QRY procedure 4-360
creation date for 5-37
DATAF1 conditional expression 3-31
delete-capable, creating
using BLDFILE procedure 4-57
using FILE OCL statement 5-40
deleting
using DELETE procedure 4-144
using RETAIN parameter of FILE OCL statement 5-32
deleting records from
using COPYDATA procedure 4-111
using SAVE procedure 4-416
displaying
using INQUIRY procedure 4-238
using LISTDATA procedure 4-261
using LISTFILE procedure 4-267
using the $COPY utility program A-14
displaying using QRYRUN procedure 4-363
duplicate key checking, bypassing 5-40
duplicate keys, allowing 5-40
existence testing 3-31
extending
using BLDFILE procedure 4-56
using FILE OCL statement 5-39
FILE OCL statement for 5-32
index blocking, changing 5-40
key length, specifying 4-57
key position, specifying 4-57
key sorting
before system shutdown 6-48

disk files *(continued)*
key sorting *(continued)*
using KEYSORT procedure 4-252
keys, duplicate allowing 5-40
linking using IDDUDISK procedure 4-223
listing
using DFU (data file utility) 4-150
using LIST procedure 4-259
using LISTDATA procedure 4-261
using LISTFILE procedure 4-267
using the $COPY utility program A-14
listing using QRYRUN procedure 4-363
location of, specifying
using BLDFILE procedure 4-56
using FILE OCL statement 5-32
multinational character set conversion D-1
record blocking, changing 5-40
removing
using DELETE procedure 4-144
using RETAIN parameter of FILE OCL statement 5-32
renaming using RENAME procedure 4-372
reorganizing
using COPYDATA procedure 4-111
using SAVE procedure 4-416
reserving space for job and scratch 5-104
restoring using RESTORE procedure 4-392
retention types 5-36
saving using OFCDATA procedure 4-312
saving using SAVE procedure 4-416
sorting
ideographic data using SRTX procedure 4-484
index keys using KEYSORT procedure 4-252
using DFU (data file utility) 4-150
using LIST procedure 4-259
using SORT procedure 4-483
using SRTX procedure 4-484
waiting to become available 5-39
WSU transaction, creating, using WSUTXCR procedure 4-559
disk space
gathering into an area
using COMPRESS procedure 4-105
using the $FREE utility program A-47
using the $PACK utility program A-84
testing for 3-30
diskettes
AUTO/NOAUTO parameters, changing defaults for 5-8
automatic advance, changing 5-9
basic data exchange
transferring disk files using TRANSFER 4-542
transferring library members using the $MAINT utility program A-66
changing using PATCH procedure C-17
copying disk files to 4-416
diskettes *(continued)*
copying folder members from using RETRIEVE procedure 4-401
copying folder members to using ARCHIVE procedure 4-22
copying folders to using SAVEFLDR procedure 4-428
copying libraries from 4-386
copying libraries to 4-193
basic data exchange format A-66
record mode format A-66
copying network resource directory to 4-434
copying using COPYI1 procedure 4-121
drive
allocating to a job 5-8
deallocating from a job 5-27
online or offline, placing 5-116, 6-53
status, displaying 6-40, 6-45
waiting for the allocate 5-10
dump, specifying 5-7
erasing
using DELETE procedure 4-144
using INIT procedure 4-233
using the $DELET utility program A-40
using the $INIT utility program A-52
FILE OCL statement for 5-43
files
continuing a series of 5-9
DATAI1 conditional expression 3-32
existence testing 3-32
listing using LISTFILE procedure 4-267
permanent 5-45
retention days 5-45
specifying for a program 5-43
formats of 4-548
preparing
using INIT procedure 4-233
using the $INIT utility program A-52
renaming
using INIT procedure 4-233
using the $INIT utility program A-52
restoring
disk files from 4-392
folder members from 4-401
libraries from 4-386
network resource directory from 4-389
retention days 5-45
saving
disk files on 4-416
folder members on 4-22
folders on 4-428
libraries on 4-431
network resource directory on 4-434
special E-format files, transferring using POST procedure 4-345
testing for 3-53

diskettes *(continued)*
volume ID, assigning
using INIT procedure 4-233
using the $INIT utility program A-52
display formats
adding
using FORMAT procedure 4-181
using the $SFGR utility program A-108
changing
using FORMAT procedure 4-181
using SDA procedure 4-437
using SEU procedure 4-465
using the $SFGR utility program A-108
creating
using FORMAT procedure 4-181
using SDA procedure 4-437
using SEU procedure 4-465
using the $SFGR utility program A-108
prompting for procedure data 5-97
removing
using FORMAT procedure 4-181
using REMOVE procedure 4-370
using the $SFGR utility program A-108
using switches to modify indicators 5-99
display IDs
exchanging using ASSIGN command 6-2
specifying for a program using WORKSTN OCL statement 5-120
DISPLAY procedure 4-158
display stations
allocating to programs 5-120
canceling sessions 6-5
exchanging IDs using ASSIGN command 6-2
ID substitution 3-26
local data area
substituting data from 3-19
local data area, changing data in 5-70
messages, sending and receiving 5-76, 6-24
online or offline, placing 5-116, 6-53
releasing from procedures 5-11
restoring after programs 5-120
sending messages to 5-76
signing off
using OFF command 6-27
using OFF OCL statement 5-81
status, displaying 6-40, 6-45
system service
allowing to be used 6-36
stopping 6-48
type of, determining 3-36
DISPLAY utility control statement
$BICR utility program A-4
$HIST utility program A-48
$LABEL utility program A-53
$POST utility program A-86
displaying
See also listing
APPC remote location status 6-40
basic data exchange diskette files, using LISTFILE procedure 4-267
communications queues and routes using OFCMAINT procedure 4-320
communications status 6-40
date and time 6-52
device error information C-16
disk files
using INQUIRY procedure 4-238
using LISTDATA procedure 4-261
using LISTFILE procedure 4-267
using QRYRUN procedure 4-363
using the $COPY utility program A-14
diskette files, using LISTFILE procedure 4-267
history file using HISTORY procedure 4-209
I-exchange diskette files, using LISTFILE procedure 4-267
job queue status 6-40, 6-45
libraries
on disk using LISTLIBR procedure 4-272
on diskette using LISTFILE procedure 4-267
on tape using LISTFILE procedure 4-267
using the $MAINT utility program A-75
mail folders using OFCMAINT procedure 4-320
mail using OFCMAIL procedure 4-318
menus
using MENU command 6-22
using MENU OCL statement 5-75
messages 5-76, 6-24
preventing the displaying of using procedure control expressions 2-24
messages at the system console 3-59
messages, informational 3-57
MSRJE status 6-40
office directory using OFCDIR procedure 4-314
online information using READINFO procedure 4-368
printed output status 6-40, 6-45
printer groups, active 4-34
procedure prompt displays 5-97
reports using QRYRUN procedure 4-363
session status 6-40, 6-45
spool file entries using COPYPRT procedure 4-126
spool writer status 6-40
SSP-ICF session status 6-40
SSP-ICF subsystem status 6-40
tape files, using LISTFILE procedure 4-267
time and date 6-52
displaying volume statistical logs
using TAPESTAT procedure 4-509

DLSLOAD
procedure 4-159
DLSSAVE
procedure 4-159
DOCCNV procedure 4-160
DOCPLOAD procedure 4-161
DOCPSAVE procedure 4-162
documents
changing using TEXTDOC procedure 4-512
converting using TEXTCONV procedure 4-511
copying
using ARCHIVE procedure 4-22
using RETRIEVE procedure 4-401
copying using TEXTDOC procedure 4-512
creating using TEXTDOC procedure 4-512
displaying using READINFO procedure 4-368
exchanging between folders and virtual disks or diskettes 4-332
printing using TEXTDOC procedure 4-512
releasing for printing using TEXTREL procedure 4-534
removing from folders using ARCHIVE procedure 4-22
renaming using TEXTDOC procedure 4-512
restoring to folders using RETRIEVE procedure 4-401
saving using ARCHIVE procedure 4-22
$DPGP utility program (prints graphics file) A-42
drive, diskette
allocating to a job 5-8
deallocating from a job 5-27
waiting for the allocate 5-10
drive, tape
allocating to a job 5-8
dropping communications line after signoff
using OFF command 6-27
using OFF OCL statement 5-81
DSPLY conditional expression 3-36
DSU procedure 4-163
DSULOAD procedure 4-166
DSUSAVE procedure 4-167
dump
canceling a job and taking a task 6-7
files, listing using DUMP procedure C-7
of program storage using SETDUMP procedure C-31
task, specifying device 5-7
dump file analysis procedure C-5
DUMP procedure C-7
duplicate keys
checking, bypassing 5-40
specifying
using BLDFILE procedure 4-57
using FILE OCL statement 5-40
$DUPRD utility program (copies diskettes) A-43
DW/36
changing documents 4-512
changing folders 4-530
changing supplemental dictionaries 4-511
changing user profiles 4-532
converting TMS documents 4-511
creating documents 4-512
creating folders 4-530
creating user profiles 4-532
displaying documents created by using READINFO procedure 4-368
displaying online information for using READINFO procedure 4-368
releasing documents for printing 4-534

E

E-format diskette files, transferring, using POST procedure 4-345
EBCDIC characters
for addressing and polling 4-17
table of F-1
editing
See changing
EDITNRD procedure 4-167
ELSE expressions 3-55
EM3270 procedure 4-168
ENABLE procedure 4-170
ENABLE utility control statement, $IENBL utility program A-51
ENABLED conditional expression 3-37
end of data statement (/*) 5-123
end of number characters, autocall, allowing 4-461
ending
other display station sessions by signing them off 6-5
powering off from a procedure 5-82
procedures
using CANCEL statement 3-60, 3-71
signing off from a procedure 5-81
signing off your display station 6-27
enrolling office users using OFCUSER procedure 4-324
ENTER procedure 4-172
entering procedures into the system 2-2
entries
history file
copying to a file 4-209
erasing 4-209
listing 4-209
job queue
canceling 6-5
holding from running 6-18
releasing for running 6-30

entries *(continued)*
job queue *(continued)*
starting to run 6-36
spool file
canceling using CANCEL command 6-5
canceling using CANCEL OCL statement 5-16
changing using CHANGE command 6-9
changing using CHANGE OCL statement 5-18
displaying status of 6-40, 6-45
holding from printing 6-18
releasing for printing 6-30
restarting the printing of 6-34
starting printing of using START command 6-36
starting printing of using START OCL statement 5-109
stopping printing of using STOP command 6-48
stopping printing of using STOP OCL statement 5-111
entry, remote jobs 4-302
EPDOWNL procedure 4-174
EPLMRG procedure 4-174
EP3270 procedure 4-175
= (equal to) conditional expression 3-50
ERAP procedure C-16
erasing
See also removing
diskettes
using DELETE procedure 4-144
using INIT procedure 4-233
using the $INIT utility program A-52
history file using HISTORY procedure 4-209
tapes
using TAPEINIT procedure 4-507
ERR procedure 4-176
ERR utility control statement, $CPPE utility program A-37
error
reporting in an IBM Token-Ring Network 4-549
retry count, changing
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
wait time, changing
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
error information, device, listing C-16
error messages
displaying
using ERR procedure 4-176
using PAUSE statement 3-69
replying to using REPLY command 6-32
ES3270 procedure 4-178
EVALUATE statement 3-60
EVOKE OCL statement 5-30
EVOKED conditional expression 3-38
evoking a procedure 5-30
examples
combining parameters 2-24
creating a procedure using the $MAINT utility program A-59
creating unique names 2-24
ELSE expressions 2-22, 3-55
files in procedures 2-1
GOTO and TAG statements 2-22
IF expressions 2-22
LISTKEYS procedure 2-13
message member 2-14
printers in procedures 2-1
procedures 2-4, 2-11
coding tips 2-22
continuation 2-7
nesting 2-8
substitution expressions 2-24
PROMPT OCL statement 2-16
SCRNPRT procedure 2-16
exchange formats
basic data
description of 4-548
preparing diskette for using INIT procedure 4-234
transferring data using 4-542
I-exchange
description of 4-548
transferring data using 4-542
exchange work station IDs 6-2
exchange, basic data, for library members A-68
executing
See running
expressions
ELSE 3-55
IF conditional 3-28
procedure control 3-1
substitution 3-10
testing value of 3-50, 3-52
extended character file, restoring 4-380
extended character file, saving 4-425
extents
for disk files, specifying
using BLDFILE procedure 4-58
using FILE OCL statement 5-39
for folders, compressing
using ALOCFLDR procedure 4-8
for libraries, compressing
using ALOCLIBR procedure 4-9
using CONDENSE procedure 4-109
EXTRACT procedure 4-180

F

?F'A,name'? substitution expression 3-19
?F'S,name'? substitution expression 3-18
?nF'value'? substitution expression 3-12
$FBLD utility program
creates alternative indexes A-46
creates new disk files A-44
File Format for COPYPRT Files 4-128
File Format for PRTFILE Files 4-516
file groups, naming conventions 4-417
file name, printer name 5-83
FILE OCL statement
for disk files 5-32
for diskette files 5-43
for tape files 5-48
FILE utility control statement, $FBLD utility program A-44, A-46
files
See also disk files, diskette files, and libraries
alternative index, creating 4-59
block number of, specifying 4-57
correcting data using BUILD procedure 4-70
delete-capable, specifying 4-57
disk
allocating 5-32
allocating to jobs 5-38
blocking, changing 5-40
bypassing duplicate key checking 5-40
copying using COPYDATA procedure 4-111
creating using FILE OCL statement 5-32
delete-capable, indicating 5-40
duplicate key checking, bypassing 5-40
duplicate keys, allowing 5-40
extending 5-39
keys, duplicate, allowing 5-40
record blocking, changing 5-40
retention types 5-36
waiting to become available 5-39
diskette
continuing a series of 5-9
permanent 5-45
processing a series of 5-9
retention days 5-45
specifying for a program 5-43
groups, naming conventions 4-417
key length, specifying 4-57
key position, specifying 4-57
listing names, using CATALOG procedure 4-73
location, specifying 4-57
maximum record length 4-56
procedure, example 2-1
record length 4-56
renaming, using RENAME procedure 4-372
files *(continued)*
size, substitution expression 3-18
specifying size of 4-56
tape
permanent 5-51
retention days 5-51
specifying for a program 5-48
Files, File Format for COPYPRT 4-128
Files, File Format for PRTFILE 4-516
finance controllers, testing using STATEST procedure 4-490
finance subsystem, loading using LOAD3601 procedure 4-293
finishing, signing off 6-27
first-level message members
assigning to programs and procedures 5-73
creating using CREATE procedure 4-132
displaying
using // * statement 3-57
using // ** statement 3-59
using ERR procedure 4-176
substitution expression 3-21
fixed disk
See disk
fixed-format menus, creating using BLDMENU procedure 4-66
flexible disk
See diskette
floppy disk
See diskette
folder
move from one disk location to another 4-298
folder members
copying
using ARCHIVE procedure 4-22
using RETRIEVE procedure 4-401
removing using ARCHIVE procedure 4-22
restoring using RETRIEVE procedure 4-401
saving using ARCHIVE procedure 4-22
folders
allocating using ALOCFLDR procedure 4-8
changing
name of using rename procedure 4-372
using TEXTFLDR procedure 4-530
changing the size of
using ALOCFLDR procedure 4-8
copying
to diskette using SAVEFLDR procedure 4-428
to tape using SAVEFLDR procedure 4-428
creating using TEXTFLDR procedure 4-530
exchanging data between virtual disks or diskettes and 4-332
listing names, using CATALOG procedure 4-73
removing
members from using ARCHIVE procedure 4-22
using DELETE procedure 4-144

folders *(continued)*
renaming using RENAME procedure 4-372
restoring members to using RETRIEVE procedure 4-401
restoring using RESTFLDR procedure 4-383
saving using OFCDATA procedure 4-312
saving using SAVEFLDR procedure 4-428
font 5-93
Format for COPYPRT Files, File 4-128
Format for PRTFILE Files, File 4-516
FORMAT procedure 4-181
formats, display
See display formats
formats, syntax, description of 1-3
forms number
changing
using CHANGE command 6-9
using CHANGE OCL statement 5-18
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SET procedure 4-454
printing order, using to control 6-37
FORMS OCL statement 5-55
forms, aligning in printer 5-83
FORTC procedure 4-184
FORTCG procedure 4-184
FORTG procedure 4-184
FORTGO procedure 4-185
FORTLOAD procedure 4-186
FORTONL procedure 4-187
FORTP procedure 4-187
FORTRAN
creating or changing using DSU procedure 4-163
FORTRAN programs
compiling using FORTRANC procedure 4-188
creating or changing
using FORTONL procedure 4-187
using FORTSEU procedure 4-192
creating or changing display formats using FORTSDA procedure 4-191
developing using FORTONL procedure 4-187
displaying menu of procedures 4-187
FORTRANC procedure 4-188
FORTSAVE procedure 4-191
FORTSDA procedure 4-191
FORTSEU procedure 4-192
$FREE utility program (compresses disk space) A-47
free-format menus, creating using BLDMENU procedure 4-66
FROMLIBR procedure 4-193
full, disk, fixing
using COMPRESS procedure 4-105
using the $FREE utility program A-47
using the $PACK utility program A-84
full, library, fixing using CONDENSE procedure 4-109
function directory 1-7
function keys, codes returned from prompt display 3-16

G

gathering
free disk space using COMPRESS procedure 4-105
free folder space using CONDENSE procedure 4-109
free library space using CONDENSE procedure 4-109
#GCFR utility program (requests an X.21 feature) A-127
generating
See also compiling
display formats
using FORMAT procedure 4-181
using the $SFGR utility program A-108
message members using CREATE procedure 4-132
WSU programs 4-554
getting
See allocating
giving the system console 6-15
GOTO statement 3-67
> (greater than) conditional expression 3-52
groups, files, naming conventions 4-417

H

halting
See stopping
halts
See messages
header, Print key, specifying
using PRINTKEY procedure 4-354
using SET procedure 4-454
using WORKSTN OCL statement 5-120
help
conceptual information xiv
messages, getting for 6-32
other manuals xiv
help information, getting from the system 4-199
HELP procedure 4-199
help support menus, logging options to history file 5-72
hexadecimal converting to and from decimal B-1
$HIST utility program (history file list, copy, and erase) A-48
HISTCOPY procedure 4-207

HISTCRT procedure 4-206
history file
automatic copy, description of 4-207
copying using HISTORY procedure 4-209
displaying using HISTORY procedure 4-209
erasing using HISTORY procedure 4-209
logging help support options to 5-72
logging OCL statements
procedure attribute 2-10
using LOG OCL statement 5-72
using LOG procedure 4-294
printing using HISTORY procedure 4-209
resetting using HISTORY procedure 4-209
HISTORY procedure 4-209
HOLD command 6-18
holding
communications line after signoff
using OFF command 6-27
using OFF OCL statement 5-81
jobs on job queue 6-18
print entries on spool file
using HOLD command 6-18
using PRINTER OCL statement 5-83
horizontal print density, changing
See also font
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
how to create a procedure 2-1

I

I-exchange files
copying 4-542
description of 4-548
IBM Token-Ring Network, error reporting in 4-549
IBM 5224 printer, considerations 5-55, 5-83
IBM 5225 printer, considerations 5-55, 5-83
IBM 5260 Retail System, transferring information using POST procedure 4-345
ICFDEBUG procedure 4-221
ICVERIFY procedure 4-222
ID, remote, changing 4-11
IDDU
changing data definitions 4-223, 4-227
changing data dictionaries 4-222
creating data definitions 4-223
creating data definitions from RPG specs 4-227
creating data dictionaries 4-222
displaying online information for using READINFO procedure 4-368
externally described data, explanation of 4-227
linking disk files 4-223
IDDU *(continued)*
linking files with definitions 4-224
printing data definitions 4-226
translating RPG specs to data definitions 4-227
IDDUDCT procedure 4-222
IDDUDFN procedure 4-223
IDDUDISK procedure 4-223
IDDULINK procedure 4-224
IDDUPRT procedure 4-226
IDDURBLD procedure 4-227
IDDUXLAT procedure 4-227
identifying tape errors
using TAPESTAT procedure 4-509
ideographic data
combining #KAMAST and #KACTIVE files using SRTXBLD procedure 4-485
sorting using SRTX procedure 4-484
ideographic extended character file, restoring 4-380
ideographic extended character file, saving 4-425
$IDSET utility program (defines remote IDs A-49
$IEDS utility program (disables subsystems) A-50
$IENBL utility program (enables subsystems) A-51
IF conditional expressions 3-28
ELSE statement 3-55
statement portion 3-54
using 2-6
IF tests 3-28
IFF tests 3-28
IFORMAT
See I-exchange files
IFT tests 3-28
IMAGE OCL statement 5-59
images, creating print belt 5-61
INCLUDE OCL statement 5-63
increasing the size of a disk file 5-39
index blocking, changing 5-40
indexed files
allowing duplicate keys
from special E-format diskettes 4-345
using COPYDATA procedure 4-111
alternative index, creating 4-59
creating
from basic data exchange diskette files 4-542
from special E-format diskettes 4-345
using BLDFILE procedure 4-56
using COPYDATA procedure 4-111
key length, specifying 4-57
key position, specifying 4-57
key sorting 4-252
indicators, alert, changing using SETALERT procedure 4-457
indicators, external
See switches
INFOMSG command 6-20

INFOMSG OCL statement 5-65
information, help 4-199
informational message statement 3-57
informational messages
preventing from displaying
using INFOMSG command 6-20
using INFOMSG statement 5-65
sending to display stations 3-57
sending to system console 3-59
$INIT utility program (prepares diskettes) A-52
INIT procedure 4-233
INITDIAG procedure 4-236
initial program load (IPL)
running procedures after 4-6
running procedures during 4-4
initializing diskettes
using INIT procedure 4-233
using the $INIT utility program A-52
initializing tapes
using TAPEINIT procedure 4-507
initiating
See also starting
preventing jobs from 6-48
INIT9332 procedure 4-237
INOUT utility control statement, $SFGR utility program A-108
input job queue
See job queue
INQUIRY conditional expression 3-39
inquiry display
preventing option 1 5-11
preventing options 2 and 3 5-11
INQUIRY procedure 4-238
insert tabs xvi
interrupting a job, preventing 5-11
introduction 1-1
OCL statements 1-2
utility programs 1-2
IPL (initial program load)
running procedures after 4-6
running procedures during 4-4
IPL procedure 4-240
ITF procedure 4-241
IWLOAD procedure 4-242
IWPTLOAD procedure 4-243
IWPTSAVE procedure 4-244
IWSAVE procedure 4-245

J

job directory 1-7
job entry, remote
displaying status of 6-40
running using MSRJE procedure 4-302
job files
creating
using BLDFILE procedure 4-56
using FILE OCL statement 5-36
description of 5-36
reserving space for 5-104
job processing priority, assigning
using ATTR OCL statement 5-11
using PRTY command 6-29
job queue
allowing jobs to run from 6-36
canceling jobs from 6-5
holding jobs on the queue 6-18
placing jobs on
using JOBQ command 6-21
using JOBQ OCL statement 5-66
releasing jobs from 6-30
starting a specific priority 6-38
status, displaying 6-40, 6-45
stopping 6-48
testing for 3-40
job step
date, changing
using DATE OCL statement 5-25
using DATE procedure 4-137
definition 2-1
example of 2-1
region size for 5-103
releasing from procedures 5-11
tracing using DEBUG OCL statement 5-28
job stream, running using JOBSTR procedure 4-246
JOBQ command 6-21
JOBQ conditional expression 3-40
JOBQ OCL statement 5-66
jobs
allowing communications sessions 6-36
allowing to run 6-36
allowing to run from job queue 6-36
allowing to start 6-36
canceling 6-5
canceling from within a procedure 3-60
canceling with a task dump 6-7
displaying status of currently running 6-40, 6-45
holding on the job queue 6-18
pausing during 3-69
placing on job queue
using JOBQ command 6-21
using JOBQ OCL statement 5-66

jobs *(continued)*
preventing from being canceled 5-11
preventing from being interrupted 5-11
preventing from starting 6-48
region size for
using REGION OCL statement 5-103
using SET procedure 4-454
releasing from job queue 6-30
running
after automatic history file copy 4-207
after IPL 4-6
during IPL 4-4
stopping 6-48
task control blocks, displaying 6-40
wait, causing a job to 5-118
JOBSTR procedure 4-246

K

key checking, duplicate, bypassing 5-40
key length, specifying
using BLDFILE procedure 4-57
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
key position, specifying
using BLDFILE procedure 4-57
using COPYDATA procedure 4-111
using RESTORE procedure 4-392
KEY utility control statement, $COPY utility program
copying files A-10
listing files A-14
keys
duplicate, allowing
using BLDFILE procedure 4-56
using FILE OCL statement 5-40
duplicate, specifying 4-57
listing using the $COPY utility program A-16
sorting using KEYSORT procedure 4-252
KEYS procedure 4-250
KEYSORT procedure 4-252
KEYSORT utility control statement, $DDST utility program A-39
keyword parameters 5-2

L

?L'position,length'? substitution expression 3-19
$LABEL utility program (lists file and library names) A-53
labels
See names
landscape printing
See rotating printed output
LANLOAD procedure 4-253
LANSAVE procedure 4-254
leaving out procedure parameters 5-99
leaving the system 5-81
length
key, specifying 4-57
parameter substitution expressions 3-15
parameters, determining for PROMPT OCL statement 5-99
level, automatic response severity, specifying 5-79
level, severity, specifying using RESPONSE procedure 4-374
levels, error reporting, for an IBM Token-Ring Network 4-549
levels, procedure 2-8
libraries
See also disk files, library members
allocating 4-9
attributes of library members 4-283
basic data exchange, transferring A-68
changing members, using DSU procedure 4-163
changing members, using SEU procedure 4-465
changing the size of 4-9
copying 4-62
basic data exchange format A-66
from a file using TOLIBR procedure 4-537
record mode format A-66
restoring using RESTLIBR procedure 4-386
to a file using FROMLIBR procedure 4-193
to another library using LIBRLIBR procedure 4-255
to diskette using SAVELIBR procedure 4-431
to tape using SAVELIBR procedure 4-431
creating 4-62
creating members
using DSU procedure 4-163
using SEU procedure 4-465
using the $MAINT utility program A-57
current, changing
using LIBRARY OCL statement 5-67
using MENU command 6-22
using SLIB procedure 4-471
current, substitution expression for 3-18
DATAF1 conditional expression 3-31
directory listing, description of 4-282
directory size
changing using ALOCLIBR procedure 4-9
changing using RESTLIBR procedure 4-386
specifying using BLDLIBR procedure 4-62
displaying
using LISTFILE procedure 4-267
using LISTLIBR procedure 4-272
using the $MAINT utility program A-75
existence testing 3-31

libraries *(continued)*
full, fixing, using CONDENSE procedure 4-109
gathering space in, using CONDENSE procedure 4-109
listing
source and procedure members using DSU procedure 4-163
using LISTFILE procedure 4-267
using LISTLIBR procedure 4-272
using the $MAINT utility program A-75
listing names
using CATALOG procedure 4-73
using the $LABEL utility program A-53
member existence testing
LOAD 3-42
PROC 3-44
SOURCE 3-46
SUBR 3-47
multinational character set conversion D-1
naming conventions 4-62
re-allocating 4-9
removing
source and procedure members using DSU procedure 4-163
removing members using REMOVE procedure 4-370
removing using DELETE procedure 4-144
renaming using RENAME procedure 4-372
restoring using RESTLIBR procedure 4-386
saving using SAVELIBR procedure 4-431
searching for phone list 5-21
session substitution expression 3-23
session, changing
using LIBRARY OCL statement 5-67
using SLIB procedure 4-471
sign-on, changing
using help support 4-201
using SET procedure 4-454
size of, specifying 4-62
library member name
changing
using CHNGEMEM procedure 4-94
library member reference number
changing
using CHNGEMEM procedure 4-94
library member subtype
changing
using CHNGEMEM procedure 4-94
library members
attributes of 4-283
attributes, description of 4-282
basic data exchange, transferring A-68
changing, using DSU procedure 4-163
changing, using SEU procedure 4-465
copying
basic data exchange format A-66
from a file using TOLIBR procedure 4-537
library members *(continued)*
copying *(continued)*
record mode format A-66
to a file using FROMLIBR procedure 4-193
to another library using LIBRLIBR procedure 4-255
using SAVELIBR procedure 4-431
creating
using DSU procedure 4-163
using SEU procedure 4-465
using the $MAINT utility program A-57
listing source and procedure members using DSU 4-163
removing source and procedure members using DSU 4-163
removing using REMOVE procedure 4-370
renaming using LIBRLIBR procedure 4-255
saving using SAVELIBR procedure 4-431
viewing source and procedure members using DSU 4-163
LIBRARY OCL statement 5-67
#LIBRARY
restoring using RESTLIBR procedure 4-386
saving using SAVELIBR procedure 4-431
LIBRLIBR procedure 4-255
line speed, communications
defining
using SETCOMM procedure 4-461
line, communications
assigning to a program 5-21
defining
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
dropping or holding after signoff
using OFF command 6-27
using OFF OCL statement 5-81
placing online or offline 5-116, 6-53
lines per inch, changing
See also font
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
lines per page, changing
See also font
using FORMS OCL statement 5-55
using LINES procedure 4-257
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SET procedure 4-454
LINES procedure 4-257
lining up, printer forms in printer 5-83
link-edit, programs, using OLINK procedure 4-325

linking
disk files using IDDUDISK procedure 4-223
files and definitions using IDDULINK procedure 4-224
list device, system
changing
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
substitution expression 3-23
list of system tasks 1-7
LIST procedure 4-259
list, phone
See phone lists
LISTARCH utility control statement, $TMSERV utility program
listing folder members A-117
LISTDATA procedure 4-261
LISTDONE conditional expression 3-41
LISTFILE procedure 4-267
listing
device error information C-16
disk files
using DFU (data file utility) 4-150
using INQUIRY procedure 4-238
using LIST procedure 4-259
using LISTDATA procedure 4-261
using LISTFILE procedure 4-267
using the $COPY utility program A-14
disk files using QRYRUN procedure 4-363
diskette files using LISTFILE procedure 4-267
display stations with messages using MSGFILE procedure 4-299
file names using CATALOG procedure 4-73
folder names using CATALOG procedure 4-73
history file using HISTORY procedure 4-209
libraries
using LISTFILE procedure 4-267
using LISTLIBR procedure 4-272
using the $MAINT utility program A-75
library names using CATALOG procedure 4-73
mail folders using OFCMAINT procedure 4-320
procedure statements, processed using DEBUG OCL statement 5-28
reports using QRYRUN procedure 4-363
resource security file using SECLIST procedure 4-445
source and procedure members using DSU 4-163
spool file entries using COPYPRT procedure 4-126
system data areas using DUMP procedure C-7
system list output
using PRINTER OCL statement 5-83
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
system measurement information 4-477
listing *(continued)*
tape files using LISTFILE procedure 4-267
user identification file using SECLIST procedure 4-445
users with messages using MSGFILE procedure 4-299
using LISTNRD procedure 4-292
LISTKEYS procedure example 2-13
LISTLIBR procedure 4-272
LISTNRD procedure 4-292
LOAD conditional expression 3-42
LOAD OCL statement 5-69
loading finance subsystem using LOAD3601 procedure 4-293
loading programs 5-69
LOADMBR utility control statement, $SFGR utility program A-108
LOAD3601 procedure 4-293
local data area
changing 5-70
clearing 5-71
OCL statement 5-70
substitution expression 3-19
local ID, changing 4-11
LOCAL OCL statement 5-70
local station switched line ID, specifying using ALTERCOM procedure 4-11
location, disk file, preferred 4-57
LOG OCL statement 5-72
LOG procedure 4-294
log, system service, adding entries to C-30
logging to history file
Help menu items
using LOG OCL statement 5-72
using LOG procedure 4-294
OCL statements
procedure attribute 2-10
using LOG OCL statement 5-72
using LOG procedure 4-294
logical display station ID 5-121
looping in procedures 3-67
lowercase to uppercase character translation 5-59
LPI
See lines per inch
LRTRLOAD procedure 4-295
LRTRSAVE procedure 4-296

M

?M'mic,position,length'? substitution expression 3-21
mail
displaying using OFCMAIL procedure 4-318
displaying using OFCMAINT procedure 4-320
main storage size, reserved for programs 5-103
$MAINT utility program
changing library size A-60
condensing library space A-63
copying
library members to files A-66
members from files A-72
copying job streams A-74
copying library members A-64
creating libraries A-56
creating library members A-57
listing library members A-75
from diskette or tape A-77
removing library members A-78
restoring libraries A-80
saving libraries A-79
maintaining an x.25 configuration 4-297
MAINTX25 procedure 4-297
making
See also creating
your own procedures 2-1
using SSP utility programs A-1
manual answer 4-13
manual call 4-13
manuals, related xiv
maximum
characters
JOBQ OCL statement 5-66
procedure continuation 2-7
file size 4-56
library directory size 4-62
library size 4-62
multiple requester terminals, testing for 3-43
record length 4-56
MCS
See multinational character set
MCSCONV procedure D-1
measuring the system's activity and performance 4-473
MEMBER OCL statement 5-73
members
See also library members
print belt, creating 5-61
?MENU? substitution expression 3-22
MENU command 6-22
menu name substitution expression 3-22
MENU OCL statement 5-75
menu options, help support, logging to history file 5-72
MENU utility control statement, $BMENU utility program A-7
menus
changing
using DSU procedure 4-163
using SDA procedure 4-437
using SEU procedure 4-465
creating
using BLDMENU procedure 4-66
using DSU procedure 4-163
using SDA procedure 4-437
using SEU procedure 4-465
displaying
using MENU command 6-22
using MENU OCL statement 5-75
restrictions, substitution expressions 3-10
sign-on, changing using help support 4-201
turning off 6-22
message file
defining size and location of using MSGFILE procedure 4-299
placing messages in 5-76, 6-24
removing messages from 4-299, 5-76, 6-24
message members
alert indicators, changing using SETALERT procedure 4-457
assigning to a program or procedure 5-73
automatic responses, changing using RESPONSE procedure 4-374
creating or changing using DSU procedure 4-163
creating or changing using SEU procedure 4-465
description of 4-132
example of 2-14
generating using CREATE procedure 4-132
ideographic considerations 4-132
severity level, changing using RESPONSE procedure 4-374
substituting from 3-21
messages
alert indicators, changing using SETALERT procedure 4-457
automatic responses, changing using RESPONSE procedure 4-374
defining message file using MSGFILE procedure 4-299
displayed from procedures, restriction 3-10
displaying
using ?R'mic? expression 3-14
using ERR procedure 4-176
using PAUSE statement 3-69
displaying those sent to you 5-76, 6-24
informational, displaying 3-57
listing display stations and users using MSGFILE procedure 4-299

messages *(continued)*
pausing and displaying 3-69
preventing from displaying informational
reasons why 2-24
using INFOMSG command 5-65, 6-20
prompting for parameters 3-14
removing from message file using MSGFILE
procedure 4-299
sending
using // * statement 3-57
using // ** statement 3-59
using MSG command 6-24
using MSG OCL statement 5-76
using OFCMSG procedure 4-321
severity level, changing using RESPONSE
procedure 4-374
subconsole, not responded to, displaying 6-40
substitution expression 3-21
system console, sending to 3-59
$MGBLD utility program (creates message members) A-82
MGBLD utility control statement, $MGBLD utility program A-82
$MMSP utility program (stops monitoring communications lines) A-83
$MMST utility program (starts monitoring communications lines) A-83
MODE command 6-23
mode, standby 6-23
modem clocking
changing using ALTERCOM procedure 4-11
defining using SETCOMM procedure 4-461
modify translation tables 4-93
move a folder from one disk location to another 4-298
MOVEFLDR procedure 4-298
MOVFLDR utility control statement, $TMSERV utility program A-119
MRT
See also multiple requester terminal
conditional expression 3-43
maximum
changing number assigned to a program 5-11
testing for 3-43
procedure attribute 2-10
MSG command 6-24
MSG OCL statement 5-76
MSGFILE procedure 4-299, 5-76, 6-24
MSRJE (multiple session remote job entry)
creating control table for 4-405
printing output from 4-405
status, displaying 6-40
MSRJE procedure 4-302
multinational character set
batch interface D-31
changed character table D-39
conversion D-1
disk files D-18
multinational character set *(continued)*
conversion D-1 *(continued)*
library members D-4
starting the conversion utility D-3
text folders D-23
disk file modification D-18
display formats D-15
hardware support D-2
library members
considerations D-4
copy back D-17
modification D-7
menu members D-15
message members D-15
procedure members D-14
RPG members D-13
sort members D-15
text folder modification D-23
WSU members D-16
multiple communications files, allowing 4-11
multiple requester terminals, number of, changing 5-11
multiple session remote job entry 4-302
creating control table for 4-405
printing output from 4-405
multipoint communications line 4-13

N

?n? substitution expression 3-11
?n'value'? substitution expression 3-11
names
creating unique 2-24
file, library, or folder, listing using CATALOG
procedure 4-73
library member, listing using LISTLIBR
procedure 4-272
naming
conventions
disk files 5-33
folders 4-530
library names 4-62
diskettes
using INIT procedure 4-233
using the $INIT utility program A-52
tapes
using TAPEINIT procedure 4-507
naming procedures 2-2
NEP (never-ending program) indicator, changing 5-11
nesting
procedures 2-8
substitution expressions 3-27

network resource directory
changing using EDITNRD procedure 4-167
copying
to diskette using SAVENRD procedure 4-434
to tape using SAVENRD procedure 4-434
creating using EDITNRD procedure 4-167
listing using LISTNRD procedure 4-292
restoring using RESTNRD procedure 4-389
saving using SAVENRD procedure 4-434
never-ending program indicator, changing 5-11
?nF'value'? substitution expression 3-12
NOAUTO advance, changing 5-9
nohalt level, specifying 5-79
NOHALT OCL statement 5-79
NOHALT procedure 4-305
nonswitched communications lines 4-13
NRZI, defining using SETCOMM procedure 4-461
?nT'value'? substitution expression 3-12
number of characters
JOBQ OCL statement 5-66
MSG OCL statement 5-76
parameter substitution 3-15
number of copies to print
changing using CHANGE command 6-9
changing using CHANGE OCL statement 5-18
specifying using PRINTER OCL statement 5-83
number of multiple requester terminals, changing 5-11

O

OCL statements 5-1
See also statements
ABEND 5-7
ALLOCATE 5-8
ATTR 5-11
called by SSP procedures 4-2
CANCEL 5-16
CHANGE 5-18
coding rules 5-4
COMM 5-21
comments 5-5
COMPILE 5-23
continuing 5-4
DATE 5-25
DEALLOC 5-27
DEBUG 5-28
EVOKE 5-30
FILE
for disk files 5-32
for diskette files 5-43
for tape files 5-48
FORMS 5-55
IMAGE 5-59
OCL statements 5-1 *(continued)*
INCLUDE 5-63
INFOMSG 5-65
introduction 1-2
JOBQ 5-66
LIBRARY 5-67
LOAD 5-69
LOCAL 5-70
LOG 5-72
logging to history file 5-72
MEMBER 5-73
MENU 5-75
MSG 5-76
NOHALT 5-79
OFF 5-81
POWER 5-82
PRINTER 5-83
PROMPT 5-97
REGION 5-103
RESERVE 5-104
RUN 5-105
SESSION 5-106
START 5-109
STOP 5-111
SWITCH 5-112
SYSLIST 5-113
VARY 5-116
WAIT 5-118
where used 3-54
WORKSTN 5-120
OFCBPRT procedure 4-307
OFCCAL procedure 4-309
OFCCANCL procedure 4-310
OFCCOMM procedure 4-310
OFCCONV procedure 4-311
OFCDATA procedure 4-312
OFCDFLT procedure 4-313
OFCDIR procedure 4-314
OFCFILE procedure 4-315
OFCGRP procedure 4-315
OFCINSTL procedure 4-316
OFCLDF procedure 4-316
OFCLOAD procedure 4-317
OFCMAIL procedure 4-318
OFCMAINT procedure 4-320
OFCMSG procedure 4-321
OFCQ procedure 4-322
OFCSAVE procedure 4-322
OFCSRCH procedure 4-323
OFCSTAT procedure 4-323
OFCUSER procedure 4-324
OFF command 6-27
OFF OCL statement 5-81

off, turning, the system 6-28
office directory
displaying using OFCDIR procedure 4-314
offline, placing devices 5-116, 6-53
OLINK procedure 4-325
OLPDLOAD procedure 4-328
OLPDSAVE procedure 4-329
omitting procedure parameters 5-99
online information
displaying using READINFO procedure 4-368
online, placing devices 5-116, 6-53
operation control language statements
See OCL statements
operator control commands
See control commands
operators, sending messages to 5-76
optional parameters
assigning a temporary value to 3-12
assigning a value to 3-11
forcing a value into 3-12
prompting for 3-13, 3-14
options for an SMF report file 4-474
options, help support menus, logging to history file 5-72
order of printing, using forms number to control 6-37
ORGANIZE procedure 4-329
other manuals xiv
output, system list 5-113
overlay linkage editor
OLINK procedure 4-325
OVERRIDE procedure 4-329
overriding
See changing
overriding BSC parameters using ALTERCOM procedure 4-11

P

$PACK utility program (compresses disk space) A-84
pack ID
See volume ID
packed keys, listing using the $COPY utility program A-16
pages, separator, changing number of 6-9
paper, aligning in printer 5-83
parameters
adding values 3-60
assigning values 3-11, 3-60
calculating 3-60
characters allowed in 5-64
coding considerations 2-6
description of 2-3
determining the length of 5-99
evaluating substitution expressions 3-60
introduction to 2-3
parameters *(continued)*
keyword 5-2
length of, specifying 5-99
length substitution 3-15
maximum number of 2-3
OCL 5-2
optional, assigning values to 3-11
passing all to another procedure 5-64
positional 2-3
procedure 5-3
procedure attribute 2-10
prompting for 5-97
required, assigning values to 3-13
setting the return code 3-60
size substitution 3-15
skipping 5-99
substitution expressions 3-10
subtracting values 3-60
testing 2-6
testing value of 3-50, 3-52
parameters, required, assigning values to 3-14
parentheses in syntax formats 1-3
PASSTHRU procedure 4-330
PASSWORD procedure 4-331
password security, installing or removing using SECDEF procedure 4-441
passwords
changing using SECEDIT procedure 4-443
listing using SECLIST procedure 4-445
restoring using SECLIST procedure 4-448
saving using SECSAVE procedure 4-451
PATCH procedure C-17
PAUSE statement 3-69
pausing during procedure processing 3-69
PCEXCH procedure 4-332
PCEXEC procedure 4-334
PCOLOAD procedure 4-335
PCOPROF procedure 4-336
PCOSAVE procedure 4-337
PCU procedure 4-338
PDATA (program data), specifying for prompts 5-99
performance
system, measuring 4-473
tips, procedures 2-22
period, retention, diskette files 5-45
period, retention, tape files 5-51
permanent disk files
See resident files
permanent diskette files 5-45
permanent tape files 5-51
personal computers, sending and receiving messages 6-24
Personal Services/36
changing defaults 4-313
controlling activity and communications queues 4-322
creating or changing a calendar 4-309

Personal Services/36 *(continued)*
displaying mail log entries 4-318
displaying online information for using READINFO procedure 4-368
displaying the directory 4-314
enrolling or changing enrollment of office users/ 4-324
installing office files 4-316
maintaining communications queues and routes 4-320
maintaining mail folders 4-320
saving files and folders 4-312
sending mail 4-318
sending messages to a group 4-321
working with user groups 4-315
phone lists
assigning to a program 5-21
creating using DEFINEPN procedure 4-140
creating using DEFINX21 procedure 4-141
restoring 5-21
searching library for 5-21
testing for completion using LISTDONE conditional expression 3-41
placing jobs on job queue 6-21
placing messages in message file 5-76, 6-24
$PNLM utility program (defines phone lists) A-84
point of sale terminal, transferring information using POST procedure 4-345
polling characters, BSC 4-17
position
key, specifying 4-57
of output in spool file, changing 6-9
positional parameters 2-3
$POST utility program
copying 5260 data files A-85
listing diskette files A-86
POST procedure 4-345
POWER command 6-28
POWER OCL statement 5-82
powering off the system 5-82, 6-28
$PRCED utility program (edits user ID file) A-87
$PRCLT utility program (lists user ID file) A-87
precautions, #STRTUP1 procedure 4-4
preparing diskettes
using INIT procedure 4-233
using the $INIT utility program A-52
preparing tapes
using TAPEINIT procedure 4-507
prerequisite manuals xiv
preventing
See also stopping
canceling of a job 5-11
displaying of informational messages
using expressions 2-24
using INFOMSG command 6-20
using INFOMSG OCL statement 5-65
preventing *(continued)*
interrupting of a job 5-11
job queue from processing 6-48
jobs from being run from the job queue 6-18
jobs from starting 6-48
print entries from being printed 6-18
print belts
changing
using IMAGE OCL statement 5-59
using SET procedure 4-454
members, creating 5-61
3262 Printer, chart of E-1
print density, horizontal or vertical, changing
See also font
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
Print key
border or heading, specifying
using PRINTKEY procedure 4-354
using SET procedure 4-454
using WORKSTN OCL statement 5-120
printer ID, specifying
using PRINT procedure 4-350
using PRINTKEY procedure 4-354
using SET procedure 4-454
using WORKSTN OCL statement 5-120
PRINT procedure 4-350
print queue
See spool file
printed output
balancing among printers 4-34
canceling
using CANCEL command 6-5
using CANCEL OCL statement 5-16
changing order of printing 6-9
continuing 5-83
controlling
using BALPRINT procedure 4-34
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SET procedure 4-454
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
copies to print, changing
using CHANGE command 6-9
using CHANGE OCL statement 5-18
forms number
causing to print first using START command 6-37
changing using CHANGE command 6-9
changing using CHANGE OCL statement 5-18
holding in spool file
using HOLD command 6-18
using PRINTER OCL statement 5-83

printed output *(continued)*
printed ID to be used, changing
using CHANGE OCL statement 5-18
printer ID to be used, changing
using CHANGE command 6-9
printer paper drawer to be used, changing
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
reduction of, changing
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
releasing for printing 6-30
responding to subconsole messages 6-40
restarting the printing of 6-34
rotation of, changing
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
separator pages, changing number of 6-9
spool writer priority, changing 6-9
starting the printing of
using START command 6-36
using START OCL statement 5-109
status, displaying 6-40, 6-45
stopping
using STOP command 6-48
using STOP OCL statement 5-111
system list 5-113
printer
drawer, changing
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
group identification number 4-34
load balancing 4-34
specifying for Print key output using PRINTKEY procedure 4-354
specifying for printed output
using PRINT procedure 4-350
using SET procedure 4-454
specifying for task dump 5-7
printer ID
specifying
using BALPRINT procedure 4-34
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using PRINTKEY procedure 4-354
using SET procedure 4-454
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
substitution expressions
session printer 3-22
system list device 3-23
PRINTER OCL statement 5-83
?PRINTER? substitution expression 3-22
printers
aligning the paper in 5-83
allowing to print before output is complete 5-83
assigning for all printer output 5-55
assigning system list device
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
balancing printed output among 4-34
belt image members, creating 5-61
canceling entries to be printed
using CANCEL command 6-5
using CANCEL OCL statement 5-16
changing the one to use
using CHANGE command 6-9
using CHANGE OCL statement 5-18
continuing 5-83
copies to print, changing
using CHANGE command 6-9
using CHANGE OCL statement 5-18
default, changing using SET procedure 4-454
exchanging 6-2
forms number, changing
using CHANGE command 6-9
using CHANGE OCL statement 5-18
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
using SET procedure 4-454
forms number, printing all forms with the same 6-37
holding entries on the spool file 6-18
ID to be used, changing
using CHANGE command 6-9
using CHANGE OCL statement 5-18
online or offline, placing 5-116, 6-53
position of output in spool file, changing 6-9
print belts for 3262 Printer E-1
Print key, changing
using PRINT procedure 4-350
using PRINTKEY procedure 4-354
using SET procedure 4-454
using WORKSTN OCL statement 5-120
priority of printed output 5-83
procedure, example 2-1
releasing held output 6-30
responding to subconsole messages 6-40
restarting printed output 6-34
separator pages, changing number of 6-9
sign-on, changing using SET procedure 4-454
specifying as a group 4-34
spool writer priority, changing 6-9
spooling 5-83
starting
using START command 6-36
using START OCL statement 5-109

printers *(continued)*
status, displaying 6-40, 6-45
stopping printing of output
using STOP command 6-48
using STOP OCL statement 5-111
system list output 5-113
system, assigning a different printer as 6-2
translation tables for 3262 printer E-7
printing
See also listing
balancing load among a group of printers 4-34
data definitions using IDDUPRT procedure 4-226
documents using TEXTDOC procedure 4-512
forms number, changing order of 6-37
lowercase characters as uppercase characters 5-59
reports using QRYRUN procedure 4-363
spool file entries
using BALPRINT procedure 4-34
using COPYPRT procedure 4-126
printing volume statistical logs
using TAPESTAT procedure 4-509
PRINTKEY procedure 4-354
priority
job queue, starting a specific 6-38
printed output 5-83
processing, specifying
using ATTR OCL statement 5-11
using PRTY command 6-29
processing, spool writer, changing 6-9
$PRLST utility program (lists user ID file) A-88
PROBLEM
procedure 4-356
problem determination, starting online 4-369
problem management
ALERT procedure 4-7
SETALERT procedure 4-457
PROBLEM procedure 4-369
problems
tape, identifying using TAPESTAT 4-509
?PROC? substitution expression 3-23
PROC conditional expression 3-44
procedure formats, description of 1-3
procedure parameters 5-3
procedures
See also control commands
#STRTUP1 4-4
#STRTUP2 4-6
ALERT 4-7
allowed statements in 2-1
allowing to run 6-36
ALOCFLDR 4-8
ALOCLIBR 4-9
ALTERBSC 4-10
ALTERCOM 4-11
ALTERSDL 4-18
APAR C-2
procedures *(continued)*
APPNINFO 4-19
ARCHIVE 4-22
ASM 4-27
ASMLOAD 4-29
ASMSAVE 4-30
attributes
data 2-10
MRT 2-10
OCL statement logging 2-10
parameters 2-10
AUTO 4-30
AUTOC 4-31
BACKUP 4-34
BALPRINT 4-34
BASIC 4-39
BASICP 4-40
BASICR 4-41
BASICS 4-42
BASLOAD 4-44
BASSAVE 4-45
BGUATTR 4-46
BGUCHART 4-47
BGUDATA 4-50
BGUGRAPH 4-51
BGULOAD 4-54
BGUSAVE 4-55
BLDFILE 4-56
BLDINDEX 4-59
BLDLIBR 4-62
BLDMENU 4-66
branching 3-67
BUILD 4-70
CACHE 4-72
calculating parameter values 3-60
canceling 3-60, 6-5
CATALOG 4-73
CGU 4-91
CGULOAD 4-92
CGUSAVE 4-93
changing using DSU procedure 4-163
changing using SEU procedure 4-465
chart of SSP utilities and OCL statements called by 4-2
CHGXLATE 4-93
CHNGEMEM 4-94
CNFIGICF 4-96
CNFIGSSP 4-97
CNFIGX25 4-97
COBLOAD 4-98
COBOL 4-98
COBOLC 4-99
COBOLCG 4-101
COBOLG 4-102
COBOLONL 4-102
COBOLP 4-102

procedures *(continued)*
COBSAVE 4-103
COBSDA 4-103
COBSEU 4-104
coding tips 2-22, 2-23
command, definition 1-1
COMPRESS 4-105
CONDENSE 4-109
conditional processing 3-28
contents of 2-1
continuation 2-7
control expression guide 3-2
control expressions 3-1
using effectively 2-23
COPYDATA 4-111
COPYDIAG 4-120
COPYI1 4-121
COPYPRT 4-126
CREATE 4-132
creating 2-1
using DSU procedure 4-163
using SEU procedure 4-465
using the $MAINT utility program A-57
DATE 4-137
debugging 2-24
using DEBUG OCL statement 5-28
using LOG procedure 4-294
DEFINEID 4-139
DEFINEPN 4-140
definition of 1-1, 2-1
DEFINLOC 4-140
DEFINX21 4-141
DEFINX25 4-142
DEFSUBD 4-143
DELETE 4-144
DFA C-5
DFU 4-150
DFULOAD 4-151
DFUSAVE 4-152
DICTLOAD 4-153
DICTSAVE 4-155
DISABLE 4-157
diskette files, allocating 5-43
DISPLAY 4-158
display stations, allocating 5-120
DLSLOAD 4-159
DLSSAVE 4-159
DOCCNV 4-160
DOCPLOAD 4-161
DOCPSAVE 4-162
DSU 4-163
DSULOAD 4-166
DSUSAVE 4-167
DUMP C-7
EDITNRD 4-167
EM3270 4-168

procedures *(continued)*
ENABLE 4-170
ending 3-60
ENTER 4-172
entering into the system 2-2
EPDOWNL 4-174
EPLMRG 4-174
EP3270 4-175
ERAP C-16
ERR 4-176
ES3270 4-178
evoking using EVOKE OCL statement 5-30
examples of 2-4, 2-11
expression guide 3-2
EXTRACT 4-180
files, how to specify 5-32
FORMAT 4-181
FORTC 4-184
FORTCG 4-184
FORTG 4-184
FORTGO 4-185
FORTLOAD 4-186
FORTONL 4-187
FORTP 4-187
FORTRANC 4-188
FORTSAVE 4-191
FORTSDA 4-191
FORTSEU 4-192
FROMLIBR 4-193
HELP 4-199
HISTCOPY 4-207
HISTCRT 4-206
HISTORY 4-209
holding on the job queue 6-18
how to create 2-1
ICFDEBUG 4-221
ICVERIFY 4-222
IDDUDCT 4-222
IDDUDFN 4-223
IDDUDISK 4-223
IDDULINK 4-224
IDDUPRT 4-226
IDDURBLD 4-227
IDDUXLAT 4-227
INIT 4-233
INITDIAG 4-236
INIT9332 4-237
INQUIRY 4-238
introduction to 1-1
IPL 4-240
ITF 4-241
IWLOAD 4-242
IWPTLOAD 4-243
IWPTSAVE 4-244
IWSAVE 4-245
job queue, placing on 6-21

procedures *(continued)*
JOBSTR 4-246
KEYS 4-250
KEYSORT 4-252
LANLOAD 4-253
LANSAVE 4-254
length of parameters, determining 5-99
levels 2-8
LIBRLIBR 4-255
LINES 4-257
LIST 4-259
LISTDATA 4-261
LISTFILE 4-267
listing statements processed using DEBUG OCL statement 5-28
LISTLIBR 4-272
LISTNRD 4-292
LOAD3601 4-293
LOG 4-294
looping 3-67
LRTRLOAD 4-295
LRTRSAVE 4-296
MAINTx25 4-297
making from utility programs A-1
making your own 2-1
maximum number of parameters 2-3
MCSCONV D-1
message members, allocating 5-73
MOVEFLDR 4-298
MSGFILE 4-299, 5-76, 6-24
MSRJE 4-302
name, substitution expression 3-23
naming 2-2
nesting 2-8
NOHALT 4-305
OFCBPRT 4-307
OFCCAL 4-309
OFCCANCL 4-310
OFCCOMM 4-310
OFCCONV 4-311
OFCDATA 4-312
OFCDFLT 4-313
OFCDIR 4-314
OFCFILE 4-315
OFCGRP 4-315
OFCINSTL 4-316
OFCLDF 4-316
OFCLOAD 4-317
OFCMAIL 4-318
OFCMAINT 4-320
OFCMSG 4-321
OFCQ 4-322
OFCSAVE 4-322
OFCSRCH 4-323
OFCSTAT 4-323
OFCUSER 4-324

procedures *(continued)*
OLINK 4-325
OLPDLOAD 4-328
OLPDSAVE 4-329
ORGANIZE 4-329
OVERRIDE 4-329
parameter data 5-64
parameter length, specifying 5-99
parameters 2-3
  defaults 2-5
  maximum number of 2-3
  testing for correct entries 3-28
parameters, passing all parameters 5-64
PASSTHRU 4-330
PASSWORD 4-331
PATCH C-17
pausing during 3-69
PCEXCH 4-332
PCEXEC 4-334
PCOLOAD 4-335
PCOPROF 4-336
PCOSAVE 4-337
PCU 4-338
performance tips 2-22
POST 4-345
preventing from starting 6-48
PRINT 4-350
PRINTKEY 4-354
PROBLEM 4-356, 4-369
PROFLOAD 4-357
PROFSAVE 4-358
program data, specifying 5-99
PRTGRAPH 4-359
PTF C-18
QRY 4-360
QRYDE 4-361
QRYLOAD 4-362
QRYRUN 4-363
QRYSAVE 4-367
READINFO 4-368
REBLD 4-369
releasing from job queue 6-30
RELOAD 4-369
REMOVE 4-370
RENAME 4-372
REQUESTX 4-373
resetting 3-70
RESPONSE 4-374
RESTEXTN 4-380
RESTFLDR 4-383
RESTLIBR 4-386
RESTNRD 4-389
RESTORE 4-392
restrictions 3-29
RETRIEVE 4-401
return code 3-16

procedures *(continued)*
returning from 3-71
RJFILE 4-405
RJTABLE 4-405
ROLLKEYS 4-406
RPG 4-406
RPGC 4-407
RPGLOAD 4-410
RPGONL 4-411
RPGP 4-411
RPGR 4-412
RPGSAVE 4-413
RPGSDA 4-414
RPGSEU 4-414
RPGX 4-415
running 5-63
after automatic history file copy 4-207
after IPL 4-6
during IPL 4-4
SAVE 4-416
SAVEEXTN 4-425
SAVEFLDR 4-428
SAVELIBR 4-431
SAVENRD 4-434
SDA 4-437
SDALOAD 4-439
SDASAVE 4-440
SECDEF 4-441
SECEDIT 4-443
SECLIST 4-445
SECREST 4-448
SECSAVE 4-451
SERVICE C-29
service aid C-1
SERVLOG C-30
SET 4-454
SETALERT 4-457
SETCOMM 4-461
SETDUMP C-31
SEU (source entry utility) 4-465
SEULOAD 4-467
SEUSAVE 4-468
SHRFLOAD 4-469
SHRFSAVE 4-470
skipping parameters 5-99
SLIB 4-471
SMF 4-473
SMFDATA 4-474
SMFPRINT 4-477
SMFSTART 4-480
SMFSTOP 4-482
SORT 4-483
SPECIFY 4-483
SRTX 4-484
SRTXBLD 4-485
SRTXLOAD 4-486

procedures *(continued)*
SRTXSAVE 4-488
STARTM 4-489
statements in 2-1
STATEST 4-490
STOPGRP 4-491
STOPM 4-492
stopping 6-48
stopping after each job step 5-29
STRTGRP 4-493
substitution expressions 2-5, 2-24
SWDLOAD 4-494
SWDSAVE 4-495
SWITCH 4-496
SYSLIST 4-498
system 4-1
tape files, allocating 5-48
TAPECOPY 4-500
TAPEINIT 4-507
TAPESTAT 4-509
testing for running 3-29
TEXTCONV 4-511
TEXTDCT 4-511
TEXTDOC 4-512
TEXTFLDR 4-530
TEXTLOAD 4-531
TEXTOBJ 4-532
TEXTPROF 4-532
TEXTPRTQ 4-533
TEXTREL 4-534
TEXTSAVE 4-536
TOLIBR 4-537
TRACE C-32
tracing 5-29
TRANSFER 4-542
TRNMGR 4-549
troubleshooting 2-24
UPDATE 4-550
wait, causing to 5-118
where used 3-54
WSFLOAD 4-552
WSFSAVE 4-553
WSU 4-554
WSULOAD 4-557
WSUSAVE 4-558
WSUTXCR 4-559
WSUTXEX 4-561
WSUTXRV 4-563
XREST 4-563
XSAVE 4-563
processing priority, assigning
using ATTR OCL statement 5-11
using PRTY command 6-29
PROFLOAD procedure 4-357

PROFSAVE procedure 4-358
program check, specifying device for task dump 5-7
program data, specifying for prompts 5-99
program date, changing
using DATE OCL statement 5-25
using DATE procedure 4-137
program temporary fixes C-18
programs
allowing communications sessions 6-36
allowing to run 6-36
allowing to start 6-36
Assembler, compiling 4-27
BASIC
converting using BASICS procedure 4-42
creating or changing using BASIC procedure 4-39
cross-referencing using BASICS procedure 4-42
running using BASIC procedure 4-39
running using BASICP procedure 4-40
running using BASICR procedure 4-41
canceling 6-5
changing
using DSU procedure 4-163
using SEU procedure 4-465
COBOL
compiling using COBOLC procedure 4-99
creating or changing display formats using COBSDA procedure 4-103
creating or changing using COBOLONL procedure 4-102
creating or changing using COBSEU procedure 4-104
developing using COBOLONL procedure 4-102
communications line, assigning 5-21
creating
using DSU procedure 4-163
using SEU procedure 4-465
debugging
using SETDUMP procedure C-31
diskette files, allocating 5-43
display stations, allocating 5-120
files, how to allocate 5-32
FORTRAN
compiling using FORTRANC procedure 4-188
creating or changing display formats using FORTSDA procedure 4-191
creating or changing using FORTONL procedure 4-187
creating or changing using FORTSEU procedure 4-192
developing using FORTONL procedure 4-187
displaying menu of procedures 4-187
holding on the job queue 6-18
job queue, placing on 6-21
link-edit using OLINK procedure 4-325
loading and running 5-69
programs *(continued)*
message members, allocating 5-73
never-ending, indicator, changing 5-11
phone list, assigning 5-21
preventing from starting 6-48
Print key printer for, specifying 5-120
releasing from job queue 6-30
restoring display stations after 5-120
return code 3-16
RPG
compiling using RPGC procedure 4-407
creating or changing display formats using RPGSDA procedure 4-414
creating or changing using RPGONL procedure 4-411
creating or changing using RPGSEU procedure 4-414
cross-referencing using RPGX procedure 4-415
developing using RPGONL procedure 4-411
displaying menu of procedures using RPGP procedure 4-411
running using RPGR procedure 4-412
running 5-105
status of currently running 6-40
stopping 6-48
tape files, allocating 5-48
task control blocks, displaying 6-40
utility, SSP
introduction to 1-2
using to make procedures A-1
wait, causing to 5-118
programs, utility
See utility programs
prompt displays, using switches to modify 5-99
PROMPT OCL statement 5-97
prompting for parameters 5-97
prompting for procedure data 5-97
$PRPWD utility program (changes user password) A-88
PRTFILE Files, File Format for 4-516
PRTGRAPH procedure 4-359
PRTY command 6-29
$PRUED utility program (edits user ID file) A-89
$PRUID utility program (password and badge security) A-90
$PRURS utility program (restores user ID file) A-91
$PRUSV utility program (saves user ID file) A-93
PTF (program temporary fixes) C-18
PTF procedure C-18
public data network
See X.21 public data network

Q

QRY procedure 4-360
QRYDE procedure 4-361
QRYLOAD procedure 4-362
QRYRUN procedure 4-363
QRYSAVE procedure 4-367
queries
changing using QRY procedure 4-360
creating using QRY procedure 4-360
running using QRYRUN procedure 4-363
Query/36
changing queries 4-360
creating queries 4-360
creating reports 4-360
displaying online information for using READINFO procedure 4-368
displaying or printing reports 4-363
running queries 4-363
queue
See job queue, spool file

R

?R? substitution expression 3-13
?R'mic'? substitution expression 3-14
re-allocating libraries 4-9
READINFO procedure 4-368
REBLD procedure 4-369
receiving messages 5-76, 6-24
record blocking, changing 5-40
record mode format, library members A-66
record separator, changing 4-11
records
converting to blocks B-3
deleted, removing using COPYDATA procedure 4-111
file size, maximum 4-56
to blocks conversion B-3
recovering
disk file data using BUILD procedure 4-70
WSU transaction file using WSUTXRV procedure 4-563
reducing size of printed output 4-350
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
reference, cross
auto report RPG program using AUTOC procedure 4-32
BASIC program using BASICS procedure 4-42
reference, cross *(continued)*
RPG programs
using RPGC procedure 4-408
using RPGX procedure 4-415
REGION OCL statement 5-103
region size, changing
using REGION OCL statement 5-103
using SET procedure 4-454
related manuals xiv
RELEASE command 6-30
releasing
display stations from procedures 5-11
documents for printing using TEXTREL procedure 4-534
job steps from procedures 5-11
jobs from job queue 6-30
printed output 6-30
RELOAD procedure 4-369
reloading system
See RESTLIBR procedure
remarks, OCL statements 5-5
remote IDs
changing using ALTERCOM procedure 4-11
defining using DEFINEID procedure 4-139
remote job entry
creating control table for using RJTABLE procedure 4-405
printing output from using RJFILE procedure 4-405
running using MSRJE procedure 4-302
status, displaying 6-40
remote work stations
dropping the communications line after signoff 5-81, 6-27
holding the communications line after signoff 5-81, 6-27
testing using STATEST procedure 4-490
REMOVE procedure 4-370
REMOVE utility control statement, $DELET utility program A-40
removing
badge security using SECDEF procedure 4-441
deleted records from disk files
using COPYDATA procedure 4-111
using SAVE procedure 4-416
disk files using DELETE procedure 4-144
display formats
using FORMAT procedure 4-181
using the $SFGR utility program A-108
folders using DELETE procedure 4-144
libraries using DELETE procedure 4-144
library members
using DSU procedure 4-163
using REMOVE procedure 4-370
using the $MAINT utility program A-78
members from folders using ARCHIVE procedure 4-22

removing *(continued)*
messages from message file 5-76, 6-24
messages from message file using MSGFILE procedure 4-299
password security using SECDEF procedure 4-441
resource security file using SECDEF procedure 4-441
resource security using SECDEF procedure 4-441
user identification file using SECDEF procedure 4-441
$RENAM utility program (renames disk files, libraries, and folders) A-94
RENAME procedure 4-372
RENAME utility control statement, $RENAM utility program A-94
renaming
disk files using RENAME procedure 4-372
diskettes
using INIT procedure 4-233
using the $INIT utility program A-52
documents using TEXTDOC procedure 4-512
folders using RENAME procedure 4-372
libraries using RENAME procedure 4-372
library members
using LIBRLIBR procedure 4-255
using the $MAINT utility program A-64
tapes
using TAPEINIT procedure 4-507
reorganizing
folders using ALOCFLDR procedure 4-8
reorganizing disk files
using COPYDATA procedure 4-111
using SAVE procedure 4-416
REPLY command 6-32
replying to subconsole messages from system console 6-40
replying to system messages 6-32
reports
creating using QRY procedure 4-360
displaying or printing using QRYRUN procedure 4-363
requesters, multiple terminal, assigning to a program 5-11
requesting an X.21 facility using REQUESTX procedure 4-373
REQUESTX procedure 4-373
required parameters, prompting for 3-13, 3-14
REQX utility control statement, #GCFR utility program A-127
RESERVE OCL statement 5-104
reserving
files for programs 5-32
main storage size for programs 5-103
space for job and scratch files 5-104
RESET statement 3-70
resetting
display stations after programs 5-120
history file using HISTORY procedure 4-209
phone lists 5-21
procedures 3-70
resident files
creating
using BLDFILE procedure 4-56
using FILE OCL statement 5-36
description of 5-36
resource security file
changing using SECEDIT procedure 4-443
creating using SECDEF procedure 4-441
listing using SECLIST procedure 4-445
removing using SECDEF procedure 4-441
restoring using SECREST procedure 4-448
saving using SECSAVE procedure 4-451
resource security, starting or stopping 4-441
RESPONSE procedure 4-374
RESPONSE utility control statement, $ARSP utility program A-3
response, automatic severity level, specifying
using NOHALT OCL statement 5-79
using NOHALT procedure 4-305
response, automatic, specifying using RESPONSE procedure 4-374
RESTART command 6-34
restarting
printed output 6-34
system activity 6-36
RESTEXTN procedure 4-380
RESTFLDR procedure 4-383
RESTFLDR utility control statement, $TMSERV utility program
restoring folders A-121
RESTLIBR procedure 4-386
RESTNRD procedure 4-389
RESTORE procedure 4-392
restoring
disk files using RESTORE procedure 4-392
display stations after programs 5-120
folder members to disk using RETRIEVE procedure 4-401
folders using RESTFLDR procedure 4-383
libraries using RESTLIBR procedure 4-386
members to folders using RETRIEVE procedure 4-401
network resource directory using RESTNRD procedure 4-389
resource security file using SECREST procedure 4-448
user identification file using SECREST procedure 4-448
WSU transaction file using WSUTXRV procedure 4-563

restoring the extended character file 4-380
restrictions
ACTIVE conditional expression 3-29
BLOCKS conditional expression 3-30
display formats 3-10
displaying messages 3-10
ELSE expressions 3-55
menus 3-10
parameters, maximum number of 3-11
substitution expressions 3-10
system console 3-10
resuming
See restarting, starting
retention days, diskette FILE OCL statement 5-45
retention days, tape FILE OCL statement 5-51
RETRIEVE procedure 4-401
RETRIEVE utility control statement, $TMSERV utility program
restoring folder members A-120
retry count, error
changing using ALTERCOM procedure 4-11
defining using SETCOMM procedure 4-461
return code (?CD?) substitution expression 3-16
return code, setting 3-60
RETURN statement 3-71
returning from procedures 3-71
RJFILE procedure 4-405
RJTABLE procedure 4-405
roll keys, set direction of 4-406
ROLLKEYS procedure 4-406
RORGFLDR utility control statement, $TMSERV utility program
reorganizing folders A-118
rotating printed output
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
RPG
creating or changing using DSU procedure 4-163
RPG procedure 4-406
RPG programs
compiling
using AUTOC procedure 4-31
using RPGC procedure 4-407
creating or changing display formats using RPGSDA procedure 4-414
creating or changing using DSU procedure 4-163
creating or changing using RPGSEU procedure 4-414
creating using SDA procedure 4-437
cross-referencing 4-408
cross-referencing using RPGX procedure 4-415
developing using RPGONL procedure 4-411
displaying menu of procedures using RPGP procedure 4-411
running using RPGR procedure 4-412
RPGC procedure 4-407
RPGLOAD procedure 4-410
RPGONL procedure 4-411
RPGP procedure 4-411
RPGR procedure 4-412
RPGSAVE procedure 4-413
RPGSDA procedure 4-414
RPGSEU procedure 4-414
RPGX procedure 4-415
$RREDT utility program (edits resource security file) A-95
$RRESC utility program (resource security) A-96
$RRLST utility program (lists resource security file) A-97
$RRSAV utility program (saves resource security file) A-98
$RRSTR utility program (restores resource security file) A-99
$RRTED utility program (edits resource security file) A-101
$RRTLT utility program (lists resource security file) A-101
rules
See directions
RUN OCL statement 5-105
running
BASIC procedures 4-40
BASIC programs 4-39, 4-41
FORTRAN programs 4-185, 4-187
jobs
after automatic history file copy 4-207
from job queue 5-66, 6-21
using EVOKE OCL statement 5-30
using INCLUDE OCL statement 5-63
jobs, allowing 6-36
preventing jobs from 6-48
procedures
after automatic history file copy 4-207
after IPL 4-6
during IPL 4-4
using INCLUDE OCL statement 5-63
programs
using LOAD OCL statement 5-69
using RUN OCL statement 5-105
queries using QRYRUN procedure 4-363
RPG programs using RPGR procedure 4-412
status of jobs 6-40, 6-45

S

SAVE procedure 4-416
SAVEEXTN procedure 4-425
SAVEFLDR procedure 4-428
SAVEFLDR utility control statement, $TMSERV utility program
copying folders A-116
SAVELIBR procedure 4-431
SAVENRD procedure 4-434
saving
disk files in compressed format using SAVE procedure 4-416
disk files using OFCDATA procedure 4-312
disk files using SAVE procedure 4-416
folder members using ARCHIVE procedure 4-22
folders using OFCDATA procedure 4-312
folders using SAVEFLDR procedure 4-428
history file using HISTORY procedure 4-209
libraries using SAVELIBR procedure 4-431
network resource directory using SAVENRD procedure 4-434
resource security file using SECSAVE procedure 4-451
spool file entries using COPYPRT procedure 4-126
user identification file using SECSAVE procedure 4-451
saving the extended character file 4-425
scratch files
creating
using BLDFILE procedure 4-56
using FILE OCL statement 5-36
description of 5-36
reserving space for 5-104
SCRATCH utility control statement, $DELET utility program A-40
screen formats
See display formats
SCRNPRT procedure example 2-16
SDA procedure 4-437
SDALOAD procedure 4-439
SDASAVE procedure 4-440
SDLC
retry count, changing using ALTERCOM procedure 4-11
retry count, defining using SETCOMM procedure 4-461
testing remote work stations using STATEST procedure 4-490
time-out value, changing using ALTERCOM procedure 4-11
time-out value, defining using SETCOMM procedure 4-461
SECDEF procedure 4-441
SECEDIT procedure 4-443
SECLIST procedure 4-445
second-level message members
assigning to programs and procedures 5-73
creating using CREATE procedure 4-132
secondary backup line, allowing 4-11
SECREST procedure 4-448
SECSAVE procedure 4-451
sector mode format, library members A-66
sectors
changing, library directory size 4-9
converting to blocks B-3
security
badge, allowing using SECDEF procedure 4-441
classification testing 3-45
existence testing 3-45
password, allowing using SECDEF procedure 4-441
resource, allowing using SECDEF procedure 4-441
SECURITY conditional expression 3-45
security procedures
SECDEF (defines system security) 4-441
SECEDIT (changes information in security files) 4-443
SECLIST (lists information in security files) 4-445
SECREST (restores security files) 4-448
SECSAVE (saves security files) 4-451
SELECT utility control statement, $COPY utility program
copying disk files A-10
listing files A-14
saving files A-24
sending mail using OFCMAIL procedure 4-318
sending messages
using // * statement 3-57
using // ** statement 3-59
using MSG command 6-24
using MSG OCL statement 5-76
separator characters, autocall, allowing 4-461
separator pages, changing number of 6-9
separator, record, changing 4-11
sequential disk files, creating, using BLDFILE procedure 4-56
series of diskette files, processing 5-9
service aid procedures C-1
APAR C-2
DFA C-5
DUMP C-7
ERAP C-16
PATCH C-17
PTF C-18
SERVICE C-29
SERVLOG C-30
SETDUMP C-31
TRACE C-32

service display station, system
allowing to be used 6-36
stopping from being used 6-48
service log
adding entries to C-30
listing using DUMP procedure C-7
SERVICE procedure C-29
servicing the system using the SERVICE procedure C-29
SERVLOG procedure C-30
session
canceling other display stations by signing them off 6-5
date, changing
using DATE OCL statement 5-25
using DATE procedure 4-137
display station status, displaying 6-40, 6-45
ending by signing off 5-81
ID substitution expression 3-26
library substitution expression 3-23
library, changing
using LIBRARY OCL statement 5-67
using MENU command 6-22
using SLIB procedure 4-471
printer
changing using PRINT procedure 4-350
substitution expression 3-22
SSP-ICF
allowing to be used 6-36
stopping 6-48
status, displaying 6-40, 6-45
SESSION OCL statement 5-106
set direction of roll keys
ROLLKEYS procedure 4-406
SET procedure 4-454
SET utility control statement, $SETCP utility program A-106
SETA utility control statement, $SETCF utility program A-102
SETALERT procedure 4-457
$SETCF utility program A-105
changing communications information A-102
defining display station A-104
SETCF utility control statement, $SETCF utility program A-104
SETCOMM procedure 4-461
$SETCP utility program (defines communications) A-106
SETDUMP procedure C-31
SETPK utility control statement, $SETCF utility program A-105
setting
See also changing
switches
using SWITCH OCL statement 5-112
using SWITCH procedure 4-496
up
communications subsystems 4-96
network configuration for X.25 4-97
setting *(continued)*
up *(continued)*
the system 4-97
setting the return code 3-60
SEU (source entry utility) procedure 4-465
SEULOAD procedure 4-467
SEUSAVE procedure 4-468
severity level, automatic response, specifying
using NOHALT OCL statement 5-79
using NOHALT procedure 4-305
severity level, specifying using RESPONSE procedure 4-374
$SFGR utility program (generates display formats) A-108
show
See display
SHRFLOAD procedure 4-469
SHRFSAVE procedure 4-470
shutting down the system 6-48
signing off other display stations 6-5
signing off the system 5-81, 6-27
$SINCT utility program (restores network resource directory) A-110
$SINDL utility program (removes network resource directory) A-110
$SINR utility program (edits network resource directory) A-110
size
changing
disk files using COPYDATA procedure 4-111
disk files using RESTORE procedure 4-392
folder 4-8
library 4-9
library directory 4-9
file substitution expression 3-18
parameter substitution 3-15
region, changing 5-103
skipping procedure parameters 5-99
?SLIB? substitution expression 3-23
SLIB procedure 4-471
SMF procedure 4-473
SMF report file
options 4-474
SMFDATA procedure 4-474
SMFPRINT procedure 4-477
SMFSTART procedure 4-480
SMFSTOP procedure 4-482
SORT procedure 4-483
sorting
disk files
ideographic data using SRTX procedure 4-484
using DFU (data file utility) 4-150
using LIST procedure 4-259
using SORT procedure 4-483
using SRTX procedure 4-484
keys of indexed file using KEYSORT procedure 4-252

SOURCE conditional expression 3-46
source entry utility 4-465
source members, creating or changing using SEU procedure 4-465
space, reserving for job and scratch files 5-104
special E-format diskette files, transferring using POST procedure 4-345
SPECIFY procedure 4-483
spool file
canceling entries from
using CANCEL command 6-5
using CANCEL OCL statement 5-16
entries
copying using COPYPRT procedure 4-126
printing using BALPRINT procedure 4-34
holding output in 5-83
restarting the printing of entries 6-34
starting the spool writers for 6-36
status, displaying 6-40, 6-45
SPOOL utility control statement, $UASF utility program A-123
spool writer
changing number of separator pages 6-9
priority, changing 6-9
starting
using START command 6-36
using START OCL statement 5-109
status, displaying 6-40
spooling
changing
number of separator pages 6-9
printer ID to be used using CHANGE command 6-9
printer ID to be used using CHANGE OCL statement 5-18
copies to be printed
changing using CHANGE command 6-9
changing using CHANGE OCL statement 5-18
specifying using PRINTER OCL statement 5-83
copying output using COPYPRT procedure 4-126
deferring output
using CHANGE command 6-9
using PRINTER OCL statement 5-83
displaying output using COPYPRT procedure 4-126
printed output, specifying using PRINTER OCL statement 5-83
printing output using COPYPRT procedure 4-126
SRTX procedure 4-484
SRTXBLD procedure 4-485
SRTXLOAD procedure 4-486
SRTXSAVE procedure 4-488
SSP-ICF
disabling subsystems 4-157
enabling subsystems 4-170
SESSION OCL statements 5-106
SSP-ICF *(continued)*
sessions
starting 6-36
status, displaying 6-40
stopping 6-48
subsystems status, displaying 6-40
standby line, allowing 4-11
standby mode 6-23
start
disk cache 4-72
START command 6-36
start monitoring communications line using STARTM procedure 4-489
START OCL statement 5-109
starting
activity queue using OFCQ procedure 4-322
BASIC 4-39
BASIC procedures 4-40
BASIC programs 4-41
communications queues using OFCQ procedure 4-322
error reporting in an IBM Token-Ring Network using TRNMGR procedure 4-549
job queue 6-36
job queue priority, a specific 6-38
jobs 6-36
jobs running 6-36
printed output
using START command 6-36
using START OCL statement 5-109
printer load balancing using BALPRINT procedure 4-34
procedures 5-63
programs 5-105
SSP-ICF sessions
using ENABLE procedure 4-170
using START command 6-36
system activity 6-36
system service device 6-36
starting online problem determination 4-369
STARTM procedure 4-489
STARTM utility control statement, $MMST utility program A-83
statement formats
coding rules 5-4
statement syntax formats
description of 1-3
statements
See also OCL statements
CANCEL 3-60
comment (*) 3-9
end of data (/*) 5-123
EVALUATE 3-60
GOTO 3-67
IF conditional expression 3-54
message, informational 3-57

statements *(continued)*
message, system console 3-59
operation control language (OCL) 5-1
PAUSE 3-69
procedure control 3-1
procedures, allowed in 2-1
RESET 3-70
RETURN 3-71
TAG 3-67
STATEST procedure 4-490
station addresses
changing using ALTERCOM procedure 4-11
table of 4-17
stations, display, canceling 6-5
stations, display, releasing from procedures 5-11
status
communications, displaying 6-40
diskette drive, displaying 6-40, 6-45
display stations, displaying 6-40, 6-45
job queue, displaying 6-40, 6-45
jobs currently running 6-40, 6-45
MSRJE, displaying 6-40
printed output, displaying 6-40, 6-45
printers, displaying 6-40, 6-45
session, display station 6-40, 6-45
spool writers 6-40
SSP-ICF sessions, displaying 6-40
SSP-ICF subsystems, displaying 6-40
tape drive, displaying 6-40, 6-45
task control blocks 6-40
STATUS command 6-40
STATUSF command 6-45
stop
disk cache 4-72
STOP command 6-48
stop monitoring communications line using STOPM procedure 4-492
STOP OCL statement 5-111
STOPGRP procedure 4-491
STOPM procedure 4-492
STOPM utility control statement, $MMSP utility program A-83
stopping
activity queue using OFCQ procedure 4-322
communications queues using OFCQ procedure 4-322
error reporting in an IBM Token-Ring Network using TRNMGR procedure 4-549
job queue 6-48
jobs from running 6-48
printer load balancing using BALPRINT procedure 4-34
printing of output
using STOP command 6-48
using STOP OCL statement 5-111
procedures after each job step 5-29
stopping *(continued)*
SSP-ICF sessions 6-48
system 6-48
system service display station 6-48
storage size, changing 5-103
storing
See saving, copying
stream, job, running using JOBSTR procedure 4-246
STRTGRP procedure 4-493
#STRTUP1 procedure 4-4
#STRTUP2 procedure 4-6
subconsole
activating support 6-2
messages not responded to 6-40
messages, replying to 6-32
SUBR conditional expression 3-47
substitution expressions 3-10
?C'value'? 3-15
?CD? 3-16
?CLIB? 3-18
?Cn? 3-15
?DATE? 3-18
?F'A,name'? 3-19
?F'S,name'? 3-18
?L'position,length'? 3-19
?M'mic,position,length'? 3-21
?MENU? 3-22
?n? 3-11
?n'value'? 3-11
?nF'value'? 3-12
?nR? 3-13
?nR'mic'? 3-14
?nT'value'? 3-12
?PRINTER? 3-22
?PROC? 3-23
?R? 3-13
?R'mic'? 3-14
?SLIB? 3-23
?SYSLIST? 3-23
?TIME? 3-24
?USER? 3-24
?VOLID? 3-25
?WS? 3-26
combining 3-27
current library 3-18
defaults 2-5
evalutating 3-60
file size 3-18, 3-19
local data area 3-19
message members 3-21
nesting 3-27
procedures 2-5
restrictions 3-10
return code 3-16
session library 3-23
using effectively 2-24

subsystems
debugging using ICFDEBUG procedure 4-221
disabling using DISABLE procedure 4-157
enabling using ENABLE procedure 4-170
setting up using CNFIGICF procedure 4-96
status, displaying 6-40
subtracting values from parameters 3-60
SUMMARY report file 4-474
support, system, directory to 1-7
suppressing informational messages 6-20
SWDLOAD procedure 4-494
SWDSAVE procedure 4-495
SWITCH conditional expression 3-48
SWITCH OCL statement 5-112
SWITCH procedure 4-496
switched communications lines, specifying
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
switches
setting
using SWITCH OCL statement 5-112
using SWITCH procedure 4-496
testing for 3-48
using in prompt displays 5-99
switching
system printer 6-2
work station IDs 6-2
symbolic display station ID 5-121
syntax formats, description of 1-3
?SYSLIST? substitution expression 3-23
SYSLIST OCL statement 5-113
SYSLIST procedure 4-498
system
configuring using CNFIGSSP procedure 4-97
debugging using the SERVICE procedure C-29
directory to 1-7
entering procedures into 2-2
help support 4-199
measurement facility 4-473
powering off 5-82, 6-28
printer, placing online or offline 5-116, 6-53
restarting activity 6-36
security, defining using SECDEF procedure 4-441
shutting down 6-48
signing off 5-81, 6-27
stopping all activity 6-48
testing using the SERVICE procedure C-29
tracing activity using TRACE procedure C-32
turning off 6-28
system console
messages, replying to 6-32
responding to subconsole messages 6-40
sending messages to
using // ** statement 3-59
using MSG command 6-24
using MSG OCL statement 5-76
system console *(continued)*
testing for in procedures 3-30
transferring 6-15
system library
restoring using RESTLIBR procedure 4-386
saving using SAVELIBR procedure 4-431
system list device
changing
using PRINT procedure 4-350
using PRINTER OCL statement 5-86
using SYSLIST OCL statement 5-113
using SYSLIST procedure 4-498
substitution expression 3-23
system local data area 5-71
system measurement facility
creating a disk file 4-474
printing collected data 4-477
start collecting data 4-480
stop collecting data 4-482
system printer
assigning a different printer as 6-2
placing online or offline 5-116, 6-53
system procedures 4-1
system service display station
allowing to be used 6-36
stopping from being used 6-48
system service log, adding entries to C-30

T

?nT'value'? substitution expression 3-12
table of system tasks 1-7
tables, print image translation
chart of E-7
creating your own 5-60
specifying 5-59
tabs, insert xvi
TAG statement 3-67
taking the system console 6-15
tape
copying folders to using SAVEFLDR procedure 4-428
FILE OCL statement for 5-48
files
specifying for a program 5-48
saving
folders on 4-428
tape errors
using TAPESTAT procedure 4-509
tape problems, identifying using TAPESTAT 4-509
tape volume information using TAPESTAT 4-509

TAPECOPY procedure 4-500
TAPEINIT procedure 4-507
tapes
AUTO/NOAUTO parameters, changing defaults for 5-8
copying disk files to 4-416
copying folder members from using RETRIEVE procedure 4-401
copying folder members to using ARCHIVE procedure 4-22
copying libraries from 4-386
copying libraries to 4-193
copying network resource directory to 4-434
drive
allocating to a job 5-8
online or offline, placing 5-116, 6-53
status, displaying 6-40, 6-45
dump, specifying 5-7
erasing
using TAPEINIT procedure 4-507
files
DATAT conditional expression 3-34
existence testing 3-34
listing using LISTFILE procedure 4-267
permanent 5-51
retention days 5-51
preparing
using TAPEINIT procedure 4-507
renaming
using TAPEINIT procedure 4-507
restoring
disk files from 4-392
folder members from 4-401
libraries from 4-386
network resource directory from 4-389
retention days 5-51
saving
disk files on 4-416
folder members on 4-22
network resource directory on 4-434
testing for 3-53, 3-54
volume ID, assigning
using TAPEINIT procedure 4-507
TAPESTAT procedure 4-509
task control block, status, displaying 6-40
task directory 1-7
task dump, cannceling a job with 6-7
task dump, specifying device 5-7
TCB
See task control block
$TCOPY utility program
copying tape files A-112
temporary disk files
See resident files
terminal
See display station
terminal, multiple requesters, assigning to a program 5-11
testing
parameters 2-6
remote work stations using STATEST procedure 4-490
system using the SERVICE procedure C-29
testing for
autocall phone list completion 3-41
available disk space 3-30
disk files 3-31
diskette files 3-32
diskettes or tapes 3-53
display station type 3-36
enabled communications subsystems 3-37
evoked procedures 3-38
expressions 3-50, 3-52
external indicators 3-48
inquiry mode 3-39
job queue procedures 3-40
libraries 3-31
library load members 3-42
library procedure members 3-44
library source members 3-46
library subroutine members 3-47
maximum users of a MRT program 3-43
parameter values 3-50, 3-52
running procedures 3-29
security classification 3-45
switches 3-48
system console 3-30
tape files 3-34
tapes 3-54
volume IDs 3-53, 3-54
TEXTCONV procedure 4-511
TEXTDCT procedure 4-511
TEXTDOC procedure 4-512
TEXTFLDR procedure 4-530
TEXTLOAD procedure 4-531
TEXTOBJ procedure 4-532
TEXTPROF procedure 4-532
TEXTPRTQ procedure 4-533
TEXTREL procedure 4-534
TEXTSAVE procedure 4-536
things you can do using procedure control expressions 3-2
time
displaying 6-52
waiting until to start a job 5-118
TIME command 6-52
time-out value, SDLC, defining
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
?TIME? substitution expression 3-24

$TINIT utility program (initializing tapes) A-114
$TMSERV utility program
copying folder members A-115
copying folders A-116
listing folder members A-117
moving folders A-119
reorganizing folders A-118
restoring folder members A-120
restoring folders A-121
TOLIBR procedure 4-537
tone, answer, defining using SETCOM procedure 4-461
TRACE procedure C-32
tracing procedure processing 5-29
TRANSFER procedure 4-542
TRANSFER utility control statement, $BICR utility program A-5
TRANSFER utility control statement, $POST utility program A-85
TRANSFER utility control statement, $TCOPY utility program A-112
transferring the system console 6-15
translating
ASCII to EBCDIC 4-93
EBCDIC to ASCII 4-93
RPG specs to data definitions using IDDUXLAT procedure 4-227
translation tables
#188E188 5-60
#188E48 E-12
#188E64 E-14
#188E96 E-16
#96E48 E-8
#96E64 E-10
3262 printer, chart of E-7
translation tables, specifying 5-59
transmit
See communications, sending
tributary address
changing 4-11
table of 4-17
TRNMGR procedure 4-549
troubleshooting procedures 2-24
turning off the system 6-28

U

$UASC utility program (displays spool file entries) A-122
$UASF utility program (copies spool file entries) A-123
UIN utility control statement, $INIT utility program A-52
underlining in syntax formats 1-3
unit, control, placing online or offline 5-116, 6-53
UPDATE procedure 4-550
UPDATE utility control statement, $SFGR utility program A-108
UPSI (user programmable status indicators)
See switches
?USER? substitution expression 3-24
user ID, substitution expression 3-24
user identification file
changing using SECEDIT procedure 4-443
creating using SECDEF procedure 4-441
listing using SECLIST procedure 4-445
removing using SECDEF procedure 4-441
restoring using SECREST procedure 4-448
saving using SECSAVE procedure 4-451
user profiles, DW/36, creating and maintaining 4-532
using
See allocating
using help 4-199
utility control statements
general description of 1-2
introduction 1-2
where used 3-54
utility programs
$ARSP
changing alert indicators A-3
changing automatic response values A-3
$BICR
copying basic data exchange files A-5
listing basic data exchange files A-4
$BMENU (creates menus) A-7
$BUILD (corrects disk files) A-9
$COPY
copying files using OCL statements A-32
copying, organizing disk files A-10
lists files A-14
restoring disk files A-18
restoring network resource directory A-23
saving disk files A-24
saving network resource directory A-31
$CPPE (displays error messages) A-37
$CZUT (specifies alert messages) A-38
$DCOPY (copies diagnostic diskettes) A-38
$DDST (sorts keys of indexed file) A-39
$DELET (deletes files or libraries) A-40
$DPGP (prints graphich file) A-42
$DUPRD (copies diskettes) A-43
$FBLD
creating alternative indexes A-46
creating new disk files A-44
$FEFIX C-23
$FREE (compresses disk space) A-47
$HELP (system help support) A-47
$HIST (history file list, copy, and erase) A-48
$IDSET (defines remote IDs for communications) A-49
$IEDS (disables communications subsystems) A-50

utility programs *(continued)*
$IENBL (enables communications subsystems) A-51
$INIT (prepares diskettes) A-52
$LABEL (lists file and library names) A-53
$MAINT
changing library size A-60
condensing free library space A-63
copying job streams A-74
copying libraries and library members A-66
copying library members A-64
copying members to libraries from files A-72
creating libraries A-56
listing library members A-75, A-77
removing library members A-78
renaming library members A-64
restoring libraries A-80
saving libraries A-79
$MGBLD (creates message members) A-82
$MMSP (stops monitoring communications lines) A-83
$MMST (starts monitoring communications lines) A-83
$PACK (compresses disk space) A-84
$PNLM (defines phone lists) A-84
$POST
copying 5260 data files A-85
listing diskette files A-86
$PRCED (edits user ID file) A-87
$PRCLT (lists user ID file) A-87
$PRLST (lists user ID file) A-88
$PRPWD (changes user password) A-88
$PRUED (edits user ID file) A-89
$PRUID (password and badge security) A-90
$PRURS (restores user ID file) A-91
$PRUSV (saves user ID file) A-93
$RENAM (renames disk files, libraries, and folders) A-94
$RREDT (edits resource security file) A-95
$RRESC (resource security) A-96
$RRLST (lists resource security file) A-97
$RRSAV (saves resource security file) A-98
$RRSTR (restores resource security file) A-99
$RRTED (edits resource security file) A-101
$RRTLT (lists resource security file) A-101
$SETCF
changing communications information A-102
changing Print key information A-105
defining display station environment A-104
$SETCP (defines communications environment) A-106
$SFGR (generates display formats) A-108
$SINCT (restores network resource directory) A-110
$SINDL (removes network resource directory) A-110

utility programs *(continued)*
$SINR (edits network resource directory) A-110
$SVCASRV A-111
$TCOPY
copying tape files A-112
$TINIT (initializing tapes) A-114
$TMSERV
copying folder members A-115
copying folders A-116
listing folder members A-117
moving folders A-119
reorganizing folders A-118
restoring folder members A-120
restoring folders A-121
$UASC (displays spool file entries) A-122
$UASF (copies spool file entries) A-123
$XNLM (defines X.21 call lists) A-124
$XNSH (defines short hold mode line configuration) A-124
$XREST
restoring extended character file A-125
$XSAVE
saving extended character file A-126
#GCFR (requests an X.21 feature) A-127
called by SSP procedures 4-2

V

value, return, substitution expression 3-16
values, parameter, assigning 3-11
forcing 3-12
prompting 3-13, 3-14
temporarily 3-12
VARY command 6-53
VARY OCL statement 5-116
varying devices online or offline 5-116, 6-53
VDU
See display station
vertical print density, changing
using FORMS OCL statement 5-55
using PRINT procedure 4-350
using PRINTER OCL statement 5-83
view
See displaying, listing
virtual diskettes, exchanging data between folders and 4-332
virtual disks, exchanging data between folders and 4-332
VOL utility control statement, $INIT utility program A-52
VOL utility control statement, $TINIT utility program A-114
VOLID conditional expression 3-53, 3-54

?VOLID? substitution expression 3-25
volume ID
diskette, assigning
using INIT procedure 4-233
using the $INIT utility program A-52
tape, assigning
using TAPEINIT procedure 4-507
testing for 3-53, 3-54

W

WAIT OCL statement 5-118
wait time, error, changing
using ALTERCOM procedure 4-11
using SETCOMM procedure 4-461
waiting
for a file to become available 5-39
for the diskette drive to be allocated 5-10
until a time of day 5-118
until an amount of time has passed 5-118
warning
See precaution
work station
See display station
?WS? substitution expression 3-26
work station IDs
exchanging 6-2
logical, specified on WORKSTN OCL statement 5-121
WORKSTN OCL statement 5-120
writers, spool, displaying status 6-40
writing procedures 2-1
WSFLOAD procedure 4-552
WSFSAVE procedure 4-553
WSU procedure 4-554
WSU programs
compiling using WSU procedure 4-554
creating using SDA procedure 4-437
WSULOAD procedure 4-557
WSUSAVE procedure 4-558
WSUTXCR procedure 4-559
WSUTXEX procedure 4-561
WSUTXRV procedure 4-563

X

X.21 public data network
canceling a facility using REQUESTX procedure 4-373
requesting a facility using REQUESTX procedure 4-373
X.25
configuring using CNFIGX25 procedure 4-97
X.25 network
specifying using SETCOMM procedure 4-461
$XNLM utility program (defines X.21 call lists) A-124
$XNSH utility program (defines short hold mode line configuration) A-124
$XREST utility program
restoring extended character file A-125
XREST procedure 4-563
$XSAVE utility program
saving extended character file A-126
XSAVE procedure 4-563

Numerics

#188E188 translation table 5-60
#188E48 translation table E-12
#188E64 translation table E-14
#188E96 translation table E-16
3262 Printer
print belts
specifying using IMAGE OCL statement 5-59
table of E-1
translation tables
creating your own 5-60
specifying 5-59
table of E-7
3270 device emulation
using EM3270 procedure 4-168
using ES3270 procedure 4-178
IBM 5224 printer, considerations 5-55, 5-83
IBM 5224 printer, considerations 5-55, 5-83
IBM 5260 Retail System, transferring information from 4-345
#96E48 translation table E-8
#96E64 translation table E-10

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name _____

Company or Organization _____

Address _____

_____
City State Zip Code

Phone No. _____
Area Code

No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape. **Please do not staple.**

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS / PERMIT NO. 40 / ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 245
Rochester, Minnesota, U.S.A. 55901

Fold and tape. **Please do not staple.**

Cut Along Line

IBM

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

☐ If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

☐ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name _____

Company or Organization _____

Address _____

_____
City State Zip Code

Phone No. _____
Area Code

No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape. **Please do not staple.**

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS / PERMIT NO. 40 / ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 245
Rochester, Minnesota, U.S.A. 55901

Fold and tape. **Please do not staple.**

Cut Along Line

IBM®

IBM®

System Reference

International Business Machines Corporation

File Number
S36-36

Order Number
SC21-9020-5

Printed in U.S.A.

SC21-9020-05