Print Services Facility/MVS

S544-3672-03

System Programming Guide

Version 2, Release 2, Modification 0

Print Services Facility/MVS

S544-3672-03

System Programming Guide

Version 2, Release 2, Modification 0

Note!

Before using this information and the product it supports, be sure to read the general information in "Notices" on page xvii.

Fourth Edition (December 1996)

This edition applies to Version 2 Release 2 Modification 0 of the IBM Print Services Facility/MVS licensed program, Program Number 5695-040, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Be sure to use the correct edition for the level of the product.

See the Summary of Changes for the changes made to this publication. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

The IBM Printing Systems Company welcomes your comments. For your convenience, a form for reader's comments is provided at the back of this publication. You may either send your comments by fax to 1-800-524-1519, or mail your comments to:

INFORMATION DEVELOPMENT
THE IBM PRINTING SYSTEMS COMPANY
DEPARTMENT H7FE BUILDING 003G
PO BOX 1900
BOULDER CO 80301-9191

If you prefer to send comments electronically, use one of the following methods:

- Internet: print_pubs@vnet.ibm.com
- Fax: 1-800-524-1519

Internet

Visit our home page at http://www.can.ibm.com/ibmprinters

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices xvii
Programming Interface xvii
Trademarks xvii

Summary of Changes xix

Chapter 1. Introducing PSF/MVS and This Publication 1
What is PSF? 1
What Is Shipped With PSF/MVS? 1
What Do I Need to Know? 2
What Can I Do with PSF? 2
Why Should I Read This Publication? 2
How to Use This Publication 3
Summary of PSF Support 4
Printers Supported by PSF 6
Printers Attached through an Intermediate Device 6

Chapter 2. Defining Channel-Attached Printers to MVS 7
Prerequisites for Defining Channel-Attached Printers to MVS 7
  Selecting the Method 7
  Using Switches with Host-Connected, Channel-Attached Printers 8
  Using an Intermediate Device with Channel-Attached Printers 8
  Using ESCON Channel-Attached Printers 8
Defining IBM 3800 Printers with MVSCP 9
Defining IBM Channel-Attached 3820 Page Printers with MVSCP 10
Defining Parallel Channel-Attached Printers (AFP1 Family) with MVSCP . . . 10
Defining ESCON Channel-Attached Printers (AFP1 Family) with MVSCP . . . 11
Defining Channel-Attached Printers with HCD 13

Chapter 3. Establishing Network Nodes for SNA-Attached Printers 17
Overview of a Network 17
  Local-Attached Devices 17
  Communication-Attached Devices 18
VTAM Definitions for PSF Network Nodes 20
  SNA Devices, Attachment Mode, and Requirements 20
  VTAM Syntax Rules 23
  Defining a PSF Application Node 23
  Defining the Logon-Mode Table Entry 24
    Local-Attached Entries (Excluding PSF/2 and PSF for AIX) 25
    Communication-Attached Entries (Excluding PSF/2, PSF for AIX, and Token Ring) 25
  Defining Local Major Nodes 28
  Defining Switched Major Nodes 30
Defining Network Control Program (NCP) Major Nodes 33
  NCP Generation 34
    Switched Line 35
    Nonswitched Line 36
    Multipoint Nonswitched Line 39
    Token-Ring Subsystem Connection 42
General Network Considerations 43

Logon-Mode Table . . . 43
VTAM Definition Names and PSF . . . 44
LOGAPPL and the LU Statement . . . 44
PSF PRINTDEV Statement Considerations for Switched Lines . . . 44
| 3130, 3160, and 3935 Configuration . . . 45
DPF Considerations . . . 45
PSF Direct Considerations . . . 45

Chapter 4. Working with Libraries . . . 47
Library Format . . . 47
Concatenating Defined Libraries to System Libraries . . . 47
User Libraries in Direct-Printing Mode . . . 48

Chapter 5. Changing the System Environment . . . 51
How to Change the System Environment for PSF 2.2.0 . . . 51

Chapter 6. PRINTDEV Parameters . . . 53
Printer Sharing in the MVS Environment . . . 60
| JES Spool-Driven . . . 61
| VTAM RELREQ . . . 61
| Port Switching . . . 61

Chapter 7. Using Deferred-Printing Mode . . . 63
Defining PSF FSSs and FSAs for Printers . . . 63
Understanding the Syntax Notation Used in This Book . . . 64
Sample Initialization Statements for JES2 . . . 64
Parameters for the JES2 Initialization Statement . . . 65
FSS Statement . . . 65
PRT*nnnn* Statement . . . 66
Sample Initialization Statements for JES3 . . . 69
Parameters for the JES3 Initialization Statement . . . 70
FSSDEF Statement . . . 70
DEVICE Statement . . . 71
PSF Startup Procedures for AFP Printers . . . 74
Sample PSF Startup Procedures . . . 75
JCL for PSF Startup Procedures . . . 80
Separating Jobs . . . 81
Job Separator Pages . . . 82
Additional Distribution Information . . . 83
Separator Page Resources . . . 86
| Printing Copies of a Separator Page . . . 87
Distributed Print Function (DPF) Considerations . . . 87
PSF and DPF on a JES3 System . . . 88
Forms Marking . . . 88
Forms Marking of Multiple-Page Job Trailers . . . 90
Forms Marking of the Interrupt Message Page . . . 90
Carrier-Strip Marking . . . 90
Copy Marking and Offset Stacking . . . 91
Separating Data Sets or Copies of Data Sets . . . 92
System Management Facility Type 6 Records . . . 93
Disabled Mechanisms . . . 94

Chapter 8. Using Direct-Printing Mode . . . 97
How to Use Direct-Printing Mode . . . 98

Defining the DPSS 99
Application Program JCL for Direct-Printing Mode 100
Requesting Multiple Copies 102
3800 Considerations 102

Chapter 9. Using Cut-Sheet Emulation Mode 103
| How Do I Get Cut-Sheet Emulation Working? 103
| Do I Have Any Control over Using Cut-Sheet Emulation? 103
| Will Cut-Sheet Emulation Work with 3800 Line-Mode Conversion? 104
| Things to Consider When Using Cut-Sheet Emulation 104

Chapter 10. Enabling PSF 3800 Line-Mode Conversion 107
What is IBM 3800 Line-Mode Conversion? 107
How Does 3800 Line-Mode Conversion Work? 107
How Can PSF 3800 Line-Mode Conversion Be Enabled? 108
Considerations and Limitations of the 3800 Line-Mode Conversion Function 109
Implementation Checklist for PSF/MVS 3800 Line-Mode Conversion 111

Chapter 11. Managing Resources and Selecting Special Processing Parameters 113
How PSF Selects Special Processing Parameters 113
Selecting Region Sizes 114
Installation Options 114
For Each FSA 115
Redirecting the Message Data Set 115
Printing PSF Error Messages 116
Specifying Checkpoint Intervals 117
How PSF Selects Resources 117
Selecting a Forms Flash for a 3800 Printer 118
Selecting a Copy Group 119
Form Definitions in Deferred-Printing Mode 119
Form Definitions in Direct-Printing Mode 119
Page Definitions in Deferred-Printing Mode 120
Page Definitions in Direct-Printing Mode 121
Fonts in Deferred-Printing Mode 121
Fonts in Direct-Printing Mode 122
Searching for a Resource Specified for a Print Job 122
Using Form Definitions for Basic N_UP Printing 122
Using Form Definitions for Enhanced N_UP Printing 123
Using PSF, the Interface Attachment Feature, and the MICR Postprocessor 124
Maintaining Resource Integrity 124

Chapter 12. Understanding PSF Data Protection and Security 127
Security Labeling 128
Controlling Print Complete Notification 128

Chapter 13. Supporting Resident Fonts 129
PSF Resident Font Table (APSRFTBL) 130
Changing APSRFTBL 130
PSF Resident-Font Utility—APSRMARK 131
APSRMARK Considerations 132
Using APSRMARK 133
Syntax of APSRMARK Control Statements 133
APSRMARK Control Statement Parameters for Resident Fonts 133

Using Control Statements to Mark a Font 135
APSRMARK Control Statement Parameters (General) 135
Sample Control Statements for APSRMARK 136
Marking a Member with Resident Font Information 136
Generating a Report of Resident Font Marking 137
Diagnosing Problems with Printer-Resident Fonts 139

| **Chapter 14. Marking Fonts for Font Capture** 141
| Single-Byte Raster Font Capture 141
| Single-Byte Outline Font Capture 141
| Double-Byte Raster Font Capture 142
| Double-Byte Outline Font Capture 142

Chapter 15. Supporting Outline Fonts 143
IBM's Font Strategy 143
Understanding Outline Fonts 143
Concatenating Outline Fonts with Other Library Resources 144
Creating New Fonts 144
| Capturing New Outline Fonts 145
Font Mapping 145
Timing of Mapping Operations 146
Mapping Algorithms 146
Using the Font Mapping Tables 146
Adding a New Mapping of a Raster Font to an Outline Font 147
Adding a New Mapping of an Outline Font to a Raster Font 148
Disabling the Mapping of a Raster Font to an Outline Font 150
Adding a New Mapping of a GRID to a Raster Font 150
Adding a New Mapping of a GRID to an Outline Font 151

Chapter 16. Marking Resources for Distributed Print Function 153
Using APSRMARK to Mark Resources 153
Control-Statement Syntax for APSRMARK 154
Control-Statement Parameters for APSRMARK 154
Examples of Control Statements for APSRMARK 156
Creating a PSF Resource Library with All Members Marked PUBLIC . 156
Creating a PSF Resource Library with Some Members Marked PRIVATE 157
Modifying an Existing PSF Resource Library 158
Printing a Report of Resource Markings in a Partitioned Data Set . . . 159
Using Page Segments 160
Using DPF Fonts and Printer-Resident Fonts with APSRMARK 161
Changing How Fonts Are Marked 162
Fonts Already Marked for Use by DPF Only 162
Fonts Already Marked for DPF and Printer-Resident Use 162
Fonts Already Marked PRIVATE 162

Chapter 17. Using Installation Exits 163
Do's and Don'ts 163
Changing Exit Routines 164
Common Control Blocks 166
APSGEXTP Area 166
Exit-Communication Areas: APSUECA and APSUECE 169
Sequence of Exit Invocation 174
Control Flow with APSUX07 174
Separator-Page Exits: APSUX01, APSUX02, and APSUX03 175

Input for Separator-Page Exits 175
Output for Separator-Page Exits 175
Processing Multiple-Page Job Trailers with Forms Marking 175
Programming with Separator-Page Exits 175
JES Job-Separator-Page Areas—IAZJSPA and IAZJSEP 177
Exit Table, APSUCOM 179
Block Letter Tables, APSUBTBL 180
Service Routines 181
Block-Letter Routine, APSUBLK 181
Input for APSUBLK 181
Output for APSUBLK 181
Programming Considerations for APSUBLK 182
Exit-Put Routine, APSUPUT 182
Input for APSUPUT 183
Output for APSUPUT 183
Logical-Record Processing Exit: APSUX04 183
Input for APSUX04 183
Output for APSUX04 184
Programming Considerations for APSUX04 184
SMF Type 6 Record Exit: APSUX05 188
Input for APSUX05 189
Output for APSUX05 189
Programming Considerations for APSUX05 189
PSF Message Exit: APSUX06 190
Input for APSUX06 191
Output for APSUX06 191
Programming Considerations for APSUX06 191
Resource-Management Exit: APSUX07 193
Invoking APSUX07 193
Sequence of Events 194
APSUX07 Subroutine Calls 194
Programming Considerations for APSUX07 195
Page Segment Considerations 195
Special Performance Considerations 195
3800-Compatible Media Origin Considerations 196
Considerations on Suppressing an Interrupt Message Page 197
| Considerations on Redirecting a Message Data Set 197
Overlay Considerations 197
| Considerations on Converting a Multiple-Celled IM1 Image 197
Considerations on Eject-to-Front-Facing Function 198
| Considerations on Restartable Abends 198
| Considerations on Cut-Sheet Emulation Mode 198
Considerations on Forms Marking of an Interrupt Message Page 199
Considerations Related to Page Definitions and Form Definitions 199
Processing Considerations 200
Selection Hierarchy 200
Data-Set Termination Considerations 200
3800 Considerations 201
3800 Forms Marking 201
Considerations Related to User Libraries 201
Considerations Related to Inline Resources 201
Considerations Related to Outline Fonts 202
Considerations Related to Mapping to Outline Fonts 202
Considerations Related to Recover-from-Font-Not-Found Flag 203

| Considerations Related to Huffman Compression 203
Initialization Call (INIT) 203
Input and Output Indicators for INIT 204
Programming Considerations for INIT 204
Begin-Data-Set Call (BDSC) 205
Input and Output Indicators for BDSC 205
Programming Considerations for BDSC 205
Resource-Access Call (RAC) 205
Input and Output Indicators for RAC 207
Programming Considerations for RAC 207
RAC and Overlays 207
RAC and Page Segments 208
RAC and Fonts 208
Restrictions 209
RAC, Page Definitions, and Form Definitions 209
Using RAC in Processing Page Definitions and Form Definitions 209
Resource-Load Call (RLC) 210
Input and Output Indicators for RLC 211
Programming Considerations for RLC 211
Resource-Deletion Call (RDC) 211
Input and Output Indicators for RDC 211
Partial Control of Resource Deletion 211
Full Control of Resource Deletion 212
Using RDC to Control Resources 212
Termination Call (TERM) 213
Input and Output Indicators for TERM 213
| MVS Download Print Parameter Exit: APSUX15 213
Direct-Print Exit: APSUX16 214
Begin-Data-Set Call (BDSC) 214
Input for APSUX16 Begin-Data-Set Call 214
Output for APSUX16 Begin-Data-Set Call 214
Message Call for APSUX16 215
Input for APSUX16 Message Call 215
Output for APSUX16 Message Call 215
End-Data-Set Call (EDSC) 215
Input for APSUX16 End-Data-Set Call 215
Output for APSUX16 End-Data-Set Call 216
Programming Considerations for APSUX16 216

Chapter 18. Understanding AFP Performance 219
The Printer 219
Mechanical Speed 219
Internal Processing Speed 219
Paper Size 220
Printer Options 220
Raster Storage 220
Printer Features and RPQs 220
Duplex Printing 220
The Print Job 220
Data Density and Complexity 220
Characteristics of Print Data Sets 221
Number and Type of Resources 222
Inline Resources 222
Resident Fonts 222

Font Pruning of Raster Fonts 222
| Captured Fonts 223
Size of Print Jobs 223
Using Outline Fonts 223
| Transmission and Compression of Data 223
The Attachment Hardware 225
Host-Connected, Channel-Attached Printer 225
Local-Attached Printer 225
Communication-Attached Printer 225
PSF Direct or Distributed Print Function (DPF) 226
PSF Direct 226
DPF 226
Size of the VTAM Request Unit (RU) 226
VTAM PSNDPAC and DELAY Parameters 227
NCP MAXOUT Value 227
| TCP/IP-Attached Printer 227
The Host System 228
Processing Capability 228
Amount of Real Storage Available 228
Capacity of the I/O Subsystem 228
System Tuning 229
Blank Compression for Line Data (COMPRESS Parameter) 229
| Compressing Data by Use of the Huffman Algorithm 229
Printer System Data Rate Requirements 230
Printer Data Checks 231
Buffers 231
Block Size 232
Data Separation 232
Functional Subsystem Interface (FSI) 232
Checkpointing 232
Tracing 233

Chapter 19. Connecting a TCP/IP-Attached IPDS Printer 235
| Requirements for Using a TCP/IP-Attached IPDS Printer 235
| Overview 235
| Connecting TCP/IP-Attached Printers 236
| Defining the Communications Control Unit to MVS 236
| Modifying the TCP/IP profile in MVS 237
| DATABUFFERPOOLSIZE Statement 237
| SMALLDATABUFFERPOOLSIZE Statement 238
| TINYDATABUFFERPOOLSIZE Statement 238
| KEEPALIVEOPTIONS Statement 238
| GATEWAY Statement 238
| Installing the i-data 7913 IPDS Printer LAN Attachment 238
| Pinging the Printer 239
| Ping Is Not Successful 239
| Ping Is Successful 239
| Defining the Printer to JES 240
| Defining the Printer to PSF 240
| Operating a TCP/IP-Attached Printer 244
| Starting a TCP/IP-Attached Printer 244
| Stopping a TCP/IP-Attached Printer 244

Chapter 20. Responding to PSF Errors 247

Accumulating PSF Error Messages 247
Repeating Error Messages 247
Recovering from PSF Errors 248
Recovering from Printer-Hardware Problems 248
Terminating a Printer FSA 248
Understanding Restartable Abends 249
Requesting Traces and Dumps 249

Chapter 21. Supporting AFP Output on Microfilm 251
PSF Hardware Attachment Requirements for Microfilm Devices 251
PSF Software Requirements for Microfilm Devices 251
PSF System Programmer Tasks for Microfilm Devices 252
Defining the Microfilm Device as Channel-Attached or ESCON-Attached . . 253
Defining the Channel-Attached or ESCON-Attached Microfilm Device to MVS 253
Defining the Microfilm Device to JES2 and JES3 253
PRINTDEV Parameters for Microfilm Devices 254
Defining the Microfilm Device to PSF 255
Redirecting Message Data Sets for Microfilm Jobs 258
Working with Resource Libraries Required by Microfilm Devices 258
Working with the Microfilm Setup Resource Object Container Library . . . 258
Working with Font Libraries 258
Managing Resources and Selecting Special Processing Parameters 259
Generating Microfilm Setup Resource Object Containers 259
Using the Font Library Update (FLU) Utility 259
Considerations When Using the FLU 260
Running the FLU 260
FLU Return Codes 261
Printing to Microfilm 261
Operating a Microfilm Device 262
Microfilm Data Protection and Security Considerations for PSF 262
Using PSF Installation Exits for Microfilm Devices 262
Understanding PSF Performance Issues That Pertain to Microfilm Devices . 262
Responding to PSF Errors Reported by Microfilm Devices 263
Recovering or Restarting AFP Print Jobs for Microfilm 263

Appendix A. APSRMARK Sample Jobs 265

Appendix B. Font Resource Tables Supplied by PSF 267
APSRFTBL Font Resource Table 267

Appendix C. Field Descriptions for APSUX07 279
XTP7 Parameter Area (DSECT) 279
XTP7 Fields 284
APSLCRL Parameter Area (Concatenation Retrieval List) 291
APSLCRL Fields 293
CRL Data Area 294
APSURLST Parameter Area (Resource List) 294
APSURLST Fields 296
RLSTFEXT Parameter Area (Font Extension Area) 300
RLSTFEXT Fields 301
Processing Indicator Cross-Reference for APSUX07 302

Appendix D. APTRCONV Resource Conversion Utility for VSE 305

Converting a Resource 305

Appendix E. Reasonable Resource Loading Values (RRLV) 307

Appendix F. Font Conversion Program, 240-pel to 300-pel (APSRCF30) 309
Understanding APSRCF30 309
  Using APSRCF30 309
  Error Handling 310
  APSRCF30 Considerations 311
  APSRCF30 Sample Jobs 311
Converting All 240-Pel Fonts in Source Library (APSRAL30) 312
  Input to APSRAL30 312
  Output from APSRAL30 313
  APSRAL30 Processing 313
  APSRAL30 Considerations 313
  Example for Selected Font Conversion 314

Appendix G. Installation Verification Program (IVP) Example 317

Appendix H. PSF Programming Interfaces 319
General-Use Programming Interfaces 319
Product-Sensitive Programming Interfaces 319
Interfaces for Tailoring PSF/MVS 319
Installation-Wide Exits 319

Appendix I. Mapping Tables for Outline Fonts 321

Appendix J. Installing the i-data 7913 IPDS Printer LAN Attachment . . 329
Understanding How the i-data 7913 IPDS Printer LAN Attachment Works . . 329
Checking the i-data 7913 IPDS Printer LAN Attachment 330
Configuring the bootp Daemon 331
  Understanding How the bootp Daemon Works 331
  Completing the bootp Configuration Worksheet 332
  Editing the bootptab File 332
    Example 1: A Basic bootp Configuration 333
    Example 2: A More Complex bootp Configuration 334
Setting Up the bootp Daemon on AIX 335
Setting Up the bootp Daemon on OS/2 336
Configuring a Twinax Printer 336
Setting the Controls for the i-data 7913 IPDS Printer LAN Attachment 336
  For the IBM Token-Ring Network Model 336
  For the Ethernet LAN Model 337
Operating the i-data 7913 IPDS Printer LAN Attachment 338
  Attaching the i-data 7913 IPDS Printer LAN Attachment 338
  Starting the i-data 7913 IPDS Printer LAN Attachment 338
  Shutting Down the i-data 7913 IPDS Printer LAN Attachment 339
Testing the i-data 7913 IPDS Printer LAN Attachment 340
Troubleshooting the i-data 7913 IPDS Printer LAN Attachment 342

Appendix K. Related Publications 345
Print Services Facility/MVS (PSF/MVS) 345
Advanced Function Presentation (AFP) 345
Advanced Function Presentation for Microfilm 346
MVS/Enterprise Systems Architecture (MVS/ESA) Version 4 346

MVS/Enterprise Systems Architecture (MVS/ESA) Version 5 347
| OS/390 Release 3 348
VTAM and NCP 349
Fonts 350
Print Services Facility for OS/2 (PSF/2) 350
Print Services Facility for AIX (PSF for AIX) 350
Printers 351
| TCP/IP 351
| i-data 352

Glossary 353
Sources 353
References 353

Index 377

Figures

1. Example of a Channel-Attached Printer Configuration 8
2. Sample JCL for Defining a 3800 Model 3 Printer 9
3. Sample JCL for Defining a Host-Connected, Channel-Attached 3820 Printer 10
4. Sample JCL for Defining a Channel-Attached Printer 11
5. Sample JCL for Defining a Host-Connected ESCON Channel-Attached Printer 12
6. Sample JCL for Defining a Host-Connected ESCON Channel-Attached Printer 12
7. Interactive System Productivity Facility Primary Option Menu 13
8. HCD Define Device Parameters/Features Screen 14
9. Sample PSF Configurations for Local-Attached Devices 18
10. Sample PSF Configurations for Communication-Attached Devices 19
11. Sample PSF Configurations for Communication Attachment to DPF or PSF Direct 20
12. PSF Application Node 24
13. Sample APPL Statement 24
14. Sample Logon-Mode Table Entry for Local-Attached Devices 25
15. Sample Logon-Mode Table Entry for Communication-Attached Devices . 25
16. Sample Logon-Mode Table Entry for Token-Ring Devices 26
17. Example Showing Three Local Major Nodes 28
18. Sample Local Major Node Definitions for Different LU Protocols 28
19. Sample Configuration of Switched Major Nodes 30
20. Sample Definitions of Switched Major Nodes 31
21. Sample NCP Major Node 34
22. Sample NCP Macroinstructions for Switched lines 35
23. Sample NCP Macroinstructions for Nonswitched Lines 36
24. Sample NCP Macroinstructions for Multipoint Lines 40
25. Sample Initialization Statements for JES2, for Three Printers 65
26. Sample Initialization Statements for JES3, for Three Printers 70
27. Sample Initialization Statements for JES3 70
28. Sample PSF Startup Procedure for Three Printers, Showing Multiple FSAs 76
29. Sample PSF Startup Procedures for Three Separate Printers 78
30. Example of a Default PSF Job-Header Page (APSUX01) 82
31. Example of a Fast-Printing Job-Header Page for PSF (APSUX01P) . . . 83
32. Sample Header Page 85
33. Sample Header Page 86
34. Example of Forms Marking for Cut-Sheet Paper 88
35. Forms Marking for Continuous Forms Not Printed on a 3800 Printer . . . 89
36. Example of Forms Marking 89
37. Example of Carrier-Strip Marking on the 3800 Printer 90
38. Example of Copy Marking to Simulate Offset Stacking 91
39. Job Stream for Creating an IEFSSNBB Member of SYS1.PARMLIB . . . 99
40. Sample JCL to Enable Direct Printing 100
41. A Printer That Supports Resident Fonts 129
42. Sample JCL for Marking Resident Fonts 135
43. Example of the Use of JCL Statements to Mark a Member of a Data Set 136
44. Example of the Use of JCL Statements in Generating a Report 137
45. Printed Output of Printer-Resident Report 138
46. Comparison of an Outline Font Character with a Raster Font Character 144

47. Access to Resources by DPF 153
48. Sample JCL for Creating a PSF Resource Library and Marking all Members PUBLIC 156
49. Sample JCL for Creating a PSF Resource Library and Marking Some Members PRIVATE 157
50. Sample JCL for Modifying an Existing PSF Resource Library 158
51. Sample JCL for Reporting Resource Markings in a Partitioned Data Set 159
52. Sample Output from a Resource-Marking Report 160
53. Use of STEPLIB Statement for Exit Routine in PSF Startup Procedure 164
54. Format of the APSGEXTP Area 167
55. Format of Exit-Communication Area, APSUECA 169
56. Format of Exit-Communication Area Extension, APSUECE 171
57. Format of JES Job-Separator-Page Area, IAZJSPA. 178
58. Format of JES Job-Separator-Page Area Extension, IAZJSEP 178
59. Format of the Output in the Work Buffer, ECAWKBUF 182
60. APSUX04 Indicators in XTP4FLAG Field 187
61. Format of JES Index Table, IAZIDX 188
62. Appearance of the TCP/IP Connection between an MVS Host and a Printer 235
63. Actual Physical TCP/IP Connection between an MVS Host and a Printer 236
64. Sample Parameters in the TCP/IP Profile 237
65. Sample JCL Startup Procedure for a TCP/IP-Attached Printer (APSWPROT) 242
66. Example of AFP Data Being Transmitted to a Microfilm Device 252
67. Sample JCL Startup Procedure for a Microfilm Device (APSWPROM) . 256
68. APSWRFLU Sample Job Provided in SYS1.SAMPLIB 261
69. XTP7 Parameter Area Format 280
70. APSLCRL Parameter Area Format 292
71. Format of the CRL Data Area 293
72. APSURLST Parameter Area Format 295
73. Format of the RLSTFEXT Parameter Area 301
74. Sample JCL for Converting the MVS PAGEDEF Resource for Use by VSE 305
75. Sample JCL to Route Job to Reader 305
76. IVP Output Printed with 240-Pel Resolution 317
77. Global Resource Identifiers (GRIDs) to Font Names 322
78. Font Names to Global Resource Identifiers (GRIDs) 325
79. i-data 7913 IPDS Printer LAN Attachment in a TCP/IP LAN Environment 330
80. Data Flow between the i-data 7913 IPDS Printer LAN Attachment and the LAN 331
81. IBM Token-Ring Network Model of the i-data 7913 IPDS Printer LAN Attachment 337
82. Ethernet LAN Model of the i-data 7913 IPDS Printer LAN Attachment . 337
83. Status indicators on the i-data 7913 IPDS Printer LAN Attachment . . . 339
84. Sample Test Page for IBM 7913 Token-Ring Network Model 341
85. Sample Test Page for Ethernet Model 341

Tables

1. SNA Devices: Requirements 21
2. Initialization Parameters for the PRINTDEV Statement: Summary 53
3. JCL DD Allocation vs. Dynamic Allocation 102
4. Minimum Storage Required for FSS and FSAs 114
5. Hierarchy of Flash Selection in Deferred-Printing Mode 118
6. Hierarchy of Flash Selection in Direct-Printing Mode 118
7. Hierarchy of Form Definition Selection in Deferred-Printing Mode . . . 119
8. Hierarchy of Form Definition Selection in Direct-Printing Mode 119
9. Hierarchy of Page Definition Selection in Deferred-Printing Mode . . . 120
10. Hierarchy of Page Definition Selection in Direct-Printing Mode 121
11. Hierarchy of Font Selection in Deferred-Printing Mode 121
12. Hierarchy of Font Selection in Direct-Printing Mode 122
13. Resident Font Diagnosis 139
14. PSF Exit Routines in SYS1.SAMPLIB 165
15. Summary of APSUX07 Subroutine Calls 194
16. Approximate Maximum Attachment Data Rates 226
17. Approximate Maximum Printer Data Rates 231
18. Sample Jobs in SYS1.SAMPLIB 265
19. Input and Output Summary for APSUX07 Processing Indicators 302
20. Reasonable Resource Loading Values (RRLV) 307
21. Font-Name-to-Grid Table, APSRNTGM 321
22. Grid-to-Font-Name Table, APSRGTNM 321
23. Configuration Worksheet for bootp 332
24. Problems in Installing the 7913 Attachment 343
25. Problems in Operating the 7913 Attachment 344

Notices

References in this publication to products or services of IBM do not suggest or imply that IBM will make them available in all countries where IBM does business or that only products or services of IBM may be used. Noninfringing equivalents may be substituted, but the user must verify that such substitutes, unless expressly designated by IBM, work correctly. No license, expressed or implied, to patents or copyrights of IBM is granted by furnishing this publication. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, USA.

Programming Interface

This publication is intended to help you customize the Print Services Facility/Multiple Virtual Storage (PSF/MVS) licensed program (Program Number 5695-040).

This publication also describes *product-sensitive programming interface information and associated guidance information*. Product-sensitive programming interfaces are provided for use by the customer installation in performing tasks such as tailoring, monitoring, modifying, or diagnosing this product. The use of such interfaces creates dependencies on the detailed design or implementation of IBM products. Product-sensitive interfaces should be used only for these specialized purposes. Because of their dependency on detailed design and implementation, programs written to such interfaces may need to be changed to run with new product releases or versions, or as a result of service. *Product-sensitive programming interface and associated guidance information* is identified where it occurs, with an introductory statement to a chapter or section.

Trademarks

The following are trademarks or registered trademarks of the IBM Corporation:

ACF/VTAM
Advanced Function Presentation
Advanced Function Common Control Unit
AFCCU
AFP
AIX
APL2
Application System/400
Bar Code Object
BCOCA
DCF
Document Composition Facility
ESCON
Extended Services for OS/2
GDDM
Hardware Configuration Definition
IBM
Intelligent Printer Data Stream
IPDS
MO:DCA
MVS/ESA
MVS/SP
MVS/370
OGL
OPC
Operating System/2
Operating System/400
OS/2
OS/400
Personal System/2
PrintManager
Print Services Facility
Proprinter
PS/2
PSF
PSF for AIX
RACF
RS/6000
System/370
System/390
VTAM

UNIX is a trademark of a company other than the IBM Corporation.

Summary of Changes

The following information is new or changed in this edition, S544-3672-03:

- New chapters and an appendix have been added:
  - Chapter 9, “Using Cut-Sheet Emulation Mode” on page 103

    Cut-sheet emulation is the ability of a continuous-forms printer to print output similar to output from a cut-sheet printer.
  - Chapter 14, “Marking Fonts for Font Capture” on page 141

    Font capture is the ability of some printers to store downloaded fonts as temporary printer-resident fonts.
  - Chapter 19, “Connecting a TCP/IP-Attached IPDS Printer” on page 235

    PSF supports TCP/IP-attached printers, including IPDS printers attached by use of the i-data 7913 IPDS Printer LAN Attachment.
  - Appendix J, “Installing the i-data 7913 IPDS Printer LAN Attachment” on page 329

    The i-data 7913 IPDS Printer LAN Attachment is a hardware protocol converter you can use to communicate between PSF and a coaxial or twinaxial IPDS printer.
- In Chapter 11, “Managing Resources and Selecting Special Processing Parameters” on page 113, the resource-hierarchy tables, which describe how PSF selects resources, have been updated.
- In Chapter 6, “PRINTDEV Parameters” on page 53, new PRINTDEV parameters are described: ACQINTV, CONNINTV, IPADDR, PORTNO, RELINTV, and RELMODE. Also, the section on printer-sharing in the MVS environment has been replaced.
- Information on installation exits has been added to Chapter 17, “Using Installation Exits” on page 163. The following new sections have been added to that chapter:
  - “Considerations on Converting a Multiple-Celled IM1 Image” on page 197

    Using the APSUX07 exit, you can control whether a multiple-celled IM1 image is converted to a single IOCA image or to multiple IOCA images.
  - “Considerations on Restartable Abends” on page 198

    Using the APSUX07 exit, you can prevent PSF from restarting after abend situations.
  - “Considerations on Cut-Sheet Emulation Mode” on page 198

    Using the APSUX07 exit, you can request that PSF use cut-sheet emulation *only* for data sets with pages small enough to print 2 on a sheet.
  - “Considerations Related to Huffman Compression” on page 203

    Using the APSUX07 exit, you can request that PSF compress IPDS data by use of the Huffman compression algorithm before sending it to the printer.

- Information about the IBM 3935 printer and the IBM 4247 printer has been added.
- Information on the use of the Huffman compression algorithm has been added.
- Information about the Remote PrintManager (RPM) has been removed, because IBM no longer supports RPM.

New or changed information is marked by change bars in the left margin.

Chapter 1. Introducing PSF/MVS and This Publication

This publication provides information about changing and maintaining Print Services Facility/Multiple Virtual Storage (PSF/MVS) licensed program (Program Number 5695-040), hereafter referred to as PSF.

This publication is written on the assumption that you have experience in system programming and are familiar with Advanced Function Presentation (AFP) printers. In this publication, the word *printing* refers to presentation on paper, foils, labels, or microfilm. (*Microfilm* can mean either microfiche or 16mm film.)

What is PSF?

PSF is a licensed program that manages and controls the input and output data streams required by supported AFP printers or microfilm devices. In other words, PSF is a printer-driver program. It runs on the following MVS systems:

- MVS/ESA
- OS/390

PSF combines print data with other resources and printing controls to produce AFP data that can be printed.

PSF can be used in either of two ways. In *deferred-printing mode*, it can serve as the output writer under Job Entry Subsystem (JES). In *direct-printing mode*, it can serve as the access method to process an output data set. Printers attached by means of Systems Network Architecture (SNA) or Transmission Control Protocol/Internet Protocol (TCP/IP) cannot be used in direct-printing mode.

What Is Shipped With PSF/MVS?

The following optional programs are shipped with PSF on their own tapes, with their own installation instructions in separate program directories, and with their own separate publications:

- Advanced Function Presentation Application Programming Interface (AFP API)
- Advanced Function Presentation Conversion and Indexing Facility (ACIF)
- Page Printer Migration Program
- MVS Download
- AFP Upload for AIX
- NetSpool
- IP PrintWay

These programs can assist you in producing the AFP data stream (MO:DCA-P data) from high-level programming languages, in converting and indexing data, in converting jobs from Xerox to AFP printers, and in transmitting data to and from your MVS system. For details, refer to the publications for these programs.

What Do I Need to Know?

This publication contains information for use by system programmers who install PSF using System Modification Program/Extended (SMP/E); change the system environment; create a communication network; maintain data security; react to abnormal events; and code exit routines.

You should have system-level programming knowledge. For example, you should thoroughly understand:

- Job control language (JCL)
- Job entry subsystem (JES)
- Advanced communication functions for non-channel-attached printers:
  - Virtual Telecommunications Access Method (VTAM)
  - Network Control Program (NCP)

You should also be familiar with PSF on the level described in *Print Services Facility/MVS: Application Programming Guide*.

What Can I Do with PSF?

You can use PSF to do the following:

- Receive print jobs, access resources for print jobs, and send print commands to a printer. PSF supports both deferred printing, under job entry subsystem (JES); and direct printing, when it is specified in an application program.
- Manage resources required for the print job, such as form definitions, page definitions, fonts, page segments, overlays, and object containers.
- Identify (mark) resources as printer-resident or as managed by the Distributed Print Function (DPF) of PSF/2.
- Provide operator control of printers in deferred-printing mode.
- Provide problem diagnosis and error recovery.

You can use exits provided with PSF to do the following:

- Modify header or trailer separator pages
- Modify, add, or suppress output records
- Modify System Management Facility (SMF) type 6 records
- Inspect messages for deferred and direct printing
- Manage resources

Why Should I Read This Publication?

The purpose of this publication is to help you with the system programmer's primary task of changing and maintaining PSF.

If you need to install PSF, read *Print Services Facility/MVS: Program Directory* before continuing. The program directory contains most of the instructions for installation. Information about modifying the host system for PSF is presented in this book.

If you need a description of the job control language (JCL) used to print jobs on AFP printers, a description of resources used in AFP printing (data streams, form definitions, page definitions, fonts, page segments, overlays, and object containers), or examples of some common printing tasks, read *Print Services Facility/MVS: Application Programming Guide*. The application guide contains instructions for using PSF.

How to Use This Publication

This publication is organized so that you can read it sequentially or refer directly to individual tasks.

The beginning sections of this publication, through Chapter 1, describe:

- Concepts underlying PSF
- Prerequisites for system programmers
- How to use this publication
- What applications are supported by PSF

Chapter 2 through Chapter 12 are more specific. They describe:

- How to define channel-attached printers to MVS
- How to create network nodes for SNA-attached printers
- How to install library data sets
- How to change the system environment
- How to use the PRINTDEV parameters
- How to use deferred-printing mode
- How to use direct-printing mode
- How to take advantage of the cut-sheet emulation capability of some AFP printers
- How to use 3800 line-mode conversion
- How to manage resources and select special processing parameters
- How PSF protects data

Chapter 13 through Chapter 21, which are specialized, describe:

- How to use and change the APSRMARK utility to support resident fonts
- How to mark fonts for font capture
- How to use outline fonts
- How to use the APSRMARK utility to mark resources for DPF
- How to use installation exits
- How to refine PSF to affect system and printer performance
- How to connect and operate a TCP/IP printer
- How to respond to PSF error messages
- How to enable users to print to microfilm

Appendixes provide the following reference information:

- Lists of sample jobs supplied by PSF
- A copy of the APSRFTBL module code that lists code pages and character sets for some printers
- Field descriptions for the APSUX07 installation exit
- Information about APTRCONV, the Virtual Storage Extended (IBM System/370) (VSE) resource conversion utility program
- Information about reasonable resource loading values (RRLV)

- Information about APSRCF30, the font conversion utility program (to convert 4028 fonts)
- An example of output obtained after running the installation verification program (IVP)
- Lists of PSF macros for programming interfaces
- Copies of the optional user tables that map raster fonts to outline fonts and outline fonts to raster fonts
- An explanation of how to install and configure an i-data 7913 IPDS Printer LAN Attachment
- Lists of publications containing additional information about the system, hardware, or software used by PSF, or about installation and maintenance

Summary of PSF Support

PSF/MVS is the IBM printer-driver program that supports AFP printers and Computer Output Microfilm (COM) in the MVS operating environment. Using the Advanced Function Presentation (AFP) data stream and the Intelligent Printer Data Stream (IPDS), The printers supported by PSF print data that is already composed into pages. If print data has not already been composed, PSF can compose pages from line data. For more information about using PSF to print, refer to *Print Services Facility/MVS: Application Programming Guide*. For more information about AFP and AFP products, refer to *Guide to Advanced Function Presentation*. For information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.” For information about submitting jobs to print to microfilm, refer to *Print Services Facility/MVS: Application Programming Guide*,

In addition to the functions available for printing output, PSF provides the following:

- **3800 line-mode conversion:** PSF supports the printing of 3800 and line data in compatibility mode, providing a dynamic translation of 3800-1 print resources to enable users to print 3800 print data sets on other AFP printers.
- **Diagnostic aids:** PSF supplies diagnostic aids that maintain a record of events within PSF. If an abnormal termination occurs or if the formatted output is not as expected, information in the diagnostic aids helps you in determining and diagnosing problems.
- **Deferred-printing mode (JES):** In deferred-printing mode, PSF is an output writer, controlled by JES, that processes the spooled output from JES and sends a data stream to a page printer.
- **Direct-printing mode:** In direct-printing mode, PSF acts as a method for access to the printer.
- **Disabled-mechanism support:** A disabled mechanism, as the phrase is used here, is a printer part that is not functioning correctly—the part that controls duplex printing, for example. PSF provides disabled-mechanism support for some printers, printing the data set while ignoring the affected function. If a printer mechanism is disabled, output may not be as expected. To avoid unexpected results, you can reassign output to another printer until the disabled mechanism is repaired. For more details about which AFP printers provide this support, refer to *Advanced Function Presentation: Printer Information*.

- **Dynamic reconfiguration management (DRM):** PSF provides DRM for AFP1 and 3820 channel-attached printers, if using JES2 4.2.0 and above under MVS/ESA SP 4.2.0 and above and, for 3820s only, VTAM 3.4.0. With DRM, these devices can be dynamically added to or deleted from the I/O configuration, or changed. For more information, see Chapter 2, “Defining Channel-Attached Printers to MVS.”
- **Error recovery:** PSF provides for error recovery. PSF maintains internal checkpoints for each print data set. If an error or an operator command interrupts the processing of a data set, recovery routines use this checkpoint data to continue printing.
- **Hardware configuration definition (HCD):** The HCD function configures AFP1 and 3820 channel-attached printers. MVSCP and HCD are in MVS SP 4.1.0 and above. For more information, see Chapter 2, “Defining Channel-Attached Printers to MVS” on page 7.
- **Operator controls:** PSF processes spooled output from JES before it sends a data stream to a printer. The operator control statements are presented in the operations guides for JES2 and JES2. For titles and order numbers, see Appendix K, “Related Publications” on page 345.
- **Output record access:** PSF has an installation exit routine that can add, change, or suppress output records.
- **Print complete notification (NOTIFY):** In certain levels of MVS/ESA, the user can optionally request that the system supply a notification message when printing is completed. This function is not supported for other MVS systems.
- **Resource management:** PSF makes sure that the resources needed for printing a data set (form definitions, page definitions, fonts, page segments, overlays, and object containers) are available. For descriptions of these resources, refer to *Print Services Facility/MVS: Application Programming Guide*.
- **Restarting from a checkpoint:** When requested, PSF takes checkpoints at a specified interval and transmits a request to JES to record checkpoint data. A smaller checkpoint interval causes more checkpoints to be taken during the processing of a data set, and thus reduces the amount duplicated when printing is restarted from a checkpoint.
- **Separator pages:** PSF prints separator pages that show the start and end of a print job. Separator pages can be replaced, changed, or deleted through the installation exit routine and through the printer configuration.
- **System Management Facility (SMF) Type-6 Records:** PSF writes the SMF type-6 record and has an installation exit routine that can change SMF type-6 records.
- **User libraries:** Under some levels of MVS/ESA, PSF enables users to specify libraries containing resources (fonts, page segments, overlays, page definitions, and form definitions) for use when the print data set is processed.
- **Virtual storage constraint relief:** To allow more active printers for each address space, PSF/MVS Version 2 obtains storage above the 16-megabyte line in 31-bit addressing mode, except where system constraints force PSF to obtain the storage below the 16-megabyte line.

A small part of the virtual storage constraint relief is related to the functional subsystem interface (FSI) between JES and PSF. PSF/MVS allows the FSI to operate above the 16-megabyte line in 31-bit addressing mode, where it has additional storage space. If, however, any PSF exit is run in the 24-bit addressing mode, PSF forces the FSI to run below the 16-megabyte line. For more information, see Chapter 17, “Using Installation Exits” on page 163.

Printers Supported by PSF

The information in this publication applies to IBM printers supported by PSF; not all models of those printers are supported. You should be familiar with the specific printers installed on your system. For the most current list of printers and printer models supported, and information about PSF-supported printers, refer to *Advanced Function Presentation: Printer Summary* or to *Advanced Function Presentation: Printer Information*.

If you are installing a 300-pel printer, and you do not have a 300-pel font library, consider using APSRCF30 to migrate your 240-pel fonts to the 300-pel format. For more information, see Appendix F, “Font Conversion Program, 240-pel to 300-pel (APSRCF30).”

Note: The microfilm device can operate as either a 240-pel device or a 300-pel device. For more information, see Chapter 21, “Supporting AFP Output on Microfilm.”

If you plan to use the PSF print-labeling function, refer to *Print Services Facility: Security Guide*. The publication you are now reading does not contain specific information about the print-labeling function.

Printers Attached through an Intermediate Device

PSF/MVS supports printing through the following programs that run on intermediate devices such as personal computers:

- Distributed Print Function (DPF) of PSF/2, running on a personal computer
- PSF Direct function of PSF/2, running on a personal computer
- PSF Direct function of PSF for AIX, running on an IBM RS/6000

If you are using an intermediate device, you should be familiar with the publications listed in Appendix K, “Related Publications.”

The microfilm device is not supported by intermediate devices for printing to microfilm. For a list of printers supported by intermediate devices, refer to *Advanced Function Presentation: Printer Information*.

With DPF you can store (cache) PSF resources in the fixed-disk storage of the personal computer; the stored resources can then be used by a printer that is attached to the intermediate device. The PSF Direct function does not store resources in the intermediate device.

Chapter 2. Defining Channel-Attached Printers to MVS

This chapter describes the prerequisites, methods, and steps for defining host-connected, channel-attached printers and microfilm devices to MVS.

Prerequisites for Defining Channel-Attached Printers to MVS

When you are installing a basic version of PSF for a particular set of printers, you must define the printers to MVS for the specific configuration of hardware being used. For this task you must know what printers are attached, how they are attached, and what version of MVS you are using.

Your hardware configuration can contain 3800 printers, 3820 printers, or channel-attached printers and microfilm devices of the AFP1 family; the definition is slightly different for each.

Your printers can be attached as follows:

- Host-connected, channel-attached printers:
  - Channel-attached by linking directly to the host
  - Channel-attached with switches providing multiple paths to the host
- SNA-attached channel printers:
  - Channel-attached to an intermediate device, to communicate with PSF through SNA

This chapter describes only host-connected, channel-attached printers. For information about defining SNA-attached printers, see Chapter 3, “Establishing Network Nodes for SNA-Attached Printers.”

Selecting the Method

To define host-connected, channel-attached printers, use either an MVS configuration program (MVSCP) or a hardware configuration definition (HCD). Use MVSCP rather than SYSGEN or IOGEN to define the I/O configuration. MVSCP accepts the same IODEVICE statements as does the SYSGEN or IOGEN process.

- If you are using a version earlier than MVS 4.1.0, use MVSCP to specify the device. For the title and order number of the configuration program user's guide for MVS/ESA, see Appendix K, “Related Publications.”
- If you are using MVS Version 4, you can use MVSCP or HCD to specify the device. MVS Version 5, however, does not support MVSCP; so HCD has to be used. See “Defining Channel-Attached Printers with HCD” on page 13. For more information on how to configure your printers by the HCD method (MVS/SP 4.1.0 and above), refer to the following publications:
  - *MVS/ESA Migration Planning: Dynamic I/O Configuration* or *OS/390 HCD Planning*.
  - *MVS/ESA Hardware Configuration: Using the Dialog* or *OS/390 HCD User's Guide*.

For information about how to migrate existing MVSCP input statements to an HCD IODF, refer to *MVS/ESA Hardware Configuration: Using the Dialog* or *OS/390 HCD User's Guide*.

Most channel-attached printers, when directly linked to the host system through an IBM System/390 channel, do not use SNA protocol. The only exceptions are 3820 channel-attached printers. Figure 1 shows an example of a channel-attached printer configuration that is directly linked to the host system.

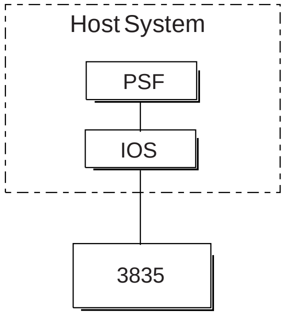


Figure 1. Example of a Channel-Attached Printer Configuration

Using Switches with Host-Connected, Channel-Attached Printers

By using the optional two-channel switch, you can define a second channel interface for a printer attached to a single host, or define the second channel interface to another host. PSF supports this configuration only in manual mode (operator control). Using this switch can provide manual printer sharing or increased reliability if a channel fails. You must perform an I/O configuration to define all the channel paths.

Using an Intermediate Device with Channel-Attached Printers

When you connect to channel-attached printers through an intermediate device, PSF uses Systems Network Architecture (SNA) protocol in communicating with PSF/2 or PSF for AIX. PSF/2 and PSF for AIX communicate with printers by means of System/390 channel commands. When setting up communications between PSF and an intermediate device, use Chapter 3, “Establishing Network Nodes for SNA-Attached Printers” and the publications for the intermediate device, listed in Appendix K, “Related Publications.”

Using ESCON Channel-Attached Printers

The Enterprise Systems Connection (ESCON) attachment improves data rate, physical path length, space requirements, and overall system connectivity.

With an ESCON card in the printer, you can use an ESCON channel, rather than the System/390 parallel channel attachment, to attach printers. The 3900 Duplex printer can be attached by this method; additional printers will be attached by this method as they become available.

The ESCON attachment also makes it possible to exploit the dynamic pathing function. Because dynamic pathing uses multiple paths, it improves reliability and performance. For information about how to define multiple paths to the device, refer to the hardware manual for your printer.

For ESCON channel-attached printers, PSF increases the minimum value of the BUFNO parameter to 9. This provides two 16K buffers for I/O; for better performance, however, consider increasing BUFNO.

PSF/MVS supports an ESCON channel-attached printer only when manual (operator-controlled) printer-sharing is used; PSF does not support dynamic printer-sharing.

Note: The microfilm device can be ESCON-attached for printing to microfilm.

When running MVS second-level on VM, make sure that APAR VM58987 has been installed.

Defining IBM 3800 Printers with MVSCP

The configuration definition shown in Figure 2 consists of IODEVICE and CNTLUNIT statements for host-connected, channel-attached printers. To define a 3800 printer to run with a version earlier than MVS 4.1.0, use the MVSCP, or migrate the existing MVSCP deck to an HCD IODF. You can modify the example in the figure.

```
CNTLUNIT CUNUMBR=380,
      PATH=(03),
      UNITADD=(80),
      PROTOCL=S,
      SHARED=N,
      UNIT=3800
IODEVICE ADDRESS=(380)
      CUNUMBR=380,
      UNIT=3800,
      MODEL=3,               Use 3 for Models 3
                             or 6; 8 for Model 8.
      FEATURE=BURSTER,       These are optional
                             parameters.
      OFFLINE=NO,            You can use YES.
      PATH=chpid,
      STADET=Y,              You can use NO.
      TIMEOUT=Y,             You can use NO.
      UNITADD=80             Same as UNITADD on
                             CNTLUNIT
```

Figure 2. Sample JCL for Defining a 3800 Model 3 Printer. Use this sample for all MVS levels that support MVSCP.

For details on defining an I/O configuration, refer to the latest version of *MVS Configuration Program Guide and Reference*. To define a 3800 printer to run with MVS Version 4, you can use MVSCP or HCD. MVS Version 5, however, does not support MVSCP; so HCD has to be used. See “Defining Channel-Attached Printers with HCD” on page 13.

Defining IBM Channel-Attached 3820 Page Printers with MVSCP

The configuration definition shown in Figure 3 consists of IODEVICE and CNTLUNIT statements for host-connected, channel-attached printers. For a host-connected, channel-attached 3820 in the MVSCP, specify *3791L*, as is done in the figure. For details on defining an I/O configuration, refer to the latest version of the MVS/ESA *MVS Configuration Program Guide and Reference*.

```
CNTLUNIT CUNUMBR=3AO,
      PATH=(03),
      UNITADD=(A0),
      PROTOCL=D,
      SHARED=N,
      UNIT=3791L

IODEVICE ADDRESS=3AO,
      CUNUMBR=3AO,
      TIMEOUT=Y,
      UNIT=3791L
```

Figure 3. Sample JCL for Defining a Host-Connected, Channel-Attached 3820 Printer. Use this sample for all MVS levels that support MVSCP.

To define a 3820 printer to run with an MVS version earlier than MVS 4.1.0, use MVSCP. To define a 3820 host-connected, channel-attached printer to run with MVS Version 4, you can use MVSCP or HCD. MVS Version 5, however, does not support MVSCP; so HCD has to be used. See “Defining Channel-Attached Printers with HCD” on page 13.

Note: For the 3820 printer, a local major node, described in “Defining Local Major Nodes” on page 28, has to be defined to VTAM.

Defining Parallel Channel-Attached Printers (AFP1 Family) with MVSCP

The configuration definition shown in Figure 4 on page 11 consists of CNTLUNIT and IODEVICE statements for channel-attached printers that use a version earlier than MVS 4.1.0. PSF/MVS 2.2.0 or later supports the generic device type AFP1 for printers, which is specified on the UNIT keyword in much the same manner as the generic device type of “3800” that is specified for 3800 printers. To define a channel-attached printer to run with a version earlier than MVS 4.1.0, use the example in the figure as a model.

```
CNTLUNIT CUNUMBR=392,
      PATH=03,              Specifying channel path
      PROTOCL=D, S, or S4   S or S4 is recommended[1]
      SHARED=N,
      UNIT=AFP1,
      UNITADD=((921))

IODEVICE ADDRESS=392,
      CUNUMBR=392,          Same as CUNUMBR on CNTLUNIT
      MODEL=0,              This is optional
      TIMEOUT=Y,
      UNIT=AFP1
```

Figure 4. Sample JCL for Defining a Channel-Attached Printer. Use this sample for all MVS levels that support MVSCP.

For details on defining an I/O configuration, refer to the latest version of *MVS Configuration Program Guide and Reference*.

To define a channel-attached printer or a device for printing to microfilm to run with MVS Version 4, you can MVSCP or HCD. MVS Version 5, however, no longer supports MVSCP; so HCD has to be used. See “Defining Channel-Attached Printers with HCD” on page 13.

Defining ESCON Channel-Attached Printers (AFP1 Family) with MVSCP

The configuration definition in Figure 5 on page 12 consists of CHPID, CNTLUNIT, and IODEVICE statements for ESCON channel-attached printers, including microfilm devices, that use a version of MVS earlier than MVS/ESA 4.1.0. An ESCON channel has an ESCON channel-to-control unit I/O interface that uses fiber optic cables as a transmission medium. (Compare to a parallel channel.) PSF supports the generic device type “AFP1,” which is specified on the UNIT keyword in much the same way as the generic device type of “3800” is specified for 3800 printers. This example assumes that an ESCON director (9032, 9033) is *not* used in the configuration and that the printer is configured with one device per control unit. For information on defining with an ESCON director, refer to *IOCP Configuration Program User's Guide and ESCON Channel-to-Channel Reference*, GC38-0401.

[1] The printer must be configured to match the protocol specified on this parameter.

```
CHPID PATH=03,TYPE=CNC   (CNC indicates an ESCON
                         channel)

CNTLUNIT CUNUMBR=392,
   PATH=03,
   UNIT=AFP1,            (PROTOCL, SHARED parameters
                         not needed. LINK parameter
                         not coded because 9032/9033
                         not being used)
   UNITADD=((92,1))

IODEVICE ADDRESS=392,
   CUNUMBR=392,          (Same as CUNUMBR on
                         CNTLUNIT)
   MODEL=0,              (This is optional)
   UNIT=AFP1             (TIMEOUT parameter ignored)
```

Figure 5. Sample JCL for Defining a Host-Connected ESCON Channel-Attached Printer. Configured with one device per control unit.

For a printer configured in dual simplex mode (two devices behind one control unit), MVSCP statements such as those shown in Figure 6 can be used. This example assumes that no ESCON director is used.

```
CHPID PATH=03,TYPE=CNC
CHPID PATH=04,TYPE=CNC       (This CHPID statement is
                             optional, but is added for
                             performance reasons)

CNTLUNIT PATH=(03,04),
      UNIT=AFP1,
      CUNUMBR=392,
      UNITADD=((92),(93))    (No LINK or PROTOCL
                             statement needed)

IODEVICE CUNUMBR=392,
      ADDRESS=392,
      UNIT=AFP1,
      MODEL=0

IODEVICE CUNUMBR=392,
      ADDRESS=393,
      UNIT=AFP1,
      MODEL=0
```

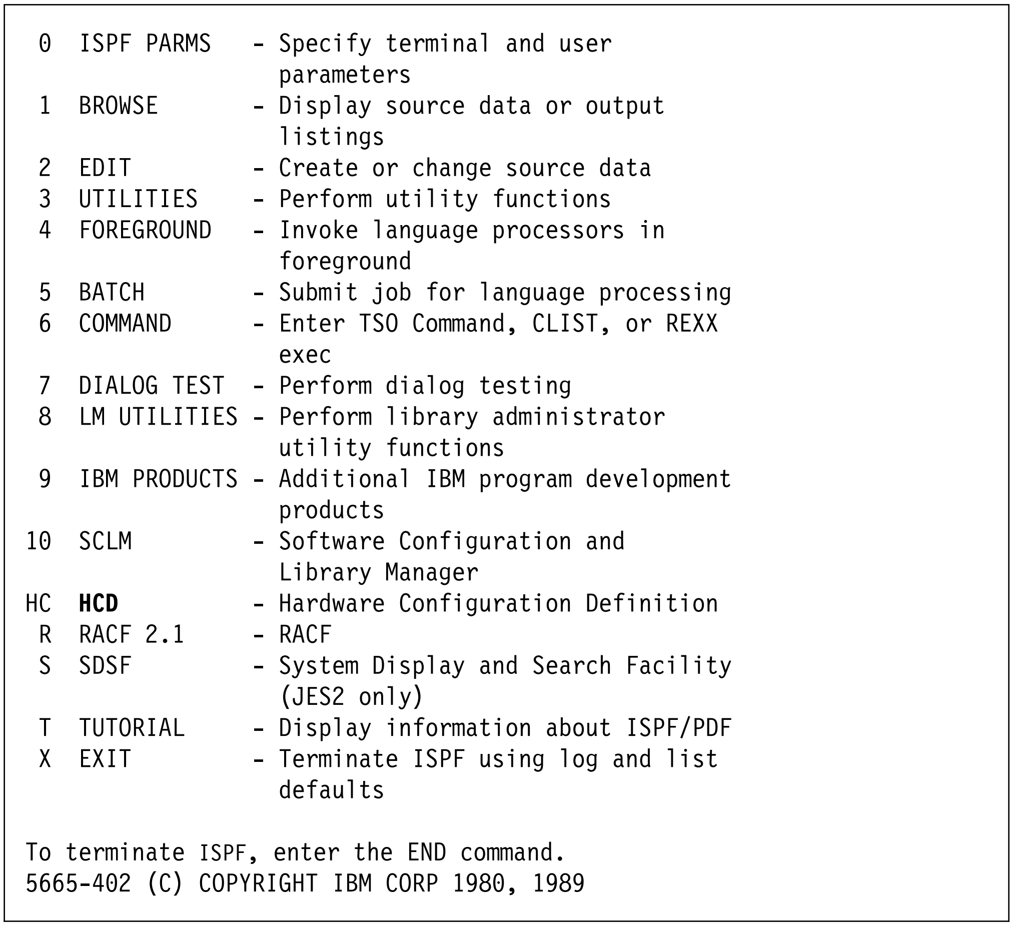
Figure 6. Sample JCL for Defining a Host-Connected ESCON Channel-Attached Printer. Configured in dual simplex mode.

Defining Channel-Attached Printers with HCD

MVS 4.1.0 supports HCD without dynamic reconfiguration management (DRM) capabilities. (DRM enables you to dynamically add AFP1 and 3820 channel-attached printers to the I/O configuration, delete them from the configuration, or change them, provided that the system is MVS/ESA SP 4.2.0 or above, or that you are running JES2 4.2.0 or above.) If you want HCD with DRM capabilities, MVS 4.2.0 and all prerequisites must be installed.

After HCD is installed, use Interactive System Productivity Facility (ISPF) to access it. HCD is menu-driven and has online help screens.

Figure 7 shows an example of an ISPF menu with HCD installed.



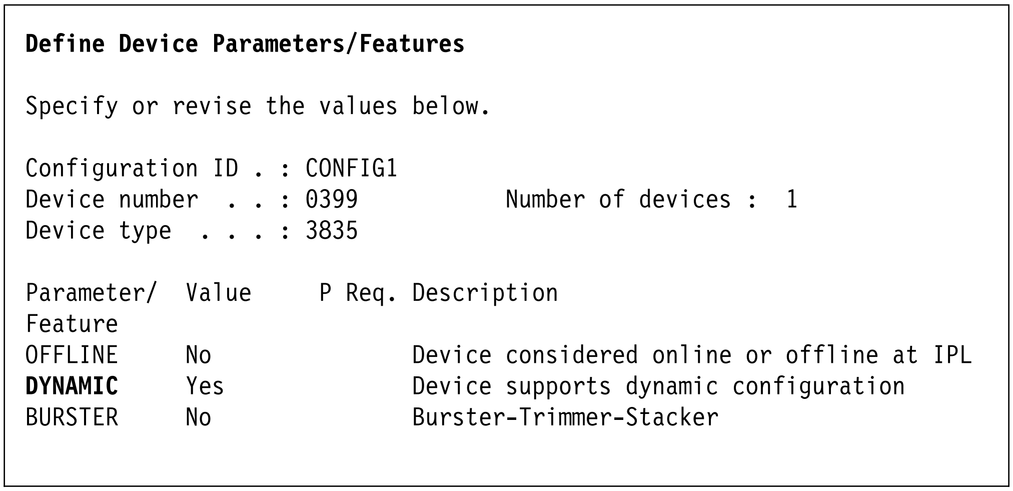
```
 0  ISPF PARMS   - Specify terminal and user
                   parameters
 1  BROWSE       - Display source data or output
                   listings
 2  EDIT         - Create or change source data
 3  UTILITIES    - Perform utility functions
 4  FOREGROUND   - Invoke language processors in
                   foreground
 5  BATCH        - Submit job for language processing
 6  COMMAND      - Enter TSO Command, CLIST, or REXX
                   exec
 7  DIALOG TEST  - Perform dialog testing
 8  LM UTILITIES - Perform library administrator
                   utility functions
 9  IBM PRODUCTS - Additional IBM program development
                   products
10  SCLM         - Software Configuration and
                   Library Manager
HC  HCD          - Hardware Configuration Definition
 R  RACF 2.1     - RACF
 S  SDSF         - System Display and Search Facility
                   (JES2 only)
 T  TUTORIAL     - Display information about ISPF/PDF
 X  EXIT         - Terminate ISPF using log and list
                   defaults

To terminate ISPF, enter the END command.
5665-402 (C) COPYRIGHT IBM CORP 1980, 1989
```

Figure 7. Interactive System Productivity Facility Primary Option Menu. Use this sample if your version is MVS 4.1.0 or above, with HCD installed.

If you choose the HCD option, the main menu for HCD will be displayed.

Figure 8 shows an example of an HCD Define Device Parameters/Features screen.



```
Define Device Parameters/Features

Specify or revise the values below.

Configuration ID . : CONFIG1
Device number  . . : 0399          Number of devices :  1
Device type  . . . : 3835

Parameter/  Value     P Req. Description
Feature
OFFLINE     No               Device considered online or offline at IPL
DYNAMIC     Yes              Device supports dynamic configuration
BURSTER     No               Burster-Trimmer-Stacker
```

Figure 8. HCD Define Device Parameters/Features Screen

In the `Device type` option, for channel-attached AFP1 printers, you can specify either AFP1 or the IBM device type. This example is for the 3835 printer.

Notice that DYNAMIC is one of the options shown in Figure 8. The DYNAMIC option is displayed only if MVS 4.2.0 or above is being used.

For more information about how to use HCD, refer to *MVS/ESA Hardware Configuration Definition: Using the Dialog* or *OS/390 HCD User's Guide*

In defining channel-attached *3820* printers, specify 3820 for the device type and the control unit type. When adding an ESCON channel-attached printer, you must attach it to an ESCON channel (type=CNC).

You can migrate your existing MVSCP deck to an HCD I/O definition file (IODF). For certain device types, you may need to make changes to the MVSCP deck. The following considerations apply when you migrate from an MVSCP deck to an IODF, using HCD, for AFP1 and channel-attached 3820 printers:

1. In the past, you coded *UNIT=3791L* on the CNTLUNIT and IODEVICE statements when defining a channel-attached 3820 printer. Now, when using HCD, change the unit parameter on the CNTLUNIT and IODEVICE statements from *3791L* to *3820*.
2. Also for *3791L*, if you have specified the model parameter on the CNTLUNIT or IODEVICE statement, delete it. The 3820 printer has no model number and cannot be specified when HCD is used.

   Note: For the 3820 printer, a local major node, described in “Defining Local Major Nodes” on page 28, has to be defined to VTAM.

Following are the steps for adding an AFP1 or 3820 channel-attached printer to an I/O configuration by using HCD and DRM and then using PSF to drive the printer in the MVS 4.2.0 environment. Use these steps if you are using HCD and DRM in adding an AFP1 or 3820 channel-attached printer to an I/O configuration. Details not described here can be found in the publications for HCD and DRM; for titles and order numbers, see Appendix K, “Related Publications.”

- Define the printer to your I/O configuration, using HCD.
- Activate the I/O configuration, using the MVS ACTIVATE command.
- If the JES2INIT deck does not already contain a printer definition for the printer that is to be started, define one, using the JES2 $ADD command.

  If a printer definition does exist but needs to be updated, use the JES2 $T command.

  For JES3, if the INIT deck needs to be updated (whether to add a new printer definition or to update an existing one), update the INIT deck and START JES3.
- Vary the printer online, using the MVS VARY command.

  You can create a new PRINTDEV statement in a new startup procedure, or you can use an existing PRINTDEV statement (APAR OY48714). To update the current startup procedure, stop the WRITER PROC:

  1. For JES2, issue the $P command for all printers in the WRITER PROC. For JES3, issue the *V command.
  2. Issue the MVS CANCEL command.
- Start the printer as you usually do.

Note: Be sure to update the JES INIT deck before you start PSF; otherwise you will probably encounter a PSF abend 024 with a return code of 203 or 104.

If you want to remove a printer from the I/O configuration being used by PSF, and only one printer is active in the WRITER PROC, cancel the WRITER PROC to change the device to an unallocated state. (If several printers are active in the WRITER PROC, issue the JES2 $P command or the JES3 *V command, followed by a *C command, to change the device to an unallocated mode.) Use the VARY command to vary the device offline. Use the MVS ACTIVATE command to remove the printer from the I/O configuration.

[2] The PSF startup procedure.

Chapter 3. Establishing Network Nodes for SNA-Attached Printers

This chapter describes sample network configurations for SNA-attached printers, and provides specific parameters for setting up a printer network. Microfilm devices are not described in this chapter, because they cannot be configured as SNA-attached printers.

For information on defining more complex networks, refer to the publications listed under “VTAM and NCP” on page 349 in Appendix K, “Related Publications.” For information on network configuration for AFCCU printers (3130, 3160, and 3935), refer to *Attachment Configuration Handbook for Printers That Use the Advanced Function Common Control Unit*.

Do not use this chapter when configuring printers supported through the Distributed Print Function (DPF) of PSF/2 or the PSF Direct function of PSF/2 or PSF for AIX. Instead, refer to the PSF/2 publications listed on page 350 and the PSF for AIX publications listed on page 350. The graphics present DPF and PSF Direct together, because they work the same way in concept even though they differ in details of configuration.

Overview of a Network

Define the SNA-attached printers (or the devices to which they are attached) to Advanced Communication Function/Virtual Telecommunications Access Method (VTAM) as either local-attached or communication-attached.

Local-Attached Devices

Figure 9 shows three sample configurations for local-attached devices. PSF sends data through VTAM to a printer, either directly or by way of a control unit. The control unit then sends the data to a printer, either directly or by way of PSF/2 or PSF for AIX.

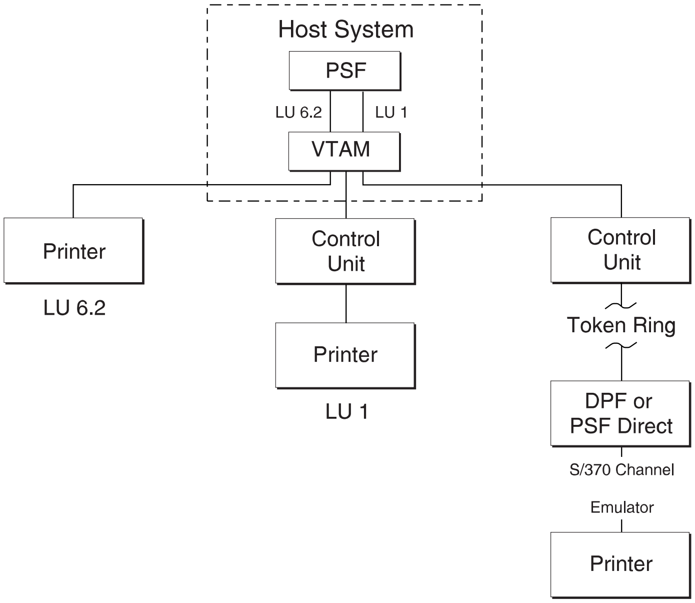


Figure 9. Sample PSF Configurations for Local-Attached Devices

To define devices that are local-attached (printer or control units), you must use either an MVS configuration program (MVSCP) or a hardware configuration definition (HCD). The choice depends on the version of MVS being used, as follows:

- If you are using a version of MVS lower than 4.1.0, you must use MVSCP.
- If you are using a version of MVS that is 4.1.0 or higher, but lower than 5.1.0, you can use either MVSCP or HCD.
- If you are using version 5.1.0 or higher of MVS, you must use HCD.

Communication-Attached Devices

Figure 10 shows three sample configurations for communication-attached devices. Here, VTAM sends data from PSF through a communication controller that runs the Network Control Program (NCP). Under NCP control, data is then sent to the printer either directly, by way of a control unit, or by way of PSF/2 or PSF for AIX.

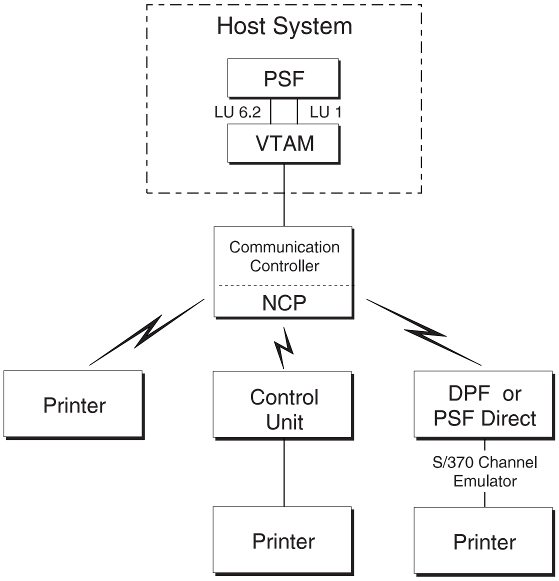


Figure 10. Sample PSF Configurations for Communication-Attached Devices

Communication can be established between the controller and the remote station by using either a switched line or a nonswitched line. (With a switched line, the connection is established by dialing; with a nonswitched line, no dialing is needed.)

Figure 11 shows four possible communication attachments with DPF and PSF Direct:

- Synchronous Data Link Control (SDLC), attached through a communication controller
- Token-ring, attached through a communication controller token-ring subsystem
- Token-ring, attached through a remote communication controller token-ring subsystem
- Token-ring, attached through a remote control unit token-ring gateway

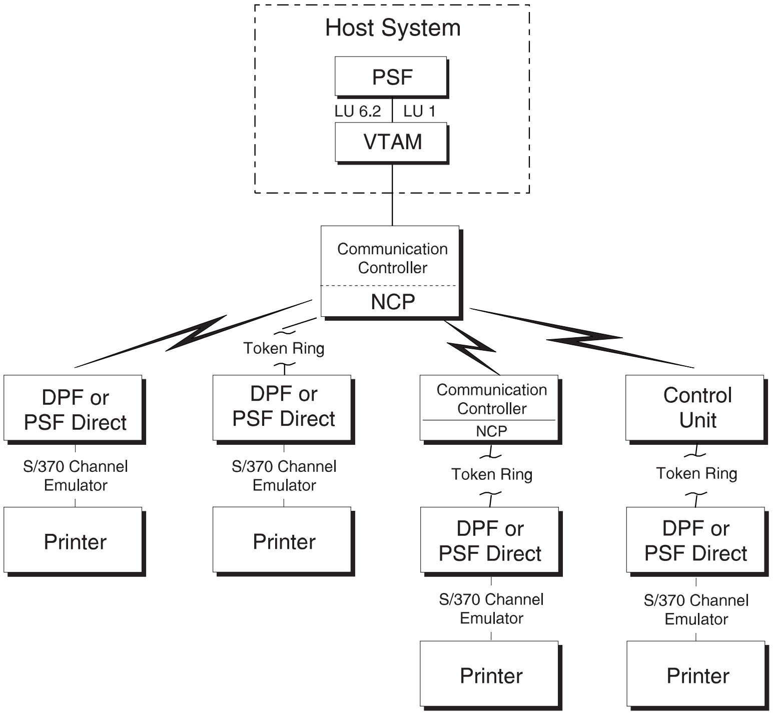


Figure 11. Sample PSF Configurations for Communication Attachment to DPF or PSF Direct

VTAM Definitions for PSF Network Nodes

Before using an SNA-attached printer, identify the following:

- PSF application program nodes
- Logon-mode table entries for each printer model
- Printer-device nodes
- Communication lines for NCP, for communication-attached devices

For communication-attached devices, you must perform an NCP generation, which includes a definition of the communication lines to be connected to each printer. For more information, see “NCP Generation” on page 34.

Note: The definition samples and explanations in this section are not intended to be a complete description of the statements and operands. For more information, refer to your publications for VTAM.

SNA Devices, Attachment Mode, and Requirements

Table 1 on page 21 gives an overall picture of SNA devices, the mode of attachment, and the details about the requirements.

Table 1. SNA Devices: Requirements

SNA DEVICES	REQUIREMENTS								
						Define NCP Major Node			
SNA devices are attached in one of the following ways, each of which requires definitions for NCP.	Define a Local-Attached Device (page 17)	Define a PSF application node (page 23)	Define the logon-mode table entry (page 24)	Define local major nodes (page 28)	Define switched major nodes (page 30)	Switched line (page 35)	Non-switched line (page 36)	Multi-point non-switched line	Token-ring sub-system connection (page 42)
Local-Attached	X	X	X	X					
Communication-Attached: Token Ring through									
• Communication Controller Token-Ring Subsystem		X	X		X				X
• Remote Communication Controller Token-Ring Subsystem		X	X		X		X		X
• Remote Control Unit Token-Ring Gateway		X	X					X	
Communication-Attached: Switched Line		X	X		X	X			
Communication-Attached: Nonswitched Line		X	X				X		

With PSF, SNA devices are attached in one of the following ways, each of which requires network definitions for NCP:

Local-attached devices

- PSF application node. See “Defining a PSF Application Node” on page 23.
- Logon-mode table entry. See “Defining the Logon-Mode Table Entry” on page 24.
- Physical unit (PU) and logical unit (LU) statements in a local major node. See “Defining Local Major Nodes” on page 28.

Make sure that the I/O device is included in the MVSCP or the HCD. (The I/O device can be either a printer or a control unit.) See “Defining Local Major Nodes” on page 28.

Communication-attached devices (switched line)

- PSF application node. See “Defining a PSF Application Node” on page 23.

- Logon-mode table entry. See “Defining the Logon-Mode Table Entry” on page 24.
- PU, PATH, and LU statements in a switched major node. See “Defining Switched Major Nodes” on page 30.
- LINE and PU statements in a switched line GROUP under an NCP major node. An NCP generation is required. See “Switched Line” on page 35.

Communication-attached devices (nonswitched line)

- PSF application node; see “Defining a PSF Application Node” on page 23.
- Logon-mode table entry; see “Defining the Logon-Mode Table Entry” on page 24.
- LINE, SERVICE (optional), PU, and LU statements in a nonswitched line GROUP under an NCP major node. An NCP generation is required. See “Nonswitched Line” on page 36.

Communication-attached devices (token ring)

Through a communication controller token-ring subsystem:

- PSF application node. See “Defining a PSF Application Node” on page 23.
- Logon-mode table entry. See “Defining the Logon-Mode Table Entry” on page 24.
- PU, PATH, and LU statements in a switched major node. See “Defining Switched Major Nodes” on page 30.
- LINE, PU, and LU statements in a token-ring GROUP under an NCP major node. An NCP generation is required. See “Token-Ring Subsystem Connection” on page 42.

Through a remote communication controller token-ring subsystem:

- PSF application node. See “Defining a PSF Application Node” on page 23.
- Logon-mode table entry. See “Defining the Logon-Mode Table Entry” on page 24.
- PU, PATH, and LU statements in a switched major node. See “Defining Switched Major Nodes” on page 30.
- LINE and PU statements in a nonswitched line GROUP under two NCP major nodes. An NCP generation is required. See “Nonswitched Line” on page 36.
- LINE, PU, and LU statements in a token-ring GROUP under an NCP major node. An NCP generation is required. See “Token-Ring Subsystem Connection” on page 42.

Through a remote control unit token-ring gateway:

- PSF application node. See “Defining a PSF Application Node” on page 23.
- Logon-mode table entry. See “Defining the Logon-Mode Table Entry” on page 24.
- LINE, SERVICE, PU, and LU statements in a multipoint nonswitched line GROUP under an NCP major node. An NCP generation is required. See “Multipoint Nonswitched Line” on page 39.

VTAM Syntax Rules

VTAM definition statements and NCP macroinstructions must follow these syntax rules:

- The name field symbolically identifies the macroinstruction, definition statement, or minor node.
- If a symbolic name is specified in the field, the name must contain from 1 to 8 characters in the following format:
  - Character 1: A–Z, or the characters @, #, or $
  - Characters 2–8: A–Z, 0–9, or the characters @, #, or $
- The name must begin in the first position of either the definition statement or the macroinstruction, and must be followed by one or more blanks.
- If the name is optional, it should be coded, because VTAM uses it in operator messages that refer to the defined resource.
- A statement that does not fit on one line can be continued on the next line by using a continuation character (any nonblank character in column 72) and continuing the statement on column 16 of the next line. If you do not add a continuation character to a VTAM statement that is continued to a second line, you will not receive an error message, but processing will stop at the end of the first line and will be unable to continue with the second line of the statement. The printer, therefore, will be incorrectly initialized.

These conventions apply to the definition-statement name fields described later in this chapter.

VTAM statements and NCP macroinstructions follow a convention called sift-down. By this convention, an operand can be coded on a higher-level node (for example, PU is a higher-level node than LU), so that the operand does not need to be recorded on each of the lower-level nodes for which the same value applies. Many operands do not affect the higher-level nodes, but are placed there to take advantage of sifting. A sifted value can be overridden by coding the same operand, with the new value, on a lower-level node. For more information on the sift-down effect, refer to *VTAM Resource Definition Reference*. The network configuration examples that follow in this chapter do not follow this convention.

Note: Some statement parameters and their values differ, depending on which LU protocol is used between VTAM and the device. For example, for a 3820 printer the parameters and values associated with LU type 6.2 are used; for a 3812 printer, the parameters and values associated with LU type 1 are used.

Defining a PSF Application Node

An APPL statement defines a PSF application program node, called a logical unit (LU), within the network domain of a VTAM supervisor services control point. See Figure 12 on page 24.

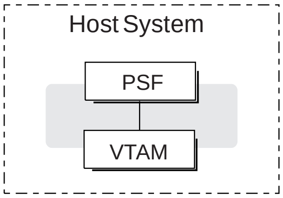


Figure 12. PSF Application Node. One logical unit (LU) is shown. The APPL statement defines an LU.

This APPL statement also defines some of the communication capabilities of the application program. You should also define an APPL statement for each PSF functional subsystem application (FSA) for each printer, and you must define at least one FSA per printer. The format in Figure 13 is used.

```
name          VBUILD TYPE=APPL
PSFFSA1       APPL   AUTH=ACQ,EAS=1,SONSCIP=YES
```

Figure 13. Sample APPL Statement. Commonly found in SYS1.VTAMLST.

Note: Do not specify APPC=YES on the APPL statement.

APPL

Defines a PSF application program node and some of the communications capabilities of the application program.

Note: The name of the application program node is given (PSFFSA1 in this example) in the name field, unless the ACBNAME parameter is specified. If the ACBNAME parameter is specified, it defines the name of the application program node. For more information, refer to your publications for VTAM.

The application program node must also be specified in the APPLID parameter of the PRINTDEV statement in the PSF startup procedure. The name of the APPL statement must be unique within the network.

AUTH

Allows PSF to use the SIMLOGON macroinstruction to initiate a session with the printer.

Do not specify AUTH=PPO on the APPL statement.

EAS

Specifies the number of concurrent sessions a PSF FSA can have with other logical units (LU-LU sessions). Specify EAS=1.

SONSCIP

Specifies that PSF is to receive UNBIND request/response units (RU) in its SCIP exit routine when PSF is acting as the primary end of a failing session.

Defining the Logon-Mode Table Entry

The logon-mode table must contain an entry (MODEENT statement) for each attachment protocol.

The logon-mode table can be one supplied by IBM, or it can be locally defined. If the table is locally defined, it must be identified by the MODETAB operand in the LU statement, and that table is searched first for the specified entry.

Local-Attached Entries (Excluding PSF/2 and PSF for AIX)

```
For an LU type 6.2 device

IBM3820C MODEENT LOGMODE=IBM3820C,FMPROF=X'13',TSPROF=X'07',
                 PRIPROT=X'B0',SECPROT=X'B0',
                 COMPROT=X'D0B1',RUSIZES=X'85C7',
                 PSERVIC=X'060200000000000000002000',
                 PSNDPAC=X'03',SRCVPAC=X'03',
                 SSNDPAC=X'00'

For an LU type 1 device

IBM3812C MODEENT LOGMODE=IBM3812C,FMPROF=X'03',TSPROF=X'03',
                 PRIPROT=X'B1',SECPROT=X'B0',
                 COMPROT=X'7080',RUSIZES=X'85C7',
                 PSERVIC=X'014000010000000001000000',
                 PSNDPAC=X'02',SRCVPAC=X'02',
                 SSNDPAC=X'00'
```

Figure 14. Sample Logon-Mode Table Entry for Local-Attached Devices. Continuation characters in column 72 are not shown. These statements are commonly found in SYS1.VTAMLIB(MODETAB1).

The LU type 1 example can be used for all the LU type 1 printers that PSF supports.

Communication-Attached Entries (Excluding PSF/2, PSF for AIX, and Token Ring):

```
For an LU type 6.2 device

IBM3820  MODEENT LOGMODE=IBM3820,FMPROF=X'13',TSPROF=X'07',
               PRIPROT=X'B0',SECPROT=X'B0',
               COMPROT=X'D0B1',RUSIZES=X'8585',
               PSERVIC=X'060200000000000000002000',
               PSNDPAC=X'03',SRCVPAC=X'03',
               SSNDPAC=X'00'

For an LU type 1 device

IBM3812  MODEENT LOGMODE=IBM3812,FMPROF=X'03',TSPROF=X'03',
               PRIPROT=X'B1',SECPROT=X'B0',
               COMPROT=X'7080',RUSIZES=X'8585',
               PSERVIC=X'014000010000000001000000',
               PSNDPAC=X'03',SRCVPAC=X'03',
               SSNDPAC=X'00'
```

Figure 15. Sample Logon-Mode Table Entry for Communication-Attached Devices. Continuation characters in column 72 are not shown. These statements are commonly found in SYS1.VTAMLIB(MODETAB1).

For VTAM definitions for DPF, refer to the PSF/2 and PSF for AIX publications listed in Appendix K, “Related Publications.”

```
For a token-ring device

IBM3820R  MODEENT LOGMODE=IBM3820R,FMPROF=X'13',TSPROF=X'07',
                PRIPROT=X'B0',SECPROT=X'B0',
                COMPROT=X'D0B1',RUSIZES=X'mnmn',
                PSERVIC=X'060200000000000000002000',
                PSNDPAC=X'xx',SRCVPAC=X'yy',
                SSNDPAC=X'00'
```

Figure 16. Sample Logon-Mode Table Entry for Token-Ring Devices. Continuation characters in column 72 are not shown in the above statements. These statements are commonly found in SYS1.VTAMLIB(MODETAB1).

Values for RUSIZES, PSNDPAC, and SRCVPAC depend upon the configuration. Recommended values are listed in the following descriptions of these parameters.

The parameters for the sample MODEENT statements follow. (For information not provided in the following descriptions, refer to your publications for VTAM.)

MODEENT

Creates an entry in the logon-mode table. The statement name is optional and has no function in the specification of the logon-mode table.

COMPROT=*hex value*

The value for an LU type 6.2 device is X'D0B1'. The value for an LU type 1 device is X'7080'.

FMPROF=*hex value*

The value for an LU type 6.2 device is X'13'. The value for an LU type 1 device is X'03'.

LOGMODE

Specifies the logon-mode name to be used as a key for the session operands in this table entry. If the same name appears twice in the table, the first occurrence is used. Alternatively, the name identifies this entry in the logon-mode table, either with the DLOGMOD operand of the LU statement for a printer or with the LOGMODE option of the PRINTDEV statement in the PSF startup procedure. If LOGMODE is omitted, eight blanks are used.

PRIPROT=*hex value*

The value for an LU type 6.2 device is X'B0'. The value for an LU type 1 device is X'B1'.

PSERVIC=*hex value*

The value for an LU type 6.2 device is X'0602 0000 0000 0000 0000 2000'. The value for an LU type 1 device is X'0140 0001 0000 0000 0100 0000'.

PSNDPAC=*hex value*

Suggested values are:

- For an LU type 6.2 device—X'03'
- For an LU type 1.0 device—X'nn'

Note: Pacing is an important performance parameter, for it determines how many RUs the host can send before a pacing response is required. This value is device-dependent. For more information on setting the pacing value, refer to the publications for your device and for VTAM.

RUSIZES=*X'mnmn'*
Specifies the maximum length (request units, RUs) in bytes that can be sent. The 2 leftmost hexadecimal digits apply to data sent by the secondary LU (printer). The 2 rightmost hexadecimal digits apply to data sent by the primary LU (PSF). In each pair, the first digit is the mantissa (*m*), and the second digit is the exponent (*n*) in the formula:

```
LENGTH = m x 2ⁿ
```

Thus, RUSIZES=X'8585' specifies length = 256 (8×2^5) for both logical units.

Recommended values are:

- For local-attached devices: X'85C7'
- For communication-attached devices: X'8585'
- For communication-attached devices: 3935
  - Communication-attached through a 3745 with SDLC modem: X'86F8'
  - Communication-attached through a 3745 with token-ring subsystem: X'87F8'
- For 3935 printer:
  - Communication-attached via 3745 with SDLC modem: X'86F8'
  - Communication-attached via 3745 with token-ring subsystem: X'87F8'

SECPROT=*hex value*
The value is X'B0'.

SRCVPAC=*hex value*
Suggested values are:

- For an LU type 6.2 device: X'03'
- For an LU type 1 device: X'nn'
  - Local-attached through a control unit token-ring Gateway: X'10'
  - Communication-attached through a remote control unit token-ring Gateway: X'10'
  - Communication-attached through a local or remote communication controller token-ring Subsystem: X'10'
  - Communication-attached by means of an SDLC link: X'03'.

Note: 3820 printers default to a pacing of X'03' if any larger number is specified.

SSNDPAC=*hex value*
The value for all devices is X'00'.

TSPROF=*hex value*
The value for an LU type 6.2 device is X'07'. The value for an LU type 1 device is X'03'.

Defining Local Major Nodes

A local-attached configuration is defined to VTAM as using a local major node. For example, for a printer connected to an LU type 1 control unit, the connection between VTAM and the control unit is defined with a PU statement. Each printer attached to the control unit is defined with an LU statement. Figure 17 shows an example.

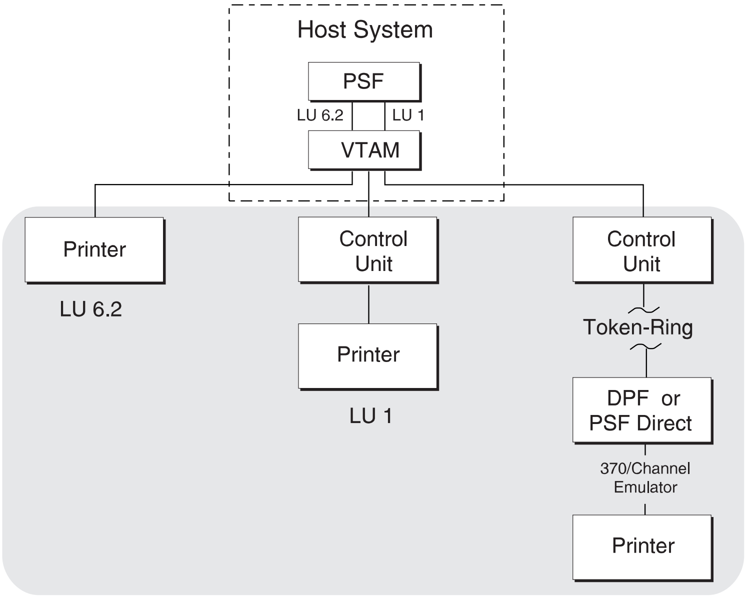


Figure 17. Example Showing Three Local Major Nodes. A node can be for either a printer or a control unit.

Note: For information about configuring host-connected, channel-attached 3820 printers, see “Defining IBM Channel-Attached 3820 Page Printers with MVSCP” on page 10. Figure 18 shows an example of VTAM definitions for local major nodes.

```
For an LU type 6.2 device

CHAN3TST  VBUILD  TYPE=LOCAL
PUYALE8   PU      CUADDR=nnn,MAXBFRU=13
LUYALE8   LU      LOCADDR=1,DLOGMOD=IBM3820C,MODETAB=MODETAB1

For an LU type 1 device

LOC3812   VBUILD  TYPE=LOCAL
PU13812   PU      CUADDR=nnn,MAXBFRU=2
LU13812   LU      LOCADDR=8,MODETAB=MODETAB1,DLOGMOD=IBM3812C,ISTATUS=ACTIVE
```

Figure 18. Sample Local Major Node Definitions for Different LU Protocols. These statements are commonly found in SYS1.VTAMLST.

VBUILD

Describes the node.

TYPE=LOCAL

Specifies a local major node.

PU

Describes the physical unit. For LU type 1 printers, the PU is the control unit.

CUADDR

Specifies the hexadecimal channel-device name to be used when activating the PU. The value must match the channel-device name specified in the device configuration for the control unit or printer connected to the host.

MAXBFRU

Specifies, as a decimal integer, the number of buffer units (elements of the IOBUF buffer pool) that are used for data by VTAM. The number of buffers depends on the maximum RU size and the size of the IOBUF elements. For information on customizing this value, refer to your publications for VTAM.

LU

Describes the logical unit (LU). The name specified on the LU statement must also be specified on the LUNAME parameter in the PRINTDEV statement of the PSF startup procedure for the printer.

DLOGMOD

Specifies the name of the VTAM logon-mode table entry if one is not otherwise provided. The name must match the LOGMODE operand of the MODEENT statement for the entry. This parameter is required when MGMTMODE=DIALIN is specified in the PRINTDEV statement. For more information, see "Defining the Logon-Mode Table Entry" on page 24. If LOGMODE is specified in the PRINTDEV statement, it overrides the name specified here when either MGMTMODE=IMMED or MGMTMODE=OUTAVAIL is specified.

ISTATUS={ACTIVE|INACTIVE}

Specifies whether this logical unit is to be activated when the major node is first activated.

LOCADDR

Specifies the LU address at the PU. For 3820 or a 3935 printer, LOCADDR=1. For an LU1 printer, use the address of the printer line coming from the control unit. For example, for a 3174 control unit using LU1, this value equals the port number of the printer plus 2.

A LOCADDR of `X'00'` defines the LU as an independent LU, and all sessions use the same `X'00'` LU address.

A LOCADDR of `X'01'` through `X'FF'` defines the LU as a dependent LU, and each session must have a unique LU address.

LOGAPPL

Specifies the name of the application program and must match the name specified in the APPLID parameter in the PRINTDEV statement. This parameter is required only when MGMTMODE=DIALIN in the PRINTDEV statement. For more information, see the note on page 24.

Note: When LOGAPPL is specified, the host application program node must be active before the device LU node is activated. Also, if MGMTMODE=DIALIN, printers cannot be shared.

MODETAB
Specifies the name of the logon-mode table if it is defined locally. If it is not specified, the logon-mode table supplied by IBM is used.

PACING
Determines how VTAM paces the flow of data from the boundary node, which performs pacing for the physical unit to the secondary logical unit (SLU). Zero means no pacing.

VPACING
Determines how VTAM paces the flow of data from the host system to the boundary node, which performs pacing for the physical unit. Zero means no pacing.

Defining Switched Major Nodes

Figure 19 shows major nodes switched to link VTAM with a communication controller.

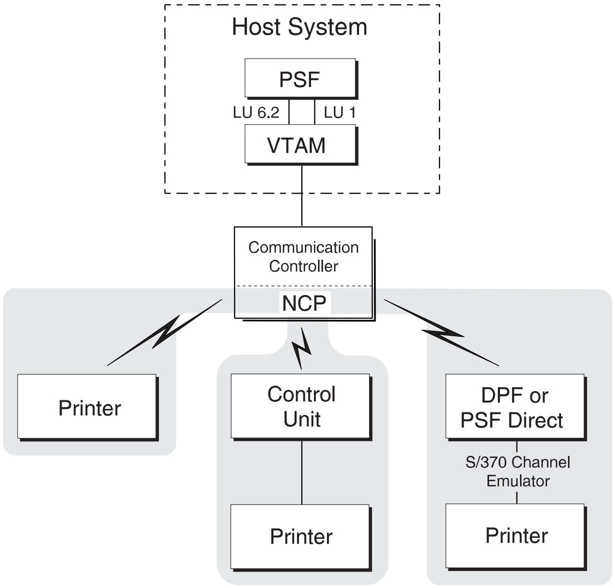


Figure 19. Sample Configuration of Switched Major Nodes. An NCP generation is required.

Figure 20 shows samples that define switched major nodes for several different protocols.

```
For an LU type 6.2 device

SWTAUTIO  VBUILD  TYPE=SWNET,MAXGRP=5,MAXNO=4
PATINOUT  PU      ADDR=C1,IDBLK=nnn,IDNUM=0F0F0,MAXPATH=1,
                  DISCNT=YES,PUTYPE=2
          PATH    DIALNO=95556928,GID=4,GRPNM=SNAS5
LUPRT1    LU      LOCADDR=1,LOGAPPL=PSFFSA1,DLOGMOD=zzzzzzzz,
                  MODETAB=MODETAB1

For an LU type 1 device

SWT3812   VBUILD  TYPE=SWNET,MAXGRP=1,MAXNO=2
PU2810    PU      ADDR=C1,IDBLK=017,IDNUM=0E0E0,MAXPATH=1,PUTYPE=2
          PATH    DIALNO=9245566,GID=1,GRPNM=G28SW10,DISCNT=NO
LU381210  LU      LOCADDR=8,DLOGMOD=IBM3812,MODETAB=MODETAB1
```

Figure 20. Sample Definitions of Switched Major Nodes. Continuation characters in column 72 are not shown. These statements are commonly found in SYS1.VTAMLST(member name).

An NCP generation is required. To do the NCP generation, see "Defining Network Control Program (NCP) Major Nodes" on page 33.

The samples consist of the following statements:

VBUILD
Defines the node.

MAXGRP
Specifies the number of unique path groups (GROUP names) that are specified in the GRPNM operand of each PATH statement in the switched major node.

MAXNO
Specifies how many unique telephone numbers are specified in the DIALNO operand of each PATH statement in the switched major node.

TYPE
Specifies that the VBUILD statement defines a switched major node.

PU
Defines the physical unit:

ADDR
Specifies the 8-bit SDLC station address for the physical unit.

Note: This value must match the station address specified for the 3820 microcode diskette configuration. For an LU type 1 printer, this address must match the address of the control unit.

DISCNT
Specifies whether VTAM should physically disconnect the physical unit when the logical unit terminates its session with PSF. DISCNT=YES prevents excess line charges; with DISCNT=NO, the line remains connected after the session is complete. For more information, refer to your publications for VTAM.

IDBLK

Specifies the device identification number. For a 3935 printer, this value must be X'071'. For most other LU type 6.2 devices, this value is X'042'.

IDNUM

Specifies the 20-bit station ID number. This number must match the configuration value of the PU.

Note: VTAM uses the value of IDBLK and the value of IDNUM when building a 48-bit station ID that is used in the XID exchange during the dial procedure. This station ID must be unique for each station in the network. Therefore, each station must have a unique IDNUM value. The bits in the station ID are defined as follows:

Bits 0–3:	Reserved
Bits 4–7:	Contain the value of PUTYPE
Bits 8–15:	Contain X'00'
Bits 16–27:	Contain the value of IDBLK
Bits 28–47:	Contain the value of IDNUM

MAXDATA

Specifies the maximum number of data bytes that the physical unit can receive in one data transfer. MAXDATA equals the maximum RUSIZE for the PU plus the header length, which is 9 bytes. The maximum RU size is specified by the logon-mode table RUSIZES parameter. If RUSIZES=X'8787', for example, MAXDATA=1033.

For non-3820 LU 6.2 devices and for the 3935 printer, MAXDATA=3849 is recommended.

MAXPATH

Specifies the number of dial-out paths to the physical unit.

PUTYPE

Specifies the physical-unit type of the station represented by this PU statement. PUTYPE=2 should be specified and is the default.

PATH

Defines a dial-out path from a communication controller to a physical unit in a switched major node:

DIALNO

Specifies the telephone number that connects with the physical unit—for example, the telephone number of a 3274 control unit.

GID

Identifies a group of paths across all physical units in the switched major node.

GRPNM

Specifies the symbolic name of a GROUP statement in an NCP major node. That GROUP statement defines a logical group of SDLC switched lines or token-ring switched lines.

PID

Identifies the path being defined.

LU

Defines the logical unit. The name specified in the LU statement must match the name specified for the LUNAME option in the PRINTDEV statement of the PSF startup procedure.

DLOGMOD

Specifies the name of the VTAM logon-mode table entry, if one is not otherwise provided. The name must match the LOGMODE operand of the MODEENT statement for the entry. This parameter is required when MGMTMODE=DIALIN is specified in the PRINTDEV statement. See “Defining the Logon-Mode Table Entry” on page 24 for more information. If LOGMODE is specified in the PRINTDEV statement, it overrides the name specified here when either MGMTMODE=IMMED or MGMTMODE=OUTAVAIL is specified.

Note: For most LU type 6.2 devices, DLOGMOD=IBM3820.

LOCADDR

Specifies the LU address at the PU. For an LU 6.2 device, LOCADDR=1.

A LOCADDR of `X'01'` through `X'FF'` defines the LU as a dependent LU, and each session must have a unique LU address.

LOGAPPL

Specifies the name of the application program and must match the name specified in the APPLID parameter in the PRINTDEV statement. This parameter is required only when MGMTMODE=DIALIN in the PRINTDEV statement. For more information, see the note on page 24.

Note: When LOGAPPL is specified, the host application program node must be active before the device LU node is activated.

When MGMTMODE=DIALIN, printers cannot be shared.

MODETAB

Specifies the name of the logon-mode table if it is defined locally. If it is not specified, the logon-mode table supplied by IBM is used.

SSCPFM

Determines whether the logical unit can support field-formatted functional subsystem (FSS) RUs or character-coded RUs over its SSCP/LU session.

VPACING

Specifies how VTAM paces the flow of data from the host system to the boundary node to which the logical unit is connected. A value of 0 indicates no pacing.

Defining Network Control Program (NCP) Major Nodes

An NCP major node defines an NCP, the set of lines attached to it, and the physical units and logical units associated with the lines. Lines of the same type are grouped into sets by the GROUP macroinstruction. Each line in a set is defined by a LINE macroinstruction, followed by PU macroinstructions that define the physical units attached to the NCP by this line and LU macroinstructions that define the logical units associated with the physical units. Each of these macroinstructions, including the GROUP macroinstruction, defines a minor node that represents a network resource.

All the macroinstructions listed in “NCP Generation” on page 34 should also be supplied to VTAM as part of the NCP major node definition. For a diagram of a sample definition, see Figure 21 on page 34. For an overview of the VTAM and NCP definitions that PSF requires see Table 1 on page 21.

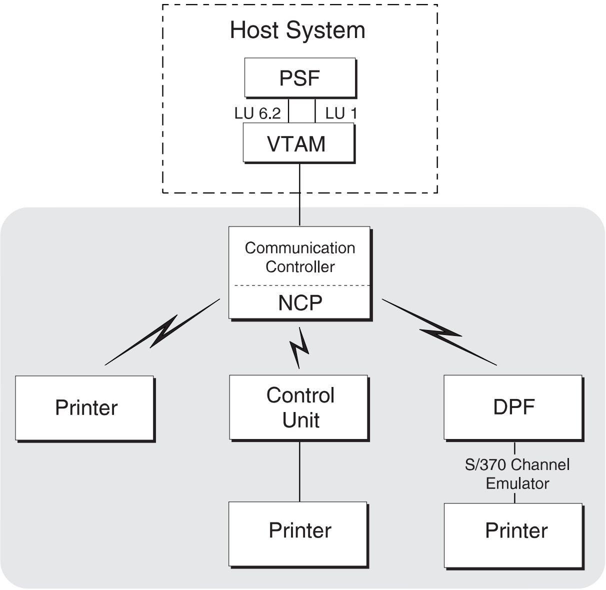


Figure 21. Sample NCP Major Node. Lines can be switched or nonswitched.

NCP Generation

The macroinstructions that compile the NCP have two purposes:

- They generate an NCP that can be loaded into a communication controller and describe to that NCP the lines, line groups, link stations, and terminals it is to control.
- They describe to VTAM the same NCP major node and the resources that the NCP is to control.

Most of the operands in NCP-generation macroinstructions are used only for generating modules of the NCP. Some of the operands, however, are directed either to both NCP and VTAM, or only to VTAM. The NCP-generation macroinstructions are statements for VTAM network definition, and true macroinstructions for NCP generation.

The NCP-only macroinstructions and the operands used by both NCP and VTAM are verified during NCP generation but not during VTAM network definition. Therefore, the NCP should be generated before the generation deck is filed in the VTAM definition library as a VTAM major node.

The macroinstructions to be added to the NCP generation for the printer depend on the communication lines. VTAM and NCP support many network configurations; the following are representative of communication lines used with PSF:

- **Switched line:** A telecommunication line in which the connection is established by dialing.

- **Nonswitched line:** A telecommunication line in which the connection is permanently established and does not require dialing.
- **Multipoint line (nonswitched):** A line connecting more than two devices.
- **Token-ring subsystem connection:** Cabling that links devices in an IBM token-ring local area network (LAN).

Note: The samples and explanations in this section are not intended to be a complete description of the macroinstructions and operands. For more information, refer to your publications for NCP.

Switched Line

Figure 22 shows an example of NCP macroinstructions for switched lines.

```
For an LU type 6.2 device

SNAS1     GROUP   DIAL=YES,LNCTL=SDLC,TYPE=NCP,ISTATUS=ACTIVE,ANSWER=ON,REPLYTO=1
SNAL022   LINE    CALL=OUT,ADDRESS=(035),SPEED=4800,CLOCKNG=EXT,AUTO=03F
PATINOUT  PU      PUTYPE=2,MAXLU=1


For an LU type 1 device

G28SW10   GROUP   DIAL=YES,LNCTL=SDLC,REPLYTO=1,ANSWER=ON,TYPE=NCP
LN2810    LINE    ADDRESS=(010,HALF),CLOCKNG=EXT,SPEED=4800,CALL=INOUT
PU2810    PU      PUTYPE=(2),MAXLU=10
```

Figure 22. Sample NCP Macroinstructions for Switched lines. These statements are commonly found in SYS1.VTAMLST(member name), and are entered during the generation of NCP.

These samples contain the following instructions:

GROUP

Describes a group of communication lines:

ANSWER

Specifies whether the physical units can dial into VTAM when the line is active.

DIAL

Specifies whether the lines in the group require switched-line control procedures.

ISTATUS

Specifies whether the lines in the group are to be activated after the first activation of the major node.

LNCTL

Specifies the line control for the lines in the group.

REPLYTO

Specifies the reply time-out value in seconds for the lines in the group.

TYPE

Specifies, for the lines in the group, that all lines operate in network control mode.

LINE
Describes one communication line:

ADDRESS
Specifies the relative line-number-interface address for the controller.

AUTO
Specifies that the auto-call facility is installed and provides the line number interface address for the controller.

CALL
When CALL=OUT or CALL=INOUT is specified, VTAM prompts the VTAM operator to dial the telephone at the correct time. When CALL=IN and MGMTMODE=DIALIN are specified, PSF waits for the printer operator to dial.

CLOCKNG
Specifies whether the modem or the communication controller for the line is to provide the clocking.

SPEED
Specifies the data rate for this line in bits per second.

PU
Defines the physical unit:

MAXLU
Specifies the maximum number of logical units that can be associated with this physical unit.

PUTYPE
Specifies the physical-unit type of the station represented by this PU macroinstruction. PUTYPE=2 must be specified.

Nonswitched Line

Figure 23 shows sample NCP macroinstructions for three nonswitched lines.

```
For an LU type 6.2 device

SNAL1     GROUP   DIAL=NO,LNCTL=SDLC,TYPE=NCP,REPLYTO=1,ISTATUS=ACTIVE
SNAL022   LINE    ADDRESS=(022),SPEED=9600,CLOCKNG=EXT,RETRIES=(5,1,2)
          SERVICE ORDER=(CTL01)

CTL01     PU      ADDR=C1,ANS=STOP,MAXDATA=nnnn,
                  MAXOUT=7,PUTYPE=2
LUPRT1    LU      LOCADDR=1,LOGAPPL=PSFFSA1,DLOGMOD=zzzzzzzz,MODETAB=MODETAB1

For an LU type 1 device

G28PTPT   GROUP   DIAL=NO,LNCTL=SDLC,TYPE=NCP,ISTATUS=ACTIVE,REPLYTO=1
LN2804    LINE    ADDRESS=(004,HALF),CLOCKNG=EXT,RETRIES=(5,1,2),SPEED=9600
          SERVICE ORDER=PU2804

PU2804    PU      ADDR=C1,ANS=STOP,MAXDATA=265,MAXOUT=7,PASSLIM=8,MAXLU=10,PUTYPE=2
LU38124   LU      DLOGMOD=IBM3812,LOCADDR=8,MODETAB=MODETAB1
```

Figure 23. Sample NCP Macroinstructions for Nonswitched Lines. Continuation characters in column 72 are not shown. These statements are commonly found in SYS1.VTAMLST(member name), and are input during the generation of NCP.

This example consists of the following instructions:

GROUP

Describes a group of communication lines:

ACTIVTO

Specifies the time-out value that the secondary NCP waits for communication from the primary NCP before initiating shutdown.

DIAL

Specifies whether the lines in the group require switched-line control procedures.

ISTATUS

Specifies whether the group is to be active after the first activation of the major node.

LNCTL

Specifies the line control for the lines in this group.

MODE

Specifies the mode of operation of a line group associated with a selection table entry defined by an SDLCST operand in a LINE definition statement.

REPLYTO

Specifies the reply time-out value in seconds. One second is the default for SDLC lines.

TYPE

Specifies, for the line group, that all lines are to operate in network control mode.

LINE

Describes one communication line:

ADDRESS

Specifies the relative line number of the line interface address for the controller.

CLOCKNG

Specifies whether the modem or the communication controller is to provide clocking.

DUPLEX

Specifies whether the communication line and modem have a half-duplex or full-duplex capability.

IPL

Specifies whether the NCP being generated can be loaded and dumped over this link.

PAUSE

Specifies a cycle delay according to the link.

RETRIES

Specifies the number of attempts to recover from errors that occurred during transmission over the link.

SDLCST

Specifies respectively which SDLC selection-table entry the NCP uses when it is operating in a primary state and which entry the NCP uses when it is operating in a secondary state.

SPEED
Specifies the data rate for this line in bits per second.

SERVICE
Generates a service-order table for a nonswitched SDLC link:

ORDER
Specifies the order in which the stations on the line are serviced.

PU
Defines the physical unit:

ADDR
Specifies the 8-bit SDLC station address for the physical unit.

Note: This value must match the station address specified for the device configuration. For an LU type 1 printer, this address must match the address for the control unit.

ANS
Specifies whether the operation of the station is to stop or continue if the NCP enters automatic network shutdown.

MAXDATA
Specifies the maximum number of data bytes that the physical unit can receive in one data transfer. MAXDATA equals the maximum RU size for the PU plus the header length, which is 9 bytes. The maximum RU size is specified by the logon-mode table RUSIZES parameter. For example, when RUSIZES=X'8585', MAXDATA=265. For most LU type 6.2 and LU type 1 devices, MAXDATA=265.

For non-3820 LU6.2 devices and the 3935 printer, MAXDATA=3849 is recommended.

MAXOUT
Specifies the maximum number of path-information units that the NCP will send to the physical unit before requesting a response. The recommended range for this value is from 3 to 7.

PASSLIM
Specifies the maximum number of consecutive path information units (PIUs) or PIU segments the NCP sends at one time to the physical unit.

PUTYPE
Specifies the physical-unit type of the SDLC station represented by this PU macroinstruction.

TGN
Specifies the transmission group number of the subarea link associated with this physical unit.

LU
Defines the logical unit. The name specified in the LU statement must match the name specified for the LUNAME parameter for the printer FSA in the PRINTDEV statement of the PSF startup procedure.

DLOGMOD
Required when MGMTMODE=DIALIN is specified in the PRINTDEV statement. Specifies the name of the VTAM logon-mode table entry if one is not otherwise provided. The name must match the LOGMODE operand of the

MODEENT statement for the entry. For more information, see “Defining the Logon-Mode Table Entry” on page 24. If LOGMODE is specified in the PRINTDEV statement, it overrides the name specified here when either MGMTMODE=IMMED or MGMTMODE=OUTAVAIL.

Note: For most LU type 6.2 devices, DLOGMOD=IBM3820.

LOCADDR
Specifies the LU address at the PU. For LU 6.2 devices, LOCADDR=1.

A LOCADDR of X'01' through X'FF' defines the LUas a dependent LU, and each session must have a unique LU address.

LOGAPPL
Specifies the name of the application program and must match the name specified in the APPLID parameter in the PRINTDEV statement. Required only when MGMTMODE=DIALIN in the PRINTDEV statement. For more information, see the note on page 24.

Note: When LOGAPPL is specified, the host application program node must be active before the device LU node is activated.

Also, if MGMTMODE=DIALIN, printers cannot be shared.

MODETAB
Specifies the name of the logon-mode table if it is defined locally. If MODETAB is not specified, the logon-mode table supplied by IBM is used.

Multipoint Nonswitched Line

Figure 24 shows an example of macroinstructions for generating NCP for multipoint lines.

```
For an LU type 6.2 device

PRTRPPM   LINE    ADDRESS=026,SPEED=9600,CLOCKNG=EXT,RETRIES=(5,1,2)
          SERVICE ORDER=(PRTRPU3,PRTRPU4)

PRTRPU3   PU      ADDR=C3,ANS=STOP,MAXDATA=nnnn,MAXLU=1,MAXOUT=7,PUTYPE=2
PRTRLU3   LU      LOCADDR=1,LOGAPPL=PSFFSA3,DLOGMOD=zzzzzzzz,MODETAB=MODETAB1
PRTRPU4   PU      ADDR=C4,ANS=STOP,MAXDATA=nnnn,MAXLU=1,MAXOUT=7,PUTYPE=2
PRTRLU4   LU      LOCADDR=1,LOGAPPL=PSFFSA4,DLOGMOD=zzzzzzzz,MODETAB=MODETAB1

For an LU type 1.0 device

LN2804    LINE    ADDRESS=(004,HALF),CLOCKNG=EXT,RETRIES=(5,1,2),PACING=3,
                  VPACING=3,SPEED=9600
          SERVICE ORDER=(PU2804A,PU2804B)


PU2804A   PU      ADDR=C1,ANS=STOP,IRETRY=YES,MAXDATA=265,MAXOUT=7,
                  PASSLIM=8,MAXLU=10,PUTYPE=2
LU32704A  LU      LOCADDR=2,MODETAB=MODETAB1,USSTAB=USSTAB1,
                  DLOGMOD=S32782,ISTATUS=ACTIVE
LU32704B  LU      LOCADDR=3,MODETAB=MODETAB1,USSTAB=USSTAB1,
                  DLOGMOD=S32782,ISTATUS=ACTIVE
LU38124A  LU      LOCADDR=8,MODETAB=MODETAB1,DLOGMOD=IBM3812
PU2804B   PU      ADDR=C2,ANS=STOP,IRETRY=NO,MAXDATA=265,MAXOUT=3,
                  MAXLU=10,PUTYPE=2
LU38204B  LU      LOCADDR=01,MODETAB=MODETAB1,DLOGMOD=IBM3820,
                  LOGAPPL=PSFFSA1
```

Figure 24. Sample NCP Macroinstructions for Multipoint Lines. Continuation characters in column 72 are not shown.

This example consists of the following instructions:

LINE
Describes one communication line:

ADDRESS
Specifies the relative line number of the line interface address for the controller.

CLOCKNG
Specifies whether the modem or the communication controller is to provide clocking.

DUPLEX
Specifies whether the communication line and the modem have a half-duplex or full-duplex capability.

RETRIES
Specifies how many times PSF is to attempt to recover from errors that occurred during transmission over the link.

SPEED
Specifies the data rate for this line in bits per second.

SERVICE
Generates a service-order table for a nonswitched SDLC link, using the following operand:

ORDER
Specifies the order in which the stations on the line are serviced.

PU
Defines the physical unit:

ADDR
Specifies the 8-bit SDLC station address for the physical unit.

Note: This value and the station address specified for the device must match. For an LU type 1 printer, this address must match the address for the control unit.

ANS
Specifies whether the operation of the station is to stop or continue if the NCP enters automatic network shutdown.

IRETRY
Specifies whether the NCP will again poll a secondary station before advancing to the next station in the service-order table.

MAXDATA
Specifies the maximum number of data bytes that the physical unit can receive in one data transfer. MAXDATA equals the maximum RU size for the PU plus the header length, which is 9 bytes. The maximum RU size is specified by the RUSIZES parameter in the logon-mode table. When RUSIZES=X'8585', for example, MAXDATA=265.

For non-3820 LU 6.2 devices, MAXDATA=3849 is recommended.

MAXLU
Specifies the maximum number of logical units that can be associated with this physical unit.

MAXOUT
Specifies the maximum number of path information units that the NCP is to send to the physical unit before requesting a response. The recommended range for this value is from 3 to 7.

PASSLIM
Specifies the maximum number of consecutive path units (PIUs) or PIU segments the NCP is to send to the physical unit at one time.

PUTYPE
Specifies the physical unit type of the SDLC station represented by this PU macroinstruction. PUTYPE=2 must be specified.

LU
Defines the logical unit. The name specified in the LU statement for the device must match the name specified for the LUNAME parameter for the printer FSA in the PRINTDEV statement of the PSF startup procedure. LU defines the logical unit:

DLOGMOD
Specifies the name of the VTAM logon-mode table entry if one is not otherwise provided. The name must match the LOGMODE operand of the MODEENT statement for the entry. For more information, see "Defining the Logon-Mode Table Entry" on page 24. If LOGMODE is specified in the

PRINTDEV statement, it overrides the name specified here when either MGMTMODE=IMMED or MGMTMODE=OUTAVAIL.

Note: For most LU type 6.2 devices, DLOGMOD=IBM3820.

ISTATUS
Specifies whether the logical unit is to be active after the first activation of the major node.

LOCADDR
Specifies the LU address at the PU. For LU 6.2 devices, LOCADDR=1.

A LOCADDR of X'01' through X'FF' defines the LU as a dependent LU, and each session must have a unique LU address.

LOGAPPL
Specifies the LU address at the PU. For LU 6.2 devices, LOCADDR=1. For an LU type 1 printer, use the address of the printer line coming from the control unit.

MODETAB
Specifies the name of the logon-mode table if it is defined locally. If MODETAB is not specified, the IBM-supplied logon-mode table is used.

USSTAB
Specifies the name of a USS definition table. If USSTAB is not specified, a USS definition table supplied by IBM is searched when VTAM receives character-coded input from the logical unit.

Note: The USSTAB operand may be specified on LU statements for terminals, but is typically not specified on LU statements for printer devices. If USSTAB is specified for printer LU statements, USSMSG10 should not be included in the USSTAB defined.

Token-Ring Subsystem Connection

GROUP

Describes a group of communication lines:

AUTOGEN
Specifies how many LINE and PU definition-statement pairs must be added to the GROUP statement.

CALL
Determines whether VTAM is able to establish connections over this group of lines.

ECLTYPE
Specifies whether this GROUP statement defines a physical or logical connection to the token ring.

ISTATUS
Specifies whether the group is to be active or inactive after the major node is first activated.

MAXLU
Specifies the number of logical units that can be associated with a physical unit.

PHYPORT
Specifies the physical port address with which the lines in this logical group will communicate. The value must correspond to a PORTADD value in a physical group LINE definition statement.

LINE
Describes one communication line:

ADDRESS
Specifies the relative line number of this line and whether data will be transferred in half-duplex or full-duplex mode.

ISTATUS
Specifies whether the line is active or inactive after the major node is first activated.

LOCADD
Specifies the address of the communication controller on the token ring.

MAXTSL
Specifies the maximum number of data bytes that the NCP token-ring interface (NTRI) can transmit. MAXTSL equals the maximum RU size for any PU plus the header length (9 bytes). The maximum RU size is specified by the RUSIZES parameter in the logon-mode table. When RUSIZES=X'8787', for example, MAXTSL=w033; when RUSIZES=X'86F8', MAXTSL=3849. For a formula for calculating RUSIZES, see page 27.

PORTADD
Specifies the physical port address.

RCVBUFC
Specifies the NCP token-ring interface (NTRI) buffer capacity for receiving data from a data link during one data transfer.

General Network Considerations

When establishing a network, you should consider the following:

- Logon-mode table
- VTAM definition names
- LOGAPPL statement and the LU statement

Logon-Mode Table

The logon-mode table entry must be specified in one of two places:

- The LOGMODE parameter of the PRINTDEV statement.
- The DLOGMOD parameter of the LU statement for the printer. In the cross-domain case, when DLOGMOD is specified, the logon-mode table entry must be defined in the VTAM that owns the printer LU.

If the LOGMODE option is specified in the PRINTDEV statement, VTAM uses it. If that option is not specified, VTAM uses the DLOGMOD operand. If neither LOGMODE nor DLOGMOD is specified, the default DLOGMOD entry for VTAM is used.

If the LOGMODE option is invalid, an error message is issued even if the DLOGMOD operand is valid. Conversely, if the DLOGMOD operand is invalid but the LOGMODE option is valid, no message is issued.

VTAM Definition Names and PSF

The following values must match:

- The name of the APPL statement in the VTAM definition for the PSF application node
- Any LOGAPPL name of the VTAM definition for generating the NCP for the printer

  Note: LOGAPPL is required only when MGMTMODE=DIALIN.
- The APPLID name specified in the PRINTDEV statement

LOGAPPL and the LU Statement

When MGMTMODE=DIALIN is specified in the PRINTDEV statement, LOGAPPL is required only on the LU statement. If LOGAPPL is specified, the DLOGMOD must also be specified in the LU statement for the printer.

PSF PRINTDEV Statement Considerations for Switched Lines

The kind of printing session PSF establishes depends on the parameter specified for MGMTMODE in the PRINTDEV statement. For communication-attached devices on switched lines, choose:

- MGMTMODE=DIALIN
- MGMTMODE=OUTAVAIL
- MGMTMODE=IMMED

MGMTMODE=DIALIN is recommended for printers where a switched line goes directly to a single printer (for example, the 3820 printer), and is dropped when the LU-LU session ends. PSF then uses a VTAM OPNDST command to accept the CINIT generated by LOGAPPL and DLOGMODE. You should also specify DISCNT=YES. For more information, see the DISCNT statement, described in “Defining Switched Major Nodes,” and Figure 20 on page 31 .

The CALL=IN | OUT | INOUT parameter determines whether PSF waits for the printer operator to dial in or prompts the VTAM operator to dial out.

MGMTMODE=OUTAVAIL and MGMTMODE=IMMED are recommended for printers in which the line goes to a control unit (for example, the 3812 printer) and cannot be dropped when the LU_LU session ends. PSF then uses a VTAM SIMLOGON command to request a session with the printer.

Note: When a printer has not been dialed, message APS925I is issued to remind the host operator that the control unit needs to be dialed.

The CALL=OUT | INOUT parameter should be used. LOGAPPL is not required.

3130, 3160, and 3935 Configuration

Specific information and examples for various attachments of these printers are presented in *Attachment Configuration Handbook for Printers That Use the Advanced Function Common Control Unit*.

DPF Considerations

When setting up communication between DPF and a printer, use the PSF/2 publications listed in Appendix K, “Related Publications.”

Note: Attachment to a microfilm device is not supported through DPF.

PSF Direct Considerations

When setting up communication between PSF Direct and a printer, use the PSF/2 and PSF for AIX publications listed in Appendix K, “Related Publications.”

Note: Attachment to a microfilm device is not supported through PSF Direct.

Chapter 4. Working with Libraries

This chapter provides general information about defining library data sets, formatting libraries, concatenating defined libraries to system libraries, and working with user libraries in deferred-printing mode.

You can define the following libraries that contain the resources needed for printing:

- Font library
- Form definition library
- Page definition library
- Overlay library
- Page segment library
- Object container library

Note: When installing PSF, you do not define user libraries. They are set up by a user and are dynamically accessed by PSF when specified in the user's JCL.

For more information about installing resources in libraries, refer to the PSF program directory provided with your PSF tapes.

Library Format

Libraries must be partitioned data sets (PDS) with record formats of variable-blocked machine (VBM), variable-blocked ANSI[3] (VBA), variable machine (VM), or variable ANSI (VA) controls.

Concatenating Defined Libraries to System Libraries

You can define libraries and concatenate them to the system libraries. For example, a single library can contain a combination of fonts, page segments, and overlays. Use unique member names within concatenated libraries so that PSF can access a particular member in a library. If a member in a concatenated library has the same name as a member in the system library, PSF uses the member it finds first. When PSF is started, libraries are searched in the order in which they were concatenated. For information about concatenating libraries, refer to the JCL user's guide for your operating system.

With user library support, you may not want to concatenate all such libraries to the system libraries. You should not concatenate to the system libraries any libraries that have only a few users, or that contain sensitive resources. You can limit access to such libraries by using RACF control, and the users can specify these libraries in their JCL with the USERLIB keyword.

[3] ANSI stands for American National Standards Institute.

User Libraries in Direct-Printing Mode

When processing the print data set in an MVS/ESA system, PSF accesses user libraries for AFP resources (fonts, page segments, overlays, page definitions, form definitions, and object containers). Print-job submitters specify user libraries by specifying the USERLIB parameter in their JCL OUTPUT statement, as in this example:

```
//OUT1   OUTPUT   USERLIB=(USER.IMAGES.AFP,USER.FONTS.AFP)
```

This example directs PSF to search for AFP resources in the two specified user libraries. If unable to find the resources in the user libraries, PSF searches the system libraries defined in the startup procedure.

You can use RACF to control access to user libraries. PSF checks whether the print-job submitter has RACF authority to access a specified user library; if not, PSF does not print the job. When the data set has finished printing, PSF deletes the resource from the printer.

Because PSF runs as a started task and must be permitted access to a user library specified with the USERLIB parameter, the RACF-Started Procedures Table (ICHRIN03) must include an entry for PSF, declaring it privileged or trusted. For information about the Started Procedures Table, refer to the *RACF System Programmer's Guide*.

Keep in mind the following constraints to dynamic allocation:

- You can specify up to 8 user libraries in your JCL.
- User library data sets must be accessible on all systems on which output is selected to print. This may necessitate reconfiguring shared volumes or storage classes, and allocating the resource libraries to such volumes or storage classes.
- PSF does not support the USERLIB parameter in direct-printing mode.
- If a virtual storage access method (VSAM) format catalog or JOBCAT/STEPCAT has a data set allocated by PSF, the catalog is allocated to the PSF address space. This means that maintenance on the catalog cannot be performed until PSF is terminated. This may not be a concern for system catalogs, but may be a concern for user catalogs.
- In Data Facility Product (DFP) 3.1.0, integrated catalog facility (ICF) catalogs are not allocated to the user's address space in PSF. To free the catalogs for maintenance, you can enter a catalog address space (CAS) restart or a MODIFY CATALOG, UNALLOCATE command. For more information, refer to specific DFP publications listed in Appendix K, “Related Publications.”
- Dynamic allocation by all tasks with an address space serializes on SYSZTIOT, which could cause a performance problem when PSF supports large numbers of FSAs in an address space. Therefore, use user libraries only when necessary.

- The maximum number of catalogs that an address space can allocate is 1635, which can optionally be increased to 3274. If every user in the system has a separate catalog, be aware of the possibility of approaching the maximum number of catalogs.

For more information about user libraries, refer to *PSF/MVS: Application Programming Guide*.

Chapter 5. Changing the System Environment

After you install PSF, you must change the system environment—MVS program properties table (PPT), system libraries, JES, and VTAM—before you can use PSF 2.2.0. The changes you make depend on whether PSF has already been installed, which printers are being used, and whether printers are controlled with JES.

How to Change the System Environment for PSF 2.2.0

To change the system environment for PSF 2.2.0:

- Update the PPT. An IBM-supplied default, internal PPT entry is provided with MVS for PSF. The MVS internal default values are:

  ```
  PPT    PGMNAME(APSPPIEP)
         KEY(1)
         NOSWAP
         NOPRIV
         CANCEL
         SYST
         NODSI
         PASS
         AFF(NONE)
         NOPREF
  ```

  These are the recommended values. You can change them by specifying a PPT entry in the SCHEDxx member in the system PARMLIB with overriding values for APSPPIEP. For more details, refer to the *Initialization and Tuning Reference* for your operating system, listed in Appendix K, “Related Publications.”

  Specifying NODSI (nonexclusive use of data sets) in the PPT enable other programs to change the resources while PSF is running. Specifying DSI and running PSF with DISP=SHR causes other programs to wait until PSF is no longer running before the other programs can update the PSF libraries. Modification of any other values may cause JES to abend, or other unexpected results to occur.

- Set up the printer for either deferred-printing mode or direct-printing mode:

  – To set up the printer for deferred-printing mode with JES:

    If the printer is used with JES, you must change the system to enable PSF 2.2.0 to act as the interface between JES and the printer. The following modifications are required:

    1. Define the PSF printer to JES. The initialization parameters for JES2 or JES3 must be changed for a PSF functional subsystem (FSS). For more information about JES, see Chapter 7, “Using Deferred-Printing Mode” on page 63.

       You should understand the hierarchy by which resources are selected when the printer is used with JES. The JES FCB and UCS parameters and work selection criteria are described in Chapter 11, “Managing Resources and Selecting Special Processing Parameters” on page 113.

2. Update the provided startup procedure—APSWPROA, APSWPROB, APSWPROC, APSWPROD, APSWPROM, APSWPROS, or APSWPROT, whichever is used for the installation verification program (IVP). For an example of IVP output from a 240-pel printer, see Appendix G, “Installation Verification Program (IVP) Example.” These jobs are placed in SYS1.PROCLIB when PSF 2.2.0 is installed.

3. For more information, see Chapter 7, “Using Deferred-Printing Mode.”

– To set up the printer for use in direct-printing mode (host-connected, channel-attached non-SNA printers only).

  - You cannot set up a printer to use both direct-printing mode and deferred-printing mode at the same time.
  - For more information, see Chapter 8, “Using Direct-Printing Mode.”
  - You cannot use microfilm devices in direct-printing mode.

- To ensure that PSF has access to RACF-protected resource libraries, add an entry for PSF to the RACF-Started Procedures Table (ICHRIN03) and declare PSF to be privileged or trusted. This ensures that PSF has access to RACF-protected resource libraries. For information about the Started Procedures Table, refer to *RACF System Programmer's Guide* for your operating system, listed in Appendix K, “Related Publications.”

- Set up for VTAM/NCP (SNA only)

  An SNA-attached printer is a networked device that uses ACF/VTAM alone or with ACF/NCP to communicate with your system. Before you can use the printer, you must update the ACF/VTAM network definition or perform an ACF/NCP generation.

  All printer configurations that contain a 37X5 communication controller require ACF/NCP and ACF/VTAM. All other SNA configurations require only ACF/VTAM.

  For more information, see Chapter 3, “Establishing Network Nodes for SNA-Attached Printers.”

For host-connected printers that are channel-attached and ESCON-attached, Dynamic Reconfiguration Management (DRM) affects the installation of channel-attached printers (only with MVS SP 4.2.0 and above). For more information, see “Defining Channel-Attached Printers with HCD” on page 13. For information about printers attached with TCP/IP, see Chapter 19, “Connecting a TCP/IP-Attached IPDS Printer” on page 235.

Chapter 6. PRINTDEV Parameters

The PRINTDEV statement specifies parameters for printer initialization. In deferred-printing mode, PRINTDEV is among the JCL statements in the PSF startup procedure. In direct-printing mode, PRINTDEV is among the JCL statements associated with the application program. For JCL descriptions and sample PRINTDEV statements, see Chapter 7, “Using Deferred-Printing Mode,” and Chapter 8, “Using Direct-Printing Mode.”

Table 2 describes the initialization parameters that can be included in a PRINTDEV statement.

Table 2 (Page 1 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
ACQINTV (optional)	Specifies the acquire interval in seconds. This is the time period between releasing a printer and attempting to acquire it again. ACQINTV is used only when RELMODE=TIME. **Syntax: ACQINTV=***nn*, where *nn* is a value from 0 to 86400.
APPLID (required for SNA-attached printers only)	Specifies the name of the VTAM application-program node for a functional subsystem application (FSA) when PSF is printing in deferred-printing mode. Each PSF FSA must have a unique name in the VTAM network. This name must match the following two values in the VTAM printer-network definition: • The name of the APPL statement in the PSF node definition for the application program • The value for the LOGAPPL parameter in the node definition for the printer For more information, see “Defining a PSF Application Node” on page 23, and “Defining Switched Major Nodes” on page 30. **Syntax: APPLID**=*name* **Note:** Only for SNA-attached printers (required).
BUFNO (optional)	Specifies a factor for reserving real, fixed storage used in printer I/O operations. (To determine the total amount of reserved storage, the value of BUFNO is multiplied by 4K (4096).) The value of BUFNO can range from 5 to 100. The minimum BUFNO is 5; or, for ESCON channel-attached printers and printers that support 16K I/O buffers, the minimum BUFNO is 9. **Syntax: BUFNO**=*nnn* **Note:** Only for channel-attached printers that do not use DPF or PSF Direct.
CHARS (optional)	Specifies up to four default font names. A font name must contain 1 to 4 alphanumeric or national characters. For information on selecting fonts, see “How PSF Selects Resources” on page 117. **Syntax: CHARS**=(*fontname1*[,*fontname2*][,*fontname3*][,*fontname4*])

Table 2 (Page 2 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
COMPRESS (optional)	Specifies whether PSF should compress data (blanks) in line data sent to printers, including printers attached to PSF/2 or PSF for AIX. Selecting YES causes PSF to use an IPDS command to compress contiguous blanks. (The printer will decompress the data.) **Syntax: COMPRESS=YES|NO** **Notes:** 1. This parameter will probably not improve data transmission for channel-attached printers. 2. Blank compression is a data-compression function in PSF/MVS. It reduces the amount of data that has to be sent through the attachment (host-connected channel, SNA, or TCP/IP) to the printer. 3. PSF compresses any string of more than five contiguous blanks within line data with an IPDS control string. The printer replaces the IPDS control string with the correct number of blanks. 4. In certain conditions, this support may improve the throughput of SNA-attached or TCP/IP-attached printers. For more information, see “Blank Compression for Line Data (COMPRESS Parameter)” on page 229.
COMSETUP (required for microfilm device only)	Specifies the member name (from 1 to 8 alphanumeric or national characters) of the printer default microfilm setup resource object container generated with the utility provided with your microfilm device. The COMSETUP parameter is used only when output is being sent to a microfilm device. IBM recommends a prefix of **H1** for microfilm setup resources, and the full member name must be specified, because PSF does not add a prefix. For an example of a startup procedure that uses the COMSETUP parameter with a full member name, see Figure 67 on page 256. For example, `COMSETUP=H1SETUPS` When PSF is attached to an microfilm device, if the COMSETUP parameter is missing from an AFP job submitted to that microfilm device, or if no member name has been specified or provided in the PRINTDEV, PSF terminates the job. For more information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.” For non-microfilm devices, this parameter is ignored. **Syntax: COMSETUP**=*membername*
CONNINTV (optional)	Specifies the connect interval, in seconds. This is the limit on the time during which PSF is to attempt to start a session or a dialog with a printer. If the connect interval expires before connection is complete, PSF issues a message and ends the FSA. If CONNINTV is not specified, or CONNINTV=0, PSF continues attempting to connect for an unlimited time. For SNA, the connect interval is set when a session request is queued after a busy or not available condition. For TCP/IP, the connect interval is set for any retriable condition. For IPDS, the connect interval is set when a dialog is restarted after a release (RELMODE=IDLE|TIME). **Syntax: CONNINTV=0**|*nn*, where *nn* is a value from 0 to 86400. The default value is 0.

Table 2 (Page 3 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
DATACK (optional)	Specifies the data-check exceptions the printer reports to PSF. Printers report two data-check exceptions: print-positioning exceptions, and invalid-character exceptions. • Print-positioning exceptions occur when attempts are made to print outside the valid printable area or off the logical page. Some printers use exception highlighting to mark print-positioning exceptions on the printed page. • Invalid-character exceptions occur when attempts are made to use a code point that is not assigned to a character. **DATACK=BLOCK** Specifies that the printer is not to report data-check exceptions to PSF. **DATACK=UNBLOCK** Specifies that the printer is to report all data-check exceptions to PSF. **DATACK=BLKCHAR** Specifies that the printer is not to report invalid-character exceptions to PSF. (The printer will report only print-positioning exceptions.) **DATACK=BLKPOS** Specifies that the printer is not to report print-positioning exceptions to PSF. (The printer will report only invalid-character exceptions.) Regardless of what the PRINTDEV statement specifies, PSF defaults to DATACK=UNBLOCK when printing security resources. For more information on security resources, refer to *Print Services Facility: Security Guide*. **Syntax: DATACK**=**BLOCK**| **UNBLOCK**|**BLKCHAR**|**BLKPOS** **Notes:** 1. Unless PIMSG=NO, data-check exceptions also produce error messages. (PIMSG is described on page 59.) 2. For microfilm devices, see Chapter 21, “Supporting AFP Output on Microfilm.”

Table 2 (Page 4 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
DISCINTV (optional)	Specifies the disconnect interval in seconds. When no output is available from JES for this time period, PSF ends the session with the printer. PSF makes sure that all pages from the last job reach the stacker before ending the session and allowing that job to be purged from the spool. If DISCINTV=0, a lack of output does not cause PSF to end the printer output. (See note 2 for additional information.) If RELMODE=IDLE or TIME, DISCINTV is ignored. For more information, see “Printer Sharing in the MVS Environment” on page 60. **Syntax: DISCINTV**=*nnn*, where *nnn* is a value from 0 to 86400. **Notes:** 1. For SNA-attached and TCP/IP-attached printers only. TCP/IP attachment includes printers attached with the i-data 7193 IPDS Printer LAN Attachment (see Appendix J, “Installing the i-data 7913 IPDS Printer LAN Attachment” on page 329). 2. Some printers have a network inactivity timer. If the timer is enabled, the DISCINTV value must be less than that of the inactivity timer and greater than 0. Otherwise, PSF abends when the inactivity timer expires. The inactivity timer can be set with the printer-configuration procedure. 3. When some printers, such as the 3820 printer, become inoperative, a *local intervention timer* is started. When the timer expires, an action code `X'22'` negative acknowledgment (NACK) is sent to PSF. PSF then issues message APS923I and performs the action specified in the FAILURE parameter. If the system operator does not need this message, you can disable the timer. PSF then waits for the printer operator to correct the problem.
DPAGELBL (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
DSHDR (optional)	Identifies an OUTPUT statement that specifies the form definition and page definition used to format and print data set header pages. **Syntax: DSHDR**=*.*label* **Note:** Not valid for direct-printing mode.
DUMP (optional)	Specifies that a conditional dump of the PSF address space is to be taken when a specific reason code, restartable abend, or message occurs. **Syntax: DUMP**=([*reascode*|**ABD0***nnn*][,*msgid*,*count*]) **Note:** For a description of the syntax of this parameter and an explanation of restartable abends, refer to *PSF/MVS: Diagnosis Guide and Reference*. To determine whether an abend is restartable, refer to *PSF/MVS: Messages and Codes*.
FAILURE (optional)	Specifies PSF action after a printer failure, an SNA session failure, or a TCP/IP network failure. If FAILURE=STOP, PSF must be restarted by an operator command. If FAILURE=WCONNECT and the printer is connected to another host when PSF attempts to establish connection, PSF waits for the printer (up to the time specified by CONNINTV), as follows: • For SNA, PSF queues a session request in VTAM, which will respond when the printer becomes available. • For TCP/IP, PSF continuously retries until the printer becomes available. **Syntax: FAILURE=STOP|WCONNECT**
FDEFDD (required)	Identifies the DD statement that specifies the form definition libraries. **Syntax: FDEFDD**=*.*label*

Table 2 (Page 5 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	**Description**
FONTDD (required)	Identifies the DD statement that specifies the font libraries. **Syntax: FONTDD**=*.*label*
FORMDEF (required)	Specifies a default form definition. For information on how PSF selects a form definition, see “How PSF Selects Resources” on page 117. **Syntax: FORMDEF**=*fdefname*
IPADDR (required for TCP/IP-attached printers only)	Identifies the Internet Protocol (IP) address of the TCP/IP-attached printer or of the i-data 7913 IPDS Printer LAN attachment. Specify the IP address in the dotted-decimal format, and enclose the value in single quotation marks. Do not specify a host name in place of the dotted-decimal address. **SYNTAX: IPADDR**='*ip-address*' **Example:** `IPADDR='9.99.12.33'`
JOBHDR (optional)	Identifies the OUTPUT statement that specifies the form definition and page definition used to format and print job-header separator pages. **Syntax: JOBHDR**=*.*label* **Note:** Not valid for direct-printing mode.
JOBTRLR (optional)	Identifies the OUTPUT statement that specifies the form definition and page definition used to format and print job-trailer separator pages. **Syntax: JOBTRLR**=*.*label* **Note:** Not valid for direct-printing mode.
LOGMODE (optional)	Specifies the name of the VTAM logon-mode table entry, which defines the session parameters for an SNA-attached printer. (See “Defining the Logon-Mode Table Entry” on page 24 for more information.) Although LOGMODE is an optional parameter, it must not be used if the printer is in a cross-domain system and if the PRINTDEV statement specifies MGMTMODE=DIALIN. (For the PRINTDEV statement, see the discussion of the MGMTMODE parameter on page 58.) If LOGMODE is used, the name must conform to standard JCL naming conventions. If LOGMODE is omitted or if the name is left blank, session parameters must be specified by the DLOGMOD parameter of the LU statement for the printer in the VTAM network definition. **Syntax: LOGMODE**=*name* **Notes:** 1. Only for SNA-attached printers. 2. If LOGMODE is omitted or the name is left blank, the name defaults to one of the following: • The name specified by the DLOGMOD parameter of the related LU statement of the VTAM network definition • The name passed by the secondary LU in the CINIT, if that LU resides in another domain.
LUNAME (required for SNA-attached printers only)	Specifies the unique logical-unit name of an SNA-attached printer. The name must conform to standard JCL naming conventions. The name must also match the printer-node name in the LU statement of the VTAM network definition. For additional information, see “Defining Switched Major Nodes” on page 30. **Syntax: LUNAME**=*name* **Note:** Only for SNA-attached printers (required).

Table 2 (Page 6 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
MESSAGE (optional)	Identifies the OUTPUT statement that specifies the form definition and page definition used to format and print messages. **Syntax: MESSAGE**=*.*label* **Note:** For microfilm devices, see Chapter 21, “Supporting AFP Output on Microfilm.”
MGMTMODE (optional)	Specifies how PSF controls an SNA-attached or TCP/IP-attached printer in deferred-printing mode. MGMTMODE=OUTAVAIL requests that PSF start a communication session with the printer only when output is available on the JES spool. (See note 2 on page 56). MGMTMODE=DIALIN requests that PSF start a session with the printer when the switched line is connected. LOGAPPL must be applied in the PSF network definition. DIALIN is recommended for some printers that use a switched line. For more information about DIALIN, see “PSF PRINTDEV Statement Considerations for Switched Lines” on page 44. For TCP/IP-attached printers, MGMTMODE=DIALIN is identical with MGMTMODE=IMMED. This parameter can also be specified for same-domain leased (nonswitched) lines. PSF waits for LOGAPPL to be triggered—for example, by the printer LU becoming active. MGMTMODE=IMMED requests that PSF start a communication session with this printer immediately. PSF then looks for output available on the JES spool. (See note 2 on page 56). **Syntax: MGMTMODE**=**OUTAVAIL|DIALIN|IMMED** **Note:** Only for SNA-attached and TCP/IP-attached printers.
NPRO (optional)	Specifies whether a nonprocess runout (NPRO) occurs at the end of data set processing. An NPRO forces printed output from the paper path to the stacker. Specify NPRO only when printing on continuous forms. If NPRO is 0 (or if no NPRO parameter is specified), PSF does not initiate an NPRO. If NPRO is 1, PSF initiates an NPRO. **Syntax: NPRO=0|1** **Notes:** 1. This parameter is used only for direct-printing mode with channel-attached printers. 2. In deferred-printing mode, the value for NPRO is derived from the JES NPRO definition. The PSF NPRO parameter described above has no effect. 3. The only valid values are 0 and 1. Any other value will result in job scheduler messages (IEF), indicating that the numeric exceeds the maximum allowed in the NPRO field and that the job failed with a JCL error. 4. In direct-printing mode, a SETPRT macro issued within an application program may cause an NPRO, even though data set processing is not complete. 5. For microfilm devices, see Chapter 21, “Supporting AFP Output on Microfilm.”
OBJCONDD (required on microfilm devices only)	Identifies the DD statement that specifies the object container library, which contains the microfilm setup resources for printing on a microfilm device. For more information, see Chapter 21, “Supporting AFP Output on Microfilm.” **Syntax: OBJCONDD**=*.*label*
OVLYDD (optional)	Identifies the DD statement that specifies the overlay libraries. **Syntax: OVLYDD**=*.*label*
PAGEDEF (required)	Specifies the default page definition. (This page definition is also used if none is specified in JOBHDR, JOBTRLR, DSHDR, and MESSAGE parameters.) For information on how PSF selects a page definition, see “How PSF Selects Resources” on page 117. **Syntax: PAGEDEF**=*pdefname*

Table 2 (Page 7 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
PDEFDD (required)	Identifies the DD statement that specifies the page definition libraries. **Syntax: PDEFDD**=*.*label*
PIMSG (optional)	Controls the handling of PSF messages. Y or YES specifies that all message groups generated in the processing of a data set are to be printed at the end of the data set, preceding any job-trailer separator pages. If a data set is not completed because of an error, message groups generated up to this error are printed, including the message group describing the terminating error. YES is the default. N or NO specifies that no message groups are to be printed unless an error occurs that forces the premature termination of the printing of the data set. If that happens, the message group describing the terminating error is printed. **Note:** Reaching the message count value specified for PIMSG is one error that forces a premature termination of printing. If you do not code a message count value, PSF uses the default value of 16. *nnn* specifies that after PSF has generated *nnn* message groups, it is to end processing of the data set and purge the data set from the spool. The final count of printed messages may be more than *nnn* message groups if the message groups are generated for errors reported by the printer for pages sent before the message count is reached. A value of 0 allows data set printing to continue, regardless of the number of message groups generated, unless a terminating error occurs. The default is 16. **Note:** Messages are printed depending on whether YES or NO is specified in this parameter. For more information on PIMSG, refer to *PSF/MVS: Application Programming Guide*. **Syntax: PIMSG**=(**YES** [,*nnn*] |**NO** [,*nnn*]) **Note:** The PIMSG specified on the OUTPUT statement in the Writer-Procedure for the message data set, referenced by the MESSAGE keyword, also affects the rerouting of messages. For more information, see “Redirecting the Message Data Set” on page 115.
PORTNO (optional)	Specifies the port number with which PSF is to establish a connection. The port number identifies the appropriate internal process in the printer or in the i-data 7913 IPDS Printer LAN Attachment. For TCP/IP-attached printers, the port number must match the TCP/IP port number in the printer configuration. The default port is 5001. **Syntax: PORTNO**=**5001**|*nnnnn*, where *nnnnn* is a value between 5001 to 32767. **Note:** For TCP/IP-attached printers only.
PSEGDD (optional)	Identifies the DD statement that specifies the page segment libraries. **Syntax: PSEGDD**=*.*label*
RELINTV (optional)	Specifies the release interval, in seconds. This is the time period after which a request to release the printer is honored. The timer for the release interval is not started until PSF is at the next job boundary. **Syntax: RELINTV**=*nn*, where *nnn* is a value from 0 to 86400.
RELMODE (optional)	Specifies the manner in which PSF is to respond to a request to release the printer. A request to release a printer can be either an IPDS release dialog request or a VTAM release session request. For more information, see “Printer Sharing in the MVS Environment” on page 60. **Syntax: RELMODE**=**IDLE**|**TIME**|**NONE** **IDLE** specifies to release the printer when (1) a request to release has been received and (2) there is no output on the spool for this printer for the time specified by the release interval. **TIME** specifies to begin the release interval when a request to release has been received, even when there is more output on the spool. **NONE** specifies that the printer is not to be released. This is the default.

Table 2 (Page 8 of 8). Initialization Parameters for the PRINTDEV Statement: Summary

Parameter	Description
SDEFDD (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
SETUP (optional)	Specifies that JES issue operator FORMS setup messages for the printer when the printer is initialized and at the start of any job that specifies a FORMS name different from the name of the form currently active for that printer. If this parameter is not coded, FORMS setup messages are not issued for SNA-attached or TCP/IP-attached printers or for channel-attached printers driven by DPF or PSF Direct. FORMS setup messages are always issued for local non-SNA channel-attached printers. **Syntax: SETUP=FORMS**
SFONTDD (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
SOVLYDD (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
SPAGELBL (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
SPSEGDD (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
SYSAREA (optional)	Refer to *Print Services Facility: Security Guide* for a description of this parameter.
TIMEOUT (optional)	Specifies the action that PSF is to take after a timeout when no output is available from JES. The DISCINTV parameter (described on page 56) specifies the timeout interval. TIMEOUT=REDRIVE requests that PSF redrive the printer FSA according to specifications in the MGMTMODE parameter (described on page 58). TIMEOUT=STOP requests that PSF stop the printer FSA, which can then be restarted only by an operator command. **Syntax: TIMEOUT=REDRIVE|STOP** **Note:** Only for SNA-attached and TCP/IP-attached printers.
TRACE (optional)	Specifies PSF tracing. For more information on using traces, refer to *PSF/MVS: Diagnosis Guide and Reference*. TRACE=YES specifies PSF tracing. TRACE=NO specifies no PSF tracing. **Syntax: TRACE=YES|NO**

Printer Sharing in the MVS Environment

For network-attached printers, PSF/MVS supports the sharing of printers with other print drivers by three different methods:

- JES spool-driven
- VTAM RELREQ
- Port switching

PSF releases a printer only at the end of a job, so that output is never split.

JES Spool-Driven

This method is valid for SNA-attached or TCP/IP-attached printers. A session with the printer is maintained while there is output on the JES spool and the printer is available. When there is no more output on the spool, and the disconnect interval expires, PSF ends the session with the printer. When there is more work on the spool for this printer, PSF attempts to restart the session. After the session is restarted, PSF must reload the resources required for following print jobs. With this method, there is no means for PSF to determine whether another print driver is active.

This method is requested by specifying the following PRINTDEV parameters:

```
MGMTMODE=OUTAVAIL
DISCINTV=nnn
TIMEOUT=REDRIVE
FAILURE=WCONNECT
```

VTAM RELREQ

This method is valid only for SNA-attached printers. A session with a printer is maintained until VTAM indicates that another print driver in the VTAM network has requested a session with the printer. The session with the printer is then ended according to the RELMODE specified. After the session is restarted, PSF must reload any resources required for following print jobs.

If RELMODE=IDLE, PSF ends the session when a VTAM request to release has been received and there is no output on the spool for the printer for the release interval specified. When there is more work on the spool for this printer, PSF attempts to restart the session.

This method is requested by specifying the following PRINTDEV parameters:

- MGMTMODE=OUTAVAIL
- RELMODE=IDLE
- RELINTV=*nn*

If RELMODE=TIME, when a VTAM request to release is received PSF waits for the time specified by the release interval, and then ends the session. PSF attempts to restart the session when the acquire interval expires, in accordance with the MGMTMODE specified.

This method is requested by specifying the following PRINTDEV parameters:

- RELMODE=TIME
- RELINTV=*nn*
- ACQINTV=*nn*

Port Switching

This method is valid for printers that support multiple active ports—for example, the 3130 printer with release 2 microcode. Currently these printers support only SNA or TCP/IP attachment. One of these printers can be in session with multiple print drivers simultaneously. Control of the print mechanism is passed between the ports by means of what is called an IPDS dialog. When an IPDS print driver starts a dialog with the printer, it has control of the print mechanism and can continue to print until it ends its dialog. This is the most efficient method for printer-sharing, because print resources are retained in the printer while the printer is switched to another port. When the printer indicates that another port has requested control, PSF ends its dialog in accordance with the RELMODE specified.

If RELMODE=IDLE, PSF ends the dialog when an IPDS request to release has been received and there is no output on the spool for the printer for the time specified by the release interval. PSF attempts to restart the dialog when there is more work on the spool for this printer.

This method is requested by specifying the following PRINTDEV parameters:

- RELMODE=IDLE
- RELINTV=*nn*

If RELMODE=TIME, when an IPDS request to release is received PSF waits for the specified release interval, and then ends the dialog. PSF attempts to restart the dialog when the acquire interval expires.

This method is requested by specifying the following PRINTDEV parameters:

- RELMODE=TIME
- RELINTV=*nn*
- ACQINTV=*nn*

Note: When a release or acquire interval of 0 is specified, the related action is taken immediately.

Chapter 7. Using Deferred-Printing Mode

PSF supports AFP printers as system output devices for deferred printing under the job entry subsystem (JES). Some system printers are managed by a JES output writer that operates entirely in the JES address space. PSF-attached printers, however, are defined as functional subsystem applications (FSAs) that run under a separate address space called a PSF functional subsystem (FSS). A functional subsystem interface (FSI) maintains communication between the FSS and the JES global address space.

Under JES2, a PSF FSA can start in either of two ways:

- An operator can enter the START DEVICE JES2 command.
- When JES2 is started, PSF starts automatically if the PRT*nnnn* statement for the printer FSA contains the START parameter.

Under JES3, a PSF FSA can start in either of two ways:

- An operator issues the CALL WRITER JES3 command.
- When printing is scheduled for a PSF printer FSA, JES3 starts PSF.

  If you do not want PSF to stop after each job, you can define the PSF FSS as a JES3-called writer, keeping the address space active.

The PSF FSA dynamically allocates a channel-attached printer or establishes a session for an SNA-attached or TCP/IP-attached printer, and manages the printer until the FSA is stopped. The JES operator can control the FSA-managed printer by using JES operator commands.

Note: Stop the printer before you enter JES repositioning commands.

To use a PSF printer as a system output device for deferred printing, you must first do the following:

1. Define one or more PSF FSSs.
2. Define FSAs for each PSF printer. For more information, see “Defining PSF FSSs and FSAs for Printers.”
3. Make sure that you have a startup procedure for each PSF FSS. For more information, see “PSF Startup Procedures for AFP Printers” on page 74.

Note: For configurations that include PSF attached to the Distributed Print Function (DPF) of PSF/2: after PSF has transmitted a data set to the DPF spool, the MVS host software deletes the print data set from the host spool. The host operator cannot control the print data set after it has been transmitted to DPF.

Defining PSF FSSs and FSAs for Printers

To use a PSF printer for deferred printing under JES, you must first code JES initialization statements to define PSF FSSs and FSAs for PSF printers. For JES2, the FSS is defined by an FSS statement, and each printer FSA is defined by a PRT*nnnn* statement. For JES3, the FSS is defined by an FSSDEF statement, and each printer FSA is defined by a DEVICE statement. For information on defining the FSS and printer FSA to JES, see “Sample Initialization Statements for JES2” on page 64 and “Sample Initialization Statements for JES3” on page 69.

The following number of FSSs and FSAs are supported:

- A maximum of 2000 FSSs
- A maximum of 64 FSAs per FSS (see note)

Note: The actual number of FSAs per FSS depends on several factors, some of which are as follows:

- The virtual region size below the 16-megabyte line
- The number and the size of form definitions and page definitions
- The number of buffers specified (BUFNO) for channel-attached printers
- The block sizes of the font, page segment, overlay, form definition, and page definition libraries

For more information, see "Selecting Region Sizes" on page 114.

Understanding the Syntax Notation Used in This Book

The following rules apply to syntax examples and coding illustrations throughout this chapter:

- The uppercase letters are to be coded as shown.
- The variable data you must provide is printed in italics. Enter specific data to replace the characters in italics; for example, for PRT*nnnn* you could enter PRT*2*.
- Do not enter any of the following symbols as part of a parameter or an option:

  Vertical bar |
  Underscore __
  Braces {}
  Brackets []

- A vertical bar between two values means that you are to select only one of the values.
- If one value in a set of values is underscored, that value is the default—that is, the value that PSF uses if no option is specified.
- If values are enclosed in braces {}, one or the other of the values in the braces must be selected.
- If a value is enclosed in brackets [], you do not have to select that value.

Sample Initialization Statements for JES2

Figure 25 on page 65 shows sample JES2 statements for a channel-attached | printer (PRT1), an SNA-attached printer (PRT2), and a TCP/IP-attached printer | (PRT3).

Note: These are samples only. For specific coding for JES parameters, refer to the JES publication for the level of JES you are using.

```
FSS(FSS1) PROC=SAMPPROC,HASPFSSM=HASPFSSM
PRT1      FSS=FSS1,MODE=FSS,PRMODE=(LINE,PAGE,SOSI1),
          UNIT=00E,CLASS=A,UCS=0,SEP=YES,SEPDS=YES,CKPTPAGE=100,
          START=NO,MARK=YES,NPRO=99,TRKCELL=YES
PRT2      FSS=FSS1,MODE=FSS,PRMODE=(LINE,PAGE,SOSI1),
          CLASS=B,UCS=0,SEP=YES,SEPDS=NO,CKPTPAGE=100,
          START=NO,MARK=YES
PRT3      FSS=FSS1,MODE=FSS,PRMODE=(LINE,PAGE,SOSI1),
          CLASS=C,UCS=0,SEP=YES,SEPDS=NO,CKPTPAGE=100,
          START=NO,MARK=YES,TRKCELL=YES
```

Figure 25. Sample Initialization Statements for JES2, for Three Printers. These statements are commonly found in SYS1.PARMLIB (initialization deck member name).

Parameters for the JES2 Initialization Statement

The following sections describe the JES2 initialization statements shown in Figure 25, which are typical of the JES2 statements used with PSF. For more detailed descriptions of all JES2 initialization statements, refer to the JES2 initialization and tuning publication for your operating system.

The descriptions below tell which statements and parameters are required. In the programming syntax shown for each parameter, defaults are underscored.

Note: When coding JES initialization statements, consider the characteristics of each printer. For example, BURST applies only to continuous-forms printers with a burster-trimmer-stacker installed.

FSS Statement: The FSS initialization statement is optional but recommended. If it is omitted, JES2 generates a default FSS for that device. The FSS initialization statement defines an FSS to JES2. The FSS statement is associated with one or more PRT*nnnn* statements that define each printer FSA. For a description of all FSS parameters, refer to the JES2 initialization and tuning publication for your operating system. The FSS statement contains the following parameters for PSF:

FSS(*fsaname*)
Specifies the unique 1- to 8-character name of a particular FSS. This name is referenced in the PRT*nnnn* statement for each printer FSA that is associated with this FSS. When you start the first printer that has an FSA definition for that FSS, an FSS address space is created for PSF. PSF manages this FSS and the printer FSA for the first printer. If you start a second printer with an FSA definition for the same FSS, PSF manages a separate FSA for the second printer in that FSS.

PROC
Specifies a procedure for starting the PSF FSS. The procedure, which must be defined before that FSS is started, is a member of either SYS1.PROCLIB or a library concatenated to SYS1.PROCLIB. For more information, see "PSF Startup Procedures for AFP Printers" on page 74.

HASPFSSM
Specifies the 1- to 8-character name of the load module that is loaded into the PSF FSS address space. This load module contains the various FSI service routines that JES2 supplies. For PSF, specify the default value (HASPFSSM=HASPFSSM).

***PRTnnnn* Statement:** A PRT*nnnn* statement, which is required to define each printer *nnnn* statement FSA, is associated with an FSS statement. Each printer FSA should have a unique PRT*nnnn* name that must match the label on the CNTL, ENDCNTL, and PRINTDEV statements for the PSF startup procedure specified in the PROC parameter. For more information, see Figure 28 on page 76.

Note: The label on the PRINTDEV statement must be 8 characters or fewer. PSF accepts the PRT*nnnn*, PRINT*nn*, and PRINTER*n* formats. PRT*nnnn* is the recommended format.

The PRT*nnnn* parameters create JES2 default values that are used unless other values are specified in the application program JCL. The PRT*nnnn* statement contains the following parameters:

BURST=YES|NO
Specifies whether to thread the paper to a burster-trimmer-stacker (BTS) for bursting and offset stacking, or to send the paper to a continuous-forms stacker (CFS) for stacking. This parameter applies only to continuous-forms printers such as the 3800 or 3900 printer with a post-processing device installed.

Notes:

1. With a 3800 printer, PSF always tells JES to issue burst setup messages for jobs with BURST specified, regardless of whether a BTS is installed. With a 3900 printer, however, PSF tells JES to issue burst setup messages *only* if a BTS is specified as installed when the FSA is started. If a BTS is installed or removed while PSF is active, PSF cannot tell JES to issue (or not to issue) burst setup messages. To receive JES burst setup messages, always specify that a BTS is installed on the 3900 printer whenever you first start the printer. If a BTS is installed or removed, you must restart the FSA to receive (or not receive) JES burst setup messages.
2. This parameter is new in JES2; it replaces the BURST|NOBURST parameter.

CKPTMODE=PAGE|SEC
Specifies whether checkpointing is to be done by page count (PAGE) or by time (SEC). If you specify both CKPTPAGE and CKPTSEC, the CKPTMODE parameter determines which value is used. When printing to microfilm output, specify CKPTMODE=SEC. For additional information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.”

Note: This parameter is new in Version 4 of JES2. It replaces the PAGECKPT and TIMECKPT parameters.

CKPTPAGE
Specifies the number of pages between data-set checkpoints. If checkpoint intervals are too frequent, printer performance is significantly reduced. If intervals are too infrequent, and a PSF error occurs, the job has to be reprocessed from the last checkpoint. In this case, printer performance is significantly reduced if PSF errors continue to occur. For printing to paper output, IBM recommends a starting value of 100. For printing to microfilm output, specify CKPTSEC instead of CKPTPAGE. For additional information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.”

CKPTSEC
Specifies the seconds between checkpoints in the data set. When printing to microfilm output, specify a CKPTSEC keyword instead of CKPTPAGE. For

additional information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.”

CLASS

Specifies the output classes processed by the printer. You may want to consider the handling of disabled mechanisms by defining print classes.

COPYMARK=DATASET|JOB|CONSTANT

Specifies how a printer is to increment copy marks or offset stacking.

DATASET The printer is to increment the copy marks or offset stacking on a data set boundary.

JOB The printer is to increment the copy marks or offset stacking on a job boundary.

CONSTANT Copy marks are constant, and no offset stacking is done.

For more information on copy marking, see “Copy Marking and Offset Stacking” on page 91.

FCB

Specifies the name of the default JES2 page definition. If this parameter is not included, the default page definition can be specified by an NIFCB parameter in the JES2 PRINTDEF statement.

Note: If you want to select a page definition from the PSF startup procedure instead of the JES2 default, do not select an FCB parameter on the PRT*nnnn* statement or an NIFCB parameter in the JES2 PRINTDEF statement.

FLASH

Specifies the name of the default JES forms flash for a 3800 printer. If this parameter is not included, the default forms flash can be specified by an NIFLASH parameter on the JES2 PRINTDEF statement.

Note: If you want to use the FLASH parameter specified in the form definition specified in the PRINTDEV statement, do not define a FLASH parameter in the JES2 PRINTDEF statement.

FSS

Specifies the FSS for this device and must match the FSA name specified on the corresponding FSS statement. This parameter is required.

MARK=YES|NO

Specifies that forms marking is to be placed on job trailer pages. If MARK=NO is specified, no forms marking is placed on trailer pages. For the 3800 printer only, if MARK=NO is specified, the printer marks the carrier strip rather than the forms. For more information, see “Forms Marking” on page 88 and “Carrier-Strip Marking” on page 90.

Note: This parameter is new in JES2; it replaces the MARK|NOMARK parameter.

MODE

Specifies that the printer is managed by an FSS (MODE=FSS). This parameter is required.

Note: For a 3800 printer running in 3800-1 compatibility mode, specify MODE=JES.

NPRO

Specifies the interval after which output in the paper path is forced to the stacker with a nonprocess run out. When printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm,” for more information.

This parameter is used only for continuous-forms printers and microfilm devices. If you specify both CKPTPAGE and CKPTSEC, JES2 uses the CKPTSEC value unless you also specify PAGECKPT.

Note: When printing to microfilm, use the CKPTSEC keyword. For more information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.”

PRMODE

Specifies the data-set processing modes supported for a printer. List all the PRMODE values that the printer is to accept.

If PRMODE is designated as a JES work-selection (WS) criterion for output selection, only jobs with a PRMODE that matches one of the following values with special meanings is selected for processing by the printer:

LINE Specifies that line-format data sets are to be printed

PAGE Specifies that composed-page data sets can be printed

SOSI*n* Specifies a shift-out/shift-in processing mode of SOSI1 or SOSI2 for double-byte fonts

PRMODE can be assigned to a print job on the JCL OUTPUT statement. If PRMODE is not coded, JES examines the print records and assigns a value of PAGE to data sets that contain structured fields, or LINE when data sets do not contain structured fields. For this reason, if PRMODE is designated as a JES work-selection (WS) criteria, specify both LINE and PAGE for printers on which both line-format and composed-page data sets are to be printed.

For a print job that uses double-byte fonts, you must specify a PRMODE of either SOSI1 or SOSI2. If your installation uses double-byte fonts, specify the appropriate SOSI*n* values for the printers that support double-byte fonts. If PRMODE is designated as a JES work-selection criterion, all other PRMODE values that you have defined in your installation to control job processing must be specified for printers. For more information about the PRMODE parameter, refer to *Print Services Facility/MVS: Application Programming Guide*.

PRESELCT=YES|NO

Specifies whether jobs are preselected for printers.

YES Specifies that two output elements or jobs are to be selected for each printer. This means that while one job is printing, a second job is also assigned to that printer. This preselected job is not available for printing on any other printer.

NO Specifies that only one output element or job are to be selected for each printer.

| **SEP=YES|NO**

SEP=YES specifies that job-header and job-trailer separator pages are to be produced; SEP=NO specifies that no separator pages are to be produced.

Notes:

1. If the Distributed Print Function (DPF) of PSF/2 is installed, you may need to specify SEP=YES. For more information, see "Distributed Print Function (DPF) Considerations" on page 87.

   For microfilm devices, you should specify the SEP=NO parameter.

2. This parameter is new in JES2; it replaces the SEP|NOSEP parameter.

SEPDS=YES|NO

SEPDS=YES specifies that formatted data-set header separator pages are to be produced; SEPDS=NO specifies that no formatted data-set header separator pages are to be produced.

Notes:

1. For microfilm devices, you should specify the SEPDS=NO parameter.
2. This parameter is new in JES2; it replaces the SEPDS|NOSEPDS parameter.

START=YES|NO

START=NO specifies that the printer is to be started by operator command. START=YES specifies that the printer (if it is ready) is to start automatically when JES2 begins processing. If you specify START=YES, and the printer is unavailable when JES2 is started, the printer does not start unless you also specify the UNIT parameter to specify a unit address.

Note: This parameter is new in JES2; it replaces the DRAIN|START parameter.

TRKCELL=YES|NO

Specifies whether track-cell despooling is to be used with this printer. TRKCELL=NO (the default) specifies that track-cell despooling is not used. TRKCELL=YES is recommended.

UCS

Specifies a default font. A table showing how fonts are selected in PSF is given in "How PSF Selects Special Processing Parameters" on page 113.

Note: To select a font in the startup procedure instead of the JES2 default, code UCS=*0* on the PRT*nnnn* statement.

UNIT

Specifies the real device address (unit address) of the printer. If you enter START and specify a unit address, the printer starts automatically when JES2 begins processing, even if the printer has been offline during the initialization of JES. This parameter applies only to host-connected, channel-attached non-SNA printers.

Sample Initialization Statements for JES3

Figure 26 on page 70 shows sample JES3 statements for a host-connected, channel-attached, non-SNA printer (JNAME=PRT1), an SNA-attached printer (JNAME=PRT2), and a TCP/IP-attached printer (JNAME=PRT3).

For a host-connected, channel-attached, non-SNA printer:

```
FSSDEF,TYPE=WTR,FSSNAME=FSS1,PNAME=SAMPPR01
DEVICE,DTYPE=PRTAFP1,JNAME=PRT1,JUNIT=(00E,SYS1,UR,ON),
       XTYPE=(PRT1,UR),XUNIT=(00E,SYS1,UR,OFF),FSSNAME=FSS1,
       MODE=FSS,PM=(LINE,PAGE,SOSI1),CHARS=(YES,GT12),DYN=(NO),
       CARRIAGE=(YES,A868),CKPNTPG=100,HEADER=YES,
       BURST=(YES,M),NPRO=99,WC=(A),FORMS=(NO,NARROW),FEATURES=(SS),
       STACKER=(NO,S)
```

For an SNA-attached printer:

```
FSSDEF,TYPE=WTR,FSSNAME=FSS2,PNAME=SAMPPR02,SYSTEM=SYS1,TERM=NO
DEVICE,DTYPE=PRTAFP1,JNAME=PRT2,JUNIT=(,SYS1,,OFF),FSSNAME=FSS2,
       MODE=FSS,PM=(LINE,PAGE,SOSI1),CHARS=(YES,GT12),
       CARRIAGE=(YES,A868),CKPNTPG=100,HEADER=YES,WC=(B)
```

For a TCP/IP-attached printer:

```
FSSDEF,TYPE=WTR,FSSNAME=FSS3,PNAME=SAMPPR03,SYSTEM=SYS1,TERM=NO
DEVICE,DTYPE=PRTAFP1,JNAME=PRT3,JUNIT=(,SYS1,,OFF),FSSNAME=FSS3,
       MODE=FSS,PM=(LINE,PAGE,SOSI1),CHARS=(YES,GT12),
       CARRIAGE=(YES,A868),CKPNTPG=100,HEADER=YES,WC=(C)
```

Figure 26. Sample Initialization Statements for JES3, for Three Printers. These statements are commonly found in SYS1.PARMLIB (initialization deck member name).

Figure 27 shows sample JES3 statements for running multiple FSAs under a single FSS.

```
FSSDEF,TYPE=WTR,FSSNAME=J3PSFSW,PNAME=J3PSFSW
DEVICE,DTYPE=PRTAFP1,JNAME=PRT12,XTYPE=(PRM,UR),CKPNTPG=5,
DGROUP=LOCAL,JUNIT=(390,SYS1,UR,OFF),XUNIT=(390,SYS1,UR,ON),
FSSNAME=J3PSFSW,MODE=FSS,PM=(LINE,PAGE),WS=(CL),WC=(M)
DEVICE,DTYPE=PRTAFP1,JNAME=PRT17,CKPNTPG=5,
DGROUP=VOYAGER,JUNIT=(,SYS1,,OFF,,SYS2,,OFF),
FSSNAME=J3PSFSW,MODE=FSS,PM=(LINE,PAGE),WS=(CL),WC=(Y)
```

Figure 27. Sample Initialization Statements for JES3. These statements are used for running multiple FSAs under a single FSS. Continuation characters are not shown.

Parameters for the JES3 Initialization Statement

The following sections describe the JES3 initialization statements shown in Figure 26, which are typical of the JES3 statements used with PSF. For more detailed descriptions of all the JES3 initialization statements, refer to the JES3 initialization and tuning publication for your operating system.

The following descriptions show which statements and parameters are required. In the programming syntax shown for each parameter, defaults are underscored.

Note: When coding JES initialization statements, keep the individual characteristics of each printer in mind. For example, STACKER=S applies only to printers using continuous forms with a burster-trimmer-stacker installed.

FSSDEF Statement: The FSSDEF initialization statement is optional but recommended. If it is omitted, JES3 generates a default FSS for the printer being invoked.

The FSSDEF statement can contain the following parameters:

TYPE
Specifies that the FSS is an output writer for deferred printing (TYPE=WTR). This parameter is required.

FSSNAME
Specifies the unique name of a particular FSS. This parameter is required.

PNAME
Specifies a procedure for starting a specific PSF FSS. The procedure, which must be defined before that FSS is started, is a member of the procedure library defined by the STCPROC parameter of the STANDARDS statement or of the IATPLBST procedure library (the default). Different FSSDEF initialization statements can refer to the same startup procedure.

The PNAME parameter can specify either a startup procedure supplied with PSF, or one of your organization's procedures. For more information about the startup procedures supplied with PSF, refer to *Print Services Facility/MVS: Program Directory*. In the examples shown in Figure 26 on page 70, SAMPPRO2 is the PNAME for the host-connected, channel-attached, non-SNA printer, and SAMPPRO3 is the PNAME for the SNA-attached printer. SAMPPRO1 and SAMPPRO2 are shown in Figure 29 on page 78.

SYSTEM
Specifies the JES3 processor on which the FSS is to run. The name must be the same as that specified on the NAME parameter of the MAINPROC statement for the processor.

TERM=YES|NO
YES specifies that the FSS is to be terminated if the JES3 global address space terminates by a *RETURN or *DUMP operator command.

DEVICE Statement: A DEVICE statement is required for each printer. The DEVICE parameters create JES3 default values that are used unless other values are specified in the JCL application program. The DEVICE statement can contain the following parameters:

BURST=(YES|NO [,M])
Specifies whether or not trailer pages are printed and marked.

YES Specifies that trailer pages are to be be printed.

NO Suppresses the printing of trailer pages.

M Specifies that the edges of the trailer pages are to be marked.

Notes:

1. This parameter is not related to the burster-trimmer-stacker (BTS) feature installed on continuous-forms printers. To specify burster-trimmer-stacker options, see the description of the STACKER parameter on page 74.
2. If DPF is installed, you must specify BURST=(YES). For more information, see “Distributed Print Function (DPF) Considerations” on page 87.

CARRIAGE=(YES|NO,*aaaa*)
Specifies the JES default page definition.

YES Specifies that the page definition can be changed during startup procedure.

NO Specifies that the page definition cannot be changed during the startup procedure.

aaaa Specifies that the un-prefixed name (1 to 4 characters) of the page definition is to be used as a default. The name specified here, or the JES3 system default, is the PSF default page definition. For information on selecting PSF defaults, see “How PSF Selects Special Processing Parameters” on page 113.

CHARS

Specifies a default JES3 font. For a chart showing how fonts are selected in PSF, see “How PSF Selects Special Processing Parameters” on page 113.

CKPNTPG

Specifies the number of pages between data-set checkpoints. Specify no more frequent checkpoints than you actually need in order to monitor a print job. If checkpoint intervals are too frequent, printer performance is significantly reduced. If intervals are too few, and a PSF error occurs, the job will need to be reprocessed from the last checkpoint. For printing to paper, a starting value of 100 is recommended. For printing to microfilm output, specify a CKPNTSEC keyword instead of CKPNTPG. For additional information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.”

CKPNTSEC

Specifies the seconds between data-set checkpoints.

Note: For printing to microfilm output, specify the CKPNTSEC parameter. For additional information about printing to microfilm, see Chapter 21, “Supporting AFP Output on Microfilm.”

COPYMARK=<u>C</u>|N|J

Specifies that you want vertical separator lines printed on the edges of the output. These separator lines help identify the beginning and end of each job or data set in a stack of output. A change in the thickness of the copy mark indicates the beginning of a new job, a new data set, or one of multiple copies of a data set.

C Specifies that copy marks are to be printed on each copy of a data set. For multiple copies of a data set, JES3 increments the thickness of the copy mark after each copy.

N Specifies that you do not want to use copy marks. Although JES3 continues to print a vertical line on the edges of output from 3800 printers, the thickness does not vary.

J Specifies that you want copy marks printed on a job basis. A change in the thickness of the copy mark indicates the beginning of a new job.

DTYPE

Identifies the printer device type, as follows:

- DTYPE=PRT38003 for a 3800 Model 3 printer
- DTYPE=PRT38003 for a 3800 Model 6 printer
- DTYPE=PRT38008 for a 3800 Model 8 printer
- DTYPE=PRT3820 for a 3820 printer
- DTYPE=PRTAFP1 for all other printers and microfilm devices

This parameter is required.

DYNAMIC=YES|NO

Specifies whether you want this device started dynamically when work is available to print, to punch, or to be sent to a remote printer or punch. For all devices except those that run under the control of an output writer FSS, the default is YES. For devices that run under the control of an output writer FSS, the default is NO. For more information, refer to the *JES3 Installation and Tuning* publication for your operating system, listed in Appendix K, “Related Publications.”

FEATURES

FEATURES=SS specifies that the printer has a burster-trimmer-stacker.

FLASH

Specifies whether forms flash is used (3800 printer only), and specifies the name of the forms overlay to be used.

Note: If you want to use the FLASH parameter specified in the form definition specified in the PSF PRINTDEV statement, do not define a FLASH parameter in the JES3 DEVICE statement.

FORMS

Specifies whether forms can be changed between print jobs; specifies the initial form loaded in the printer.

FSSNAME

Specifies a unique FSS for this printer DEVICE statement. The value must match the value coded for the FSSNAME parameter in the corresponding FSSDEF statement.

HEADER=YES|NO

Specifies if job and data set header pages are printed.

For microfilm devices, you should specify HEADER=NO.

Note: If DPF is installed, you may need to specify HEADER=YES. For more information, see “Distributed Print Function (DPF) Considerations” on page 87.

JNAME

Specifies the name of the printer FSA. Each printer FSA should have a unique JNAME. This unique JNAME is 8 characters or fewer and must correspond to its label on the CNTL, ENDCNTL, and PRINTDEV statements for the PSF startup procedure specified in the PNAME parameter. For more information, see Figure 29 on page 78. This parameter is required.

JUNIT

Specifies:

- The device address (host-connected, channel-attached, non-SNA printers only)
- The name of the processor to which the device is attached
- A destination class for messages about the device
- Whether the device is initially online or offline

This parameter is required.

| **Note:** Do not specify a device address for SNA-attached or TCP/IP-attached
| printers.

MODE=FSS

Specifies that the printer is managed by an FSS. This parameter is required.

NPRO

Specifies the interval after which output in the paper path is forced to the stacker with a nonprocess runout.

This parameter is used only for continuous-forms printers.

Note: For printing to microfilm, a value of zero is recommended. For more information, see Chapter 21, “Supporting AFP Output on Microfilm.”

PM

Specifies which data-set processing mode is supported. PM=LINE specifies a printer that can print line format data sets. PM=PAGE specifies a printer that can print MO:DCA-P data sets. PM=SOSI1 or PM=SOSI2 specifies a printer that can process double-byte fonts. For more information about SOSI processing, refer to *Print Services Facility/MVS: Application Programming Guide.*

STACKER

Specifies whether output stacking can change, and whether the paper is to be threaded to the burster-trimmer-stacker for bursting and offset stacking (S) or sent to the continuous-forms stacker (C). This parameter is only for a continuous-forms printer with a burster-trimmer-stacker installed.

WC

Specifies the print output class that the printer is to process. You may want to consider handling disabled mechanisms by defining print classes. For more information, see “Disabled Mechanisms” on page 94.

XTYPE

Specifies the characteristics of the FSS-managed printer as it is used by jobs in execution. This parameter is valid only for channel-attached printers, and is required if the JUNIT parameter specifies a device address. XTYPE, if used, must precede the XUNIT parameter.

| **Note:** XTYPE is not valid for SNA-attached or TCP/IP-attached printers.

XUNIT

Specifies the device address, the name of the processor to which the device is attached, a destination code for messages about the device, and the initial status of the device—online or offline. It also specifies whether the device is initially online or offline.

This parameter is valid only for channel-attached printers, and is required if the JUNIT parameter specifies a device address.

| **Note:** XUNIT is not valid for SNA-attached or TCP/IP-attached printers.

PSF Startup Procedures for AFP Printers

Before starting a PSF FSS, you must have a cataloged startup procedure in SYS1.PROCLIB or any other system procedure library. This procedure specifies PSF initialization parameters and libraries that contain system and installation resources. The name of this procedure can be specified in the FSSDEF statement of the JES initialization deck. If the name omitted, JES3 supplies a default, chosen for the printer defined.

Several PSF startup procedures are supplied with PSF. For information about them, refer to *Print Services Facility/MVS: Program Directory*. You can modify the startup procedures supplied with PSF, or write your own startup. For sample JES2 and JES3 PSF startup procedures, see “Sample PSF Startup Procedures” on page 75. Publications on the initialization and tuning of JES are listed in Appendix K, “Related Publications”; these publications contain details about procedure libraries.

When you install PSF, you must create or update the required libraries referenced by the startup procedure, and you should consider any need to increase the region size, as specified in the EXEC statement, to accommodate resources and an increased number of printer FSAs.

The startup procedure can also specify defaults that cannot be set with JESinitialization statements for printer FSA definitions. For more information on JES defaults, see “Defining PSF FSSs and FSAs for Printers” on page 63 and “How PSF Selects Special Processing Parameters” on page 113.

Sample PSF Startup Procedures

Note: *Print Services Facility/MVS: Program Directory* includes source startup procedures for many AFP printers. The samples are not identical with the ones provided in this publication. Samples can be edited to produce your own startup procedures.

Figure 28 on page 76 shows a sample startup procedure for three FSSs, each with a single printer FSA. PRT1 is a channel-attached printer. PRT2 is an SNA-attached printer. PRT3 is any supported TCP/IP-attached printer, including any printer attached by use of the i-data IPDS Printer LAN attachment.

The JES2 initialization for this FSS is shown in Figure 25 on page 65. The JES2 initialization statements must have the following relationships with the associated PSF startup procedure:

- The name of the PSF startup procedure must be specified in the PROCoption of the FSS statement. For this example, the procedure name is SAMPPROC.
- In the PSF startup procedure, the labels for the CNTL, PRINTDEV, and ENDCNTL statements must match the JES2 PRT*nnnn* statements associated with the FSS.

Figure 29 on page 78 shows a sample startup procedure for two FSSs, each with a single printer FSA. PRT1 FSA is a channel-attached printer. PRT2 is an SNA-attached printer.

The JES3 initialization statements for these FSSs and printer FSAs are shown in Figure 26 on page 70. The JES3 initialization statements must have the following relationships with the associated PSF startup procedure:

- The name of the PSF startup procedure must be specified in the PNAME parameter of the FSSDEF statement. In Figure 26 on page 70, the FSSDEF statement named FSS1 specifies procedure SAMPPRO1; FSS2 specifies procedure SAMPPRO2.
- In the PSF startup procedure, the labels for the CNTL, PRINTDEV, and ENDCNTL statements must match the JNAME values in the JES3 DEVICE statements associated with the FSS.

The following are examples showing how startup procedures could be modified, and do not necessarily use the recommended values. For recommended values, refer to SYS1.PROCLIB.

```
//SAMPPROC PROC
//* THE FOLLOWING STARTUP PROCEDURE DEFINES THE JCL AND PRINTDEV
//* STATEMENTS FOR THREE DIFFERENT PSF-SUPPORTED PRINTERS.
//*
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K,TIME=1440
//STEPLIB  DD DSN=PSF.LINKLIB,DISP=SHR
//JOBHDR   OUTPUT PAGEDEF=V06483,       /* JOB HEADER PAGE           */
//         FORMDEF=A10120,CHARS=GT12    /*   FORMDEF: ALTERNATIVE BIN*/
//JOBTLR   OUTPUT PAGEDEF=V06483,       /* JOB TRAILER PAGE          */
//         FORMDEF=A10110,CHARS=GT12    /*   FORMDEF: MAIN BIN       */
//DSHDR    OUTPUT PAGEDEF=V06483,       /* DATA SET SEPARATOR        */
//         FORMDEF=A10110,CHARS=GT12    /*   FORMDEF: MAIN BIN       */
//MSGDS    OUTPUT PAGEDEF=A08682,       /* MESSAGE DATA SET          */
//         FORMDEF=A10110,CHARS=GT15    /*                           */
//*
//FONT01 DD DSN=SYS1.FONTLIB,DISP=SHR   /* SYSTEM FONTS              */
//       DD DSN=INST.FONTLIB,DISP=SHR   /* INSTALLATION USER FONTS   */
//*
//PSEG01 DD DSN=INST.PSEGLIB,DISP=SHR   /* INSTALLATION PAGE SEGMENTS*/
//       DD DSN=SPEC.PSEGLIB,DISP=SHR   /* SPECIAL PAGE SEGMENTS     */
//*
//PSEG02 DD DSN=INST.PSEGLIB,DISP=SHR   /* INSTALLATION PAGE SEGMENTS*/
//*
//OLAY01 DD DSN=INST.OVERLIB,DISP=SHR   /* INSTALLATION OVERLAYS     */
//*
//PDEF01 DD DSN=SYS1.PDEFLIB,DISP=SHR   /* SYSTEM PAGE DEFINITIONS   */
//       DD DSN=INST.PDEFLIB,DISP=SHR   /* INSTALLATION PAGE DEFS    */
//*
//FDEF01 DD DSN=SYS1.FDEFLIB,DISP=SHR   /* SYSTEM FORM DEFINITIONS   */
//       DD DSN=INST.FDEFLIB,DISP=SHR   /* INSTALLATION FORM DEFS    */
//*
//PRT1     CNTL
//PRT1     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD          */
//          OVLYDD=*.OLAY01,        /* OVERLAY  LIBRARY DD          */
//          PSEGDD=*.PSEG01,        /* SEGMENT  LIBRARY DD          */
//          PDEFDD=*.PDEF01,        /* PAGEDEF  LIBRARY DD          */
//          FDEFDD=*.FDEF01,        /* FORMDEF  LIBRARY DD          */
//          JOBHDR=*.JOBHDR,        /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,       /* JOB TRAILER SEPARATOR OUTPUT */
//          DSHDR=*.DSHDR,          /* DATA SET HEADER SEPARATOR    */
//          MESSAGE=*.MSGDS,        /* MESSAGE DATA SET OUTPUT      */
//          BUFNO=5,                /* NUMBER OF WRITE DATA BUFFERS */
//          PAGEDEF=A08682,         /* DEVICE PAGEDEF DEFAULT       */
//          FORMDEF=A10110,         /* DEVICE FORMDEF DEFAULT       */
//          CHARS=GT12,             /* DEFAULT FONT                 */
//          PIMSG=(YES,16),         /* ACCUMULATE DATA SET MESSAGES */
//          DATACK=UNBLOCK,         /* UNBLOCK DATA CHECKS          */
//          TRACE=NO,               /* BUILD INTERNAL TRACE         */
//          SETUP=FORMS             /* SETUP MESSAGE                */
//PRT1      ENDCNTL
//*
```

Figure 28 (Part 1 of 2). Sample PSF Startup Procedure for Three Printers, Showing Multiple FSAs

```
//PRT2     CNTL
//PRT2     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD          */
//          OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//          PSEGDD=*.PSEG02,         /* SEGMENT  LIBRARY DD          */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD          */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD          */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR OUTPUT */
//          MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT      */
//          PAGEDEF=A08682,          /* DEVICE PAGEDEF DEFAULT       */
//          FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT       */
//          CHARS=GT12,              /* DEFAULT FONT                 */
//          PIMSG=(YES,16),          /* ACCUMULATE DATA SET MESSAGES */
//          DATACK=BLOCK,            /* BLOCK DATA CHECKS            */
//          TRACE=NO,                /* BUILD INTERNAL TRACE         */
//          FAILURE=WCONNECT,        /* VTAM: ATTEMPT RECONNECT      */
//          DISCINTV=0,              /* VTAM: NO TIME OUT            */
//          MGMTMODE=IMMED,          /* VTAM: MAINTAIN SESSION       */
//          APPLID=name,             /* VTAM: APPLICATION PGM NODE   */
//          LUNAME=name              /* VTAM: LOGICAL UNIT NODE      */
//PRT2      ENDCNTL
//*
//PRT3     CNTL
//PRT3     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD          */
//         OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//         PSEGDD=*.PSEG02,         /* SEGMENT  LIBRARY DD          */
//         PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD          */
//         FDEFDD=*.FDEF-1,         /* FORMDEF  LIBRARY DD          */
//         JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT  */
//         JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR OUTPUT */
//         MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT      */
//         PAGEDEF=A08682,          /* DEVICE PAGEDEF DEFAULT       */
//         FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT       */
//         CHARS=GT12,              /* DEFAULT FONT                 */
//         PIMSG=(YES,16),          /* ACCUMULATE DATA SET MESSAGES */
//         DATACK=BLOCK,            /* BLOCK DATA CHECKS            */
//         TRACE=NO,                /* BUILD INTERNAL TRACE         */
//         FAILURE=WCONNECT,        /* ATTEMPT RECONNECT            */
//         DISCINTV=0,              /* NO TIME OUT                  */
//         MGMTMODE=IMMED,          /* MAINTAIN SESSION             */
//         IPADDR='9.99.12.33'      /* IP ADDRESS OF PRINTER        */
//PRT3     ENDCNTL
```

Figure 28 (Part 2 of 2). Sample PSF Startup Procedure for Three Printers, Showing Multiple FSAs

For a Channel-Attached Printer:

```
//         MESSAGE= *.MSGDS,        /*MESSAGE DATA SET OUTPUT       */
//SAMPPRO1 PROC
//* THE FOLLOWING STARTUP PROCEDURE DEFINES THE JCL AND PRINTDEV
//* STATEMENTS FOR PRT1, A CHANNEL-ATTACHED PRINTER.
//*
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K
//STEPLIB  DD DSN=PSF.LINKLIB,DISP=SHR
//JOBHDR OUTPUT PAGEDEF=V06483,         /* JOB HEADER PAGE           */
//       FORMDEF=A10120,CHARS=GT12      /*   FORMDEF: ALTERNATIVE BIN*/
//JOBTLR OUTPUT PAGEDEF=V06483,         /* JOB TRAILER PAGE          */
//       FORMDEF=A10110,CHARS=GT12      /*   FORMDEF: MAIN BIN       */
//DSHDR  OUTPUT PAGEDEF=V06483,         /* DATA SET SEPARATOR        */
//       FORMDEF=A10110,CHARS=GT12      /*   FORMDEF: MAIN BIN       */
//MSGDS  OUTPUT PAGEDEF=A08682,         /* MESSAGE DATA SET          */
//       FORMDEF=A10110,CHARS=GT15      /*                           */
//*
//FONT01 DD DSN=SYS1.FONTLIB,DISP=SHR   /* SYSTEM FONTS              */
//       DD DSN=INST.FONTLIB,DISP=SHR   /* INSTALLATION USER FONTS   */
//*
//PSEG01 DD DSN=INST.PSEGLIB,DISP=SHR   /* INSTALLATION PAGE SEGMENTS*/
//       DD DSN=SPEC.PSEGLIB,DISP=SHR   /* SPECIAL PAGE SEGMENTS     */
//*
//OLAY01 DD DSN=INST.OVERLIB,DISP=SHR   /* INSTALLATION OVERLAYS     */
//*
//PDEF01 DD DSN=SYS1.PDEFLIB,DISP=SHR   /* SYSTEM PAGE DEFINITIONS   */
//       DD DSN=INST.PDEFLIB,DISP=SHR   /* INSTALLATION PAGE DEFS    */
//*
//FDEF01 DD DSN=SYS1.FDEFLIB,DISP=SHR   /* SYSTEM FORM DEFINITIONS   */
//       DD DSN=INST.FDEFLIB,DISP=SHR   /* INSTALLATION FORM DEFS    */
//*
//PRT1     CNTL
//PRT1     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD          */
//          OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//          PSEGDD=*.PSEG01,         /* SEGMENT  LIBRARY DD          */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD          */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD          */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR OUTPUT */
//          DSHDR=*.DSHDR,           /* DATA SET HEADER SEPARATOR    */
//          MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT      */
//          BUFNO=5,                 /* NUMBER OF WRITE DATA BUFFERS */
//          PAGEDEF=A08682,          /* DEVICE PAGEDEF DEFAULT       */
//          FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT       */
//          CHARS=GT12,              /* DEFAULT FONT                 */
//          PIMSG=(YES,16),          /* ACCUMULATE DATA SET MESSAGES */
//          DATACK=UNBLOCK,          /* UNBLOCK DATA CHECKS          */
//          TRACE=NO,                /* BUILD INTERNAL TRACE         */
//          SETUP=FORMS              /* SETUP MESSAGE                */
//*  DEFAULT PAGEDEF AND CHARS ARE SPECIFIED  BY JES3                */
//*         PAGEDEF=A868             /* COPY OF A08682               */
//*         CHARS=GT12               /* DEFAULT FONT                 */
//PRT1      ENDCNTL
```

Figure 29 (Part 1 of 3). Sample PSF Startup Procedures for Three Separate Printers

For an SNA-Attached Printer:

```
//SAMPPRO2 PROC
//* THE FOLLOWING STARTUP PROCEDURE DEFINES THE JCL AND PRINTDEV
//* STATEMENTS FOR PRT2, AN SNA-ATTACHED PRINTER.
//*
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K,TIME=1440
//STEPLIB  DD DSN=PSF.LINKLIB,DISP=SHR
//JOBHDR OUTPUT PAGEDEF=V06483,         /* JOB HEADER PAGE           */
//       FORMDEF=A10120,CHARS=GT12      /*   FORMDEF: ALTERNATIVE BIN*/
//JOBTLR OUTPUT PAGEDEF=V06483,         /* JOB TRAILER PAGE          */
//       FORMDEF=A10110,CHARS=GT12      /*   FORMDEF: MAIN BIN       */
//DSHDR  OUTPUT PAGEDEF=V06483,         /* DATA SET SEPARATOR        */
//       FORMDEF=A10110,CHARS=GT12      /*   FORMDEF: MAIN BIN       */
//MSGDS  OUTPUT PAGEDEF=A08682,         /* MESSAGE DATA SET          */
//       FORMDEF=A10110,CHARS=GT15      /*                           */
//*
//FONT01 DD DSN=SYS1.FONTLIB,DISP=SHR   /* SYSTEM FONTS              */
//       DD DSN=INST.FONTLIB,DISP=SHR   /* INSTALLATION USER FONTS   */
//*
//PSEG02 DD DSN=INST.PSEGLIB,DISP=SHR   /* INSTALLATION PAGE SEGMENTS*/
//*
//OLAY01 DD DSN=INST.OVERLIB,DISP=SHR   /* INSTALLATION OVERLAYS     */
//*
//PDEF01 DD DSN=SYS1.PDEFLIB,DISP=SHR   /* SYSTEM PAGE DEFINITIONS   */
//       DD DSN=INST.PDEFLIB,DISP=SHR   /* INSTALLATION PAGE DEFS    */
//*
//FDEF01 DD DSN=SYS1.FDEFLIB,DISP=SHR   /* SYSTEM FORM DEFINITIONS   */
//       DD DSN=INST.FDEFLIB,DISP=SHR   /* INSTALLATION FORM DEFS    */
//*
//PRT2     CNTL
//PRT2     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD          */
//         OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//         PSEGDD= *.PSEG02,         /*SEGMENT  LIBRARY DD           */
//         PDEFDD= *.PDEF01,         /*PAGEDEF  LIBRARY DD           */
//         FDEFDD= *.FDEF01,         /*FORMDEF  LIBRARY DD           */
//         JOBHDR= *.JOBHDR,         /*JOB HEADER SEPARATOR OUTPUT   */
//         JOBTRLR= *.JOBTLR,        /*JOB TRAILER SEPARATOR OUTPUT  */
//         DSHDR= *.DSHDR,           /*DATA SET HEADER SEPARATOR     */
//         FORMDEF=A10110,           /* DEVICE FORMDEF DEFAULT       */
//         PIMSG=(YES,16),           /* ACCUMULATE DATA SET MESSAGES */
//         DATACK=BLOCK,             /* BLOCK DATA CHECKS            */
//         TRACE=NO,                 /* BUILD INTERNAL TRACE         */
//         FAILURE=WCONNECT,         /* VTAM: ATTEMPT RECONNECT      */
//         DISCINTV=0,               /* VTAM: NO TIME OUT            */
//         MGMTMODE=IMMED,           /* VTAM: MAINTAIN SESSION       */
//          APPLID=name,             /* VTAM: APPLICATION PGM NODE   */
//          LUNAME=name              /* VTAM: LOGICAL UNIT NODE      */
//*  DEFAULT PAGEDEF AND CHARS ARE SPECIFIED  BY JES3                */
//*         PAGEDEF=A868             /* COPY OF A08682               */
//*         CHARS=GT12               /* JES3 DEFAULT FONT            */
//PRT2      ENDCNTL
```

Figure 29 (Part 2 of 3). Sample PSF Startup Procedures for Three Separate Printers

For a TCP/IP-attached printer:

```
//SAMPPRO3 PROC
//* THE FOLLOWING STARTUP PROCEDURE DEFINES THE JCL AND PRINTDEV
//* STATEMENTS FOR PRT3, A TCP/IP-ATTACHED PRINTER.
//*
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K,TIME=1440
//STEPLIB  DD DSN=PSF,LINKLIB,DISP=SHR
//JOBHDR   OUTPUT PAGEDEF=V06483,        /* JOB HEADER PAGE           */
//         FORMDEF=A10120,CHARS=GT12     /*   FORMDEF: ALTERNATIVE BIN*/
//JOBTLR   OUTPUT PAGEDEF=V06483,        /* JOB TRAILER PAGE          */
//         FORMDEF=A10110,CHARS=GT12     /*   FORMDEF: MAIN BIN       */
//         FORMDEF=A10110,CHARS=GT12     /*   FORMDEF: MAIN BIN       */
//MSGDS    OUTPUT PAGEDEF=A08682         /* MESSAGE DATA SET          */
//         FORMDEF=A10110,CHARS=GT15     /*                           */
//*
//FONT01 DD DSN=SYS1.FONTLIB,DISP=SHR   /* SYSTEM FONTS              */
//       DD DSN=INST.FONTLIB,DISP=SHR   /* INSTALLATION USER FONTS   */
//*
//PSEG02 DD DSN=INST.PSEGLIB,DISP=SHR   /* INSTALLATION PAGE SEGMENTS*/
//*
//OLAY01 DD DSN=INST.OVERLIB,DISP=SHR   /* INSTALLATION OVERLAYS     */
//*
//PDEF01 DD DSN=SYS1.PDEFLIB,DISP=SHR   /* SYSTEM PAGE DEFINITIONS   */
//       DD DSN=INST.PDEFLIB,DISP=SHR   /* INSTALLATION PAGE DEFS    */
//*
//FDEF01 DD DSN=INST.FDEFLIB,DISP=SHR   /* SYSTEM FORM DEFINITIONS   */
//       DD DSN=INST.FDEFLIB,DISP=SHR   /* INSTALLATION FORM DEFS    */
//*
//PRT3     CNTL
//PRT3     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD          */
//PRT3      OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD          */
//          PSEGDD=*.PSEG02,         /* SEGMENT  LIBRARY DD          */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD          */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD          */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT  */
//          JOBTRLR=*.JOBTLR,        /* DATA SET HEADER SEPARATOR    */
//          DSHDR=*.DSHDR,           /* MESSAGE DATA SET OUTPUT      */
//          FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT       */
//          PIMSG=(YES,16),          /* ACCUMULATE DATA SET MESSAGES */
//          DATACK=BLOCK,            /* BLOCK DATA CHECKS            */
//          TRACE=NO,                /* BUILD INTERNAL TRACE         */
//          FAILURE=WCONNECT,        /* ATTEMPT RECONNECT            */
//          DISCINTV=0,              /* NO TIME OUT                  */
//          MGMTMODE=IMMED,          /* MAINTAIN SESSION             */
//          IPADDR='9.99.12.33'      /* IP ADDRESS OF PRINTER        */
//*  DEFAULT PAGEDEF AND CHARS ARE SPECIFIED BY JES3                */
//*         PAGEDEF=A868             /* COPY OF A08682               */
//*         CHARS=GT12               /* JES3 DEFAULT FONT            */
//PRT3      ENDCNTL
```

Figure 29 (Part 3 of 3). Sample PSF Startup Procedures for Three Separate Printers

JCL for PSF Startup Procedures

This section describes the statements and the parameters shown in Figure 28 on page 76 and Figure 29 on page 78.

nnnnnnnn **PROC**

Specifies the name of the startup procedure. The procedure name must be specified in the JES2 FSS or the JES3 FSSDEF initialization statement.

Note: The STEPLIB DD statement specifies the name of the data set containing the PSF program. If that statement is not used, the standard MVS search order is used to determine which data set contains the PSF program.

EXEC

Specifies the name of the PSF program and establishes the size of the region. For more information on region size, see “Selecting Region Sizes” on page 114. To prevent TIMEOUT abends, you can include TIME=W440.

OUTPUT

Specifies the page definitions, form definitions, and fonts used to format and print different pages in a print job or data set. In the examples, these pages include job and data-set header pages, job-trailer pages, and pages used for printing messages.

For information about how to specify OUTPUT so that the messages are not printed, see “Redirecting the Message Data Set” on page 115.

DD

Identifies the system libraries that contain resources (fonts, page segments, overlays, page definitions, and form definitions) used for printing.

Note: If you are using Overlay Generation Language/370, any 3800 double-byte fonts must be in a library separate from other double-byte fonts.

CNTL

Specifies the beginning of program control statements for the printer. This statement must precede the PRINTDEV statement, and has no parameters.

The label on the CNTL statement must correspond to the label on the ENDCNTL and PRINTDEV statements and to the name specified on the JNAME parameter on the JES3 DEVICE initialization statement. For JES2, the label on the CNTL, ENDCNTL, and PRINTDEV statements must correspond to the printer name on the PRT*nnnn* initialization statement.

PRINTDEV

Specifies printer-initialization parameters. A PRINTDEV statement is entered for each printer FSA belonging to the FSS that names this PSF startup procedure. The label on the PRINTDEV statement must match the label on the JES2 PRT*nnnn* initialization statement, or the JNAME parameter on the JES3 DEVICE initialization statement.

For sample JES2 and JES3 initialization statements, see Figure 25 on page 65 and Figure 26 on page 70. For a description of the PRINTDEV parameters, see Chapter 6, “PRINTDEV Parameters.”

ENDCNTL

Specifies the end of program control statements for the printer.

This statement must follow the PRINTDEV statement and has no parameters.

The label on the ENDCNTL statement must match the label on the CNTL and PRINTDEV statements and the PRT*nnnn* statement (under JES2), or the JNAME parameter (under JES3).

Separating Jobs

Depending on the printer model, users' jobs can be separated by the following:

- Job separator pages
- Forms marking
- Carrier-strip marking
- Copy-marking or offset stacking

Job separation is controlled by means of JES initialization statements or JES operator commands.

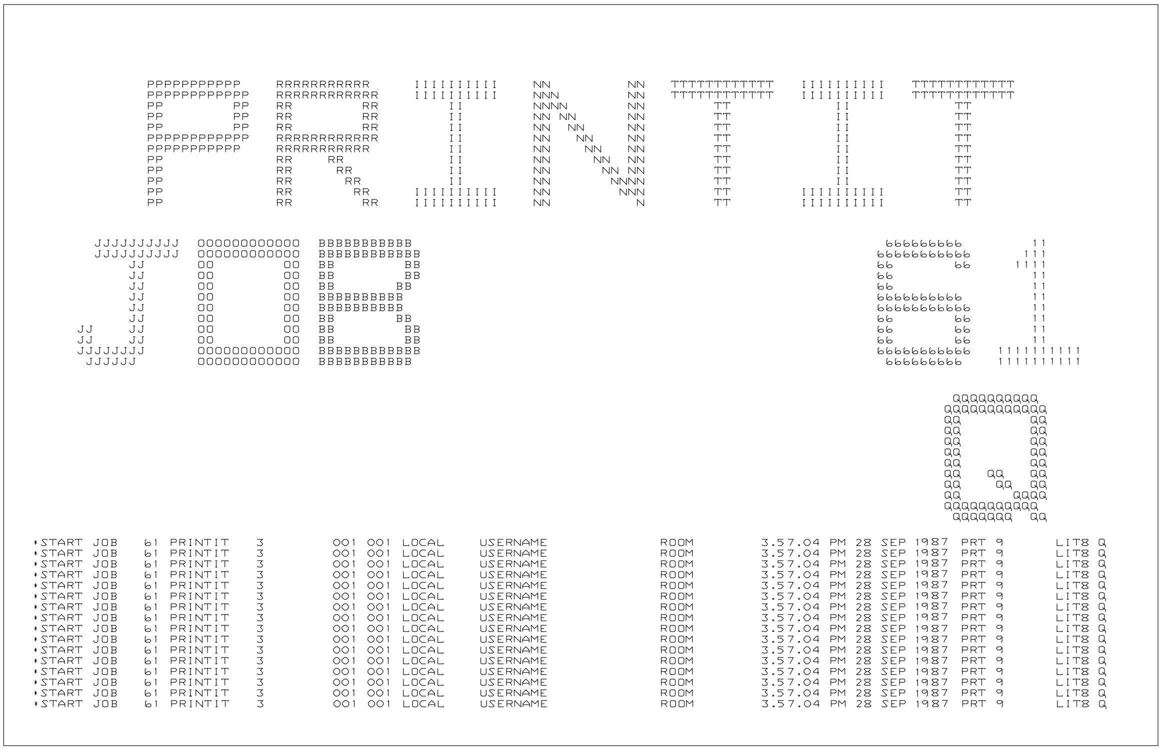
Job Separator Pages

PSF creates and prints job-header and job-trailer separator pages before and after each print job. Whether job separator pages are printed is specified with the following JES initialization statements:

- Under JES2, the SEP parameter determines whether job-header and job-trailer separator pages are printed.
- Under JES3, the HEADER parameter specifies printing of job-header pages, and the BURST parameter specifies job-trailer pages.

For additional information, see “Parameters for the JES2 Initialization Statement” on page 65 and “Parameters for the JES3 Initialization Statement” on page 70.

The content of the header and trailer pages is controlled through PSF installation exits APSUX01 and APSUX02. PSF supplies default versions of these exits. A sample header page produced by the APSUX01 module supplied with PSF is shown in Figure 30.

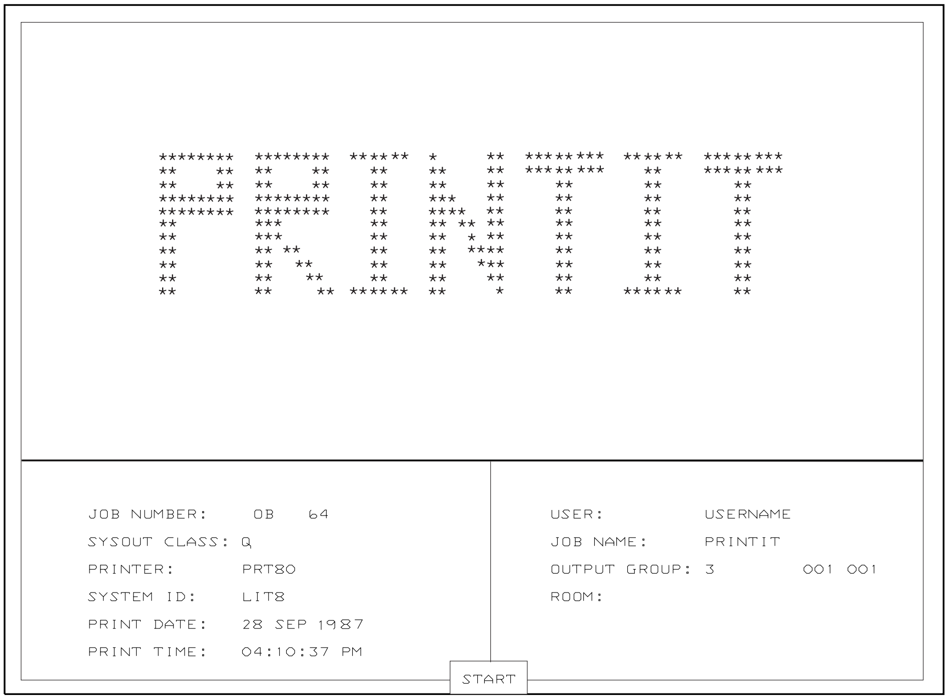


```
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
*START JOB   61 PRINTIT   3       001 001 LOCAL    USERNAME         ROOM      3.57.04 PM 28 SEP 1987 PRT 9      LIT8 Q
```

Figure 30. Example of a Default PSF Job-Header Page (APSUX01)

PSF also provides optional replacements for APSUX01 and APSUX02 that create header and trailer pages containing fewer characters. A sample header page created by the optional separator page exit is shown in Figure 31 on page 83.
| This format should print faster than the default separator pages on SNA-attached
| and TCP/IP-attached printers, because it contains less data to be transmitted to the printer.

For better performance on the 4224 and 4234 printers, use the default separator pages (APSUX01, APSUX02), shown in Figure 30, rather than the optional separator pages (APSUX01P, APSUX02P).



```
********  ********  ******   *     **  ********   ******   ********
**    **  **    **    **     **    **  ********     **     ********
**    **  **    **    **     **    **     **        **        **
********  ********    **     ***   **     **        **        **
********  ********    **     ****  **     **        **        **
**        ***         **     ** ** **     **        **        **
**        ***         **     **  * **     **        **        **
**        ** **       **     **  ****     **        **        **
**        **   **     **     **   ***     **        **        **
**        **    **    **     **    **     **        **        **
**        **     ** ******   **     *     **      ******      **
```

```
JOB NUMBER:     OB   64            USER:           USERNAME
SYSOUT CLASS: Q                    JOB NAME:       PRINTIT
PRINTER:      PRT80                OUTPUT GROUP: 3          001 001
SYSTEM ID:    LIT8                 ROOM:
PRINT DATE:   28 SEP 1987
PRINT TIME:   04:10:37 PM
```

Figure 31. Example of a Fast-Printing Job-Header Page for PSF (APSUX01P). The appearance of your page may differ slightly from the example shown.

In addition, PSF supplies source code for a version of APSUX01 and APSUX02 to create job separators that use guaranteed print labeling (APSUX01S, APSUX02S). For more information about guaranteed print labeling, refer to *Print Services Facility: Security Guide*.

You can also create your own separator page exits. For more information, see “Changing Exit Routines” on page 164 and “Separator-Page Exits: APSUX01, APSUX02, and APSUX03” on page 175.

Additional Distribution Information

The separator page produced by the IBM-supplied exits may contain additional distribution information. This information is obtained from the OUTPUT JCL keywords for the print data set. The data from the following keywords can appear on the separator page:

ADDRESS
BUILDING
DEPT
NAME
ROOM
TITLE

In addition, if the data set is segmented, the segment ID is placed on the separator page. For more information on these keywords and how to specify them, refer to *Print Services Facility/MVS: Application Programming Guide* and the JCL reference publication. The order numbers for these publications are listed in Appendix K, “Related Publications.”

If the following requirements are met, additional distribution information appears on the separator pages:

- The MVS/ESA SP version is 4.1.0 or above.
- The JES2 version is 4.1.0 or above.
- The JES3 version is 4.2.1 or above.
- The IBM-supplied exits are assembled with SYSPARM set to any value except null.

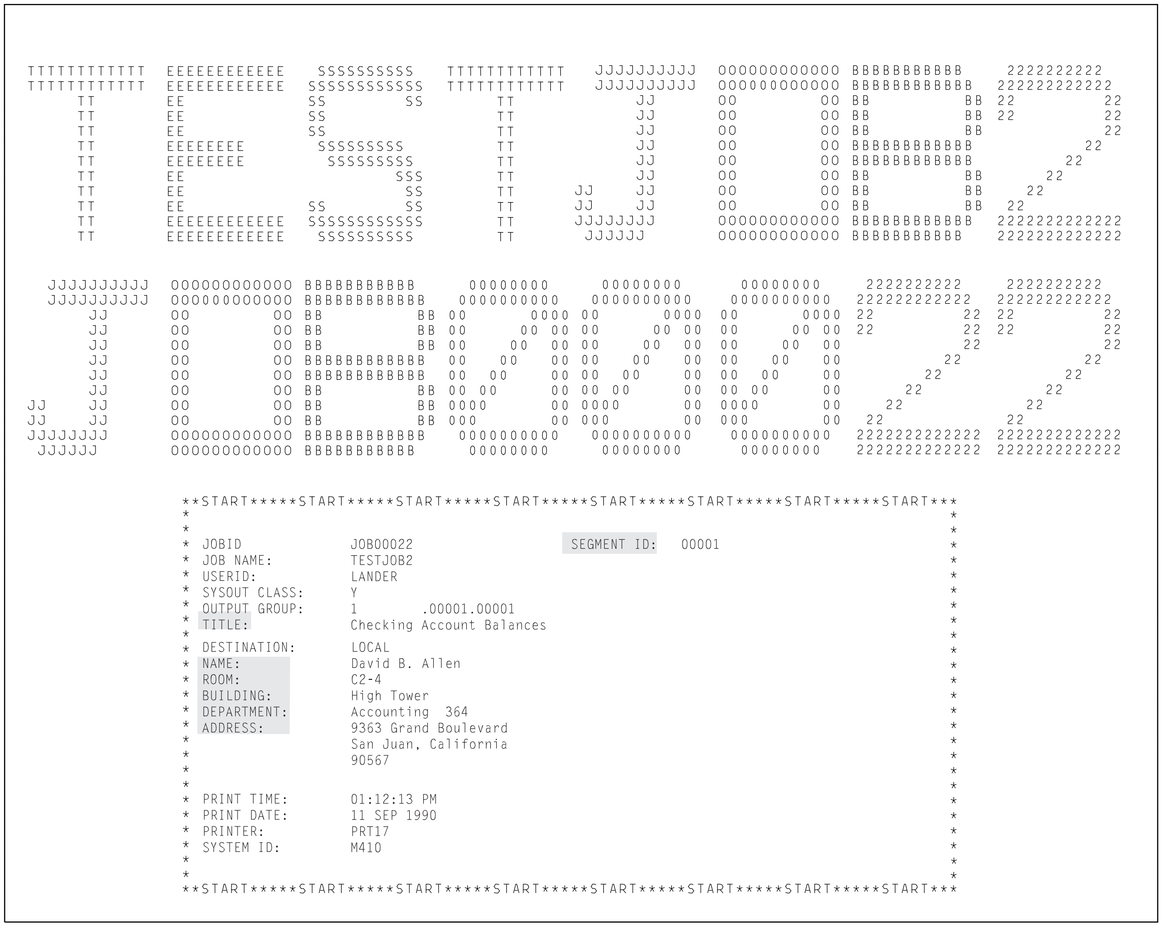
  The SYSPARM value causes the exits to use the format that places the additional distribution information on the separator page. If SYSPARM is omitted, the exits are assembled to produce the old format.

  The default APSUX01 and APSUX02 exits are shipped assembled with SYSPARM specified. The default exits place the additional distribution information on the separator page if the required levels of MVS and JES are installed. If the default APSUX01 and APSUX02 exits are used without modification, no assembly is required to obtain the additional distribution information.

  Here is an example of how to specify SYSPARM:

  ```
  //ASM   EXEC asm_pgm,PARM=(SYSPARM)YES))
  ```

A sample header page containing the additional distribution information produced by the default APSUX01 exit supplied with PSF is shown in Figure 32 on page 85.



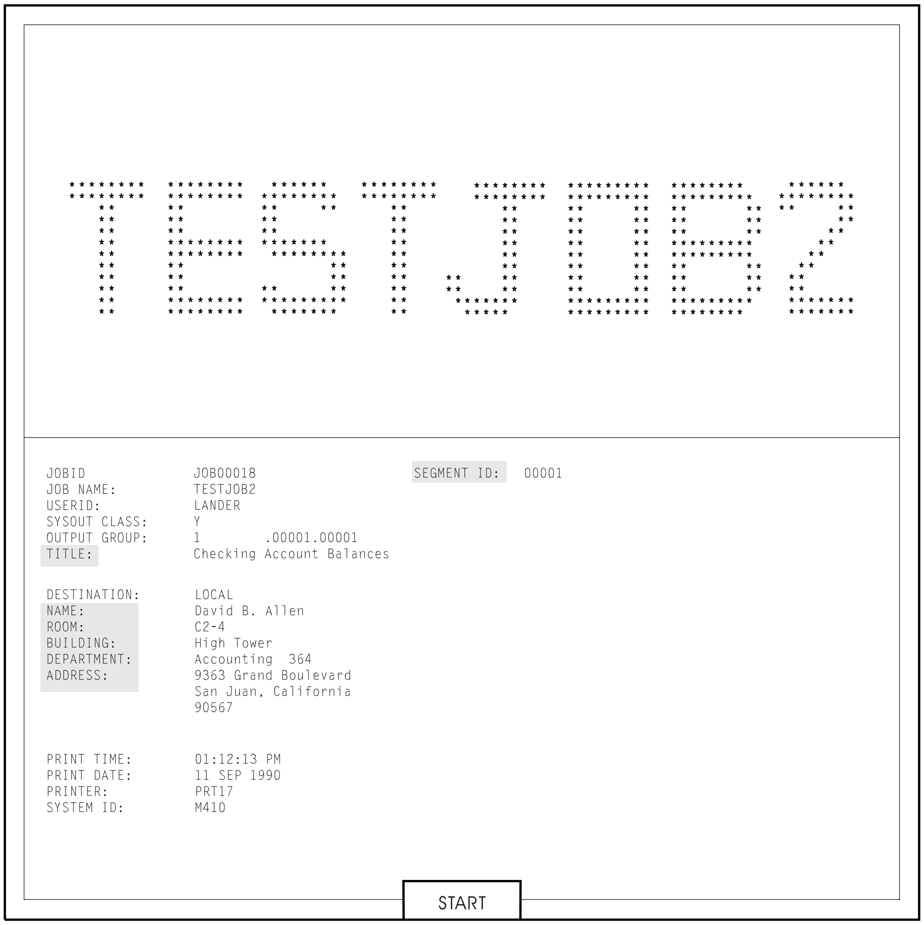
```
TTTTTTTTTTTT  EEEEEEEEEEEE   SSSSSSSSSS   TTTTTTTTTTTT    JJJJJJJJJJ  OOOOOOOOOOOO  BBBBBBBBBBB    2222222222
TTTTTTTTTTTT  EEEEEEEEEEEE  SSSSSSSSSSSS  TTTTTTTTTTTT    JJJJJJJJJJ  OOOOOOOOOOOO  BBBBBBBBBBBB  222222222222
     TT       EE            SS        SS       TT             JJ      OO        OO  BB        BB  22        22
     TT       EE            SS                 TT             JJ      OO        OO  BB        BB  22        22
     TT       EE            SS                 TT             JJ      OO        OO  BB        BB            22
     TT       EEEEEEEE       SSSSSSSSS         TT             JJ      OO        OO  BBBBBBBBBBBB           22
     TT       EEEEEEEE        SSSSSSSSS        TT             JJ      OO        OO  BBBBBBBBBBBB         22
     TT       EE                     SSS       TT             JJ      OO        OO  BB        BB       22
     TT       EE                      SS       TT       JJ    JJ      OO        OO  BB        BB     22
     TT       EE            SS        SS       TT       JJ    JJ      OO        OO  BB        BB   22
     TT       EEEEEEEEEEEE  SSSSSSSSSSSS       TT       JJJJJJJJ      OOOOOOOOOOOO  BBBBBBBBBBBB  222222222222
     TT       EEEEEEEEEEEE   SSSSSSSSSS        TT        JJJJJJ       OOOOOOOOOOOO  BBBBBBBBBBB   222222222222

  JJJJJJJJJJ  OOOOOOOOOOOO  BBBBBBBBBBB     00000000      00000000      00000000     2222222222    2222222222
  JJJJJJJJJJ  OOOOOOOOOOOO  BBBBBBBBBBBB   0000000000    0000000000    0000000000   222222222222  222222222222
      JJ      OO        OO  BB        BB  00      0000  00      0000  00      0000  22        22  22        22
      JJ      OO        OO  BB        BB  00     00 00  00     00 00  00     00 00  22        22  22        22
      JJ      OO        OO  BB        BB  00    00  00  00    00  00  00    00  00            22            22
      JJ      OO        OO  BBBBBBBBBBBB  00   00   00  00   00   00  00   00   00           22            22
      JJ      OO        OO  BBBBBBBBBBBB  00  00    00  00  00    00  00  00    00         22            22
      JJ      OO        OO  BB        BB  00 00     00  00 00     00  00 00     00       22            22
JJ    JJ      OO        OO  BB        BB  0000      00  0000      00  0000      00     22            22
JJ    JJ      OO        OO  BB        BB  000       00  000       00  000       00   22            22
JJJJJJJJ      OOOOOOOOOOOO  BBBBBBBBBBBB   0000000000    0000000000    0000000000   222222222222  222222222222
 JJJJJJ       OOOOOOOOOOOO  BBBBBBBBBBB     00000000      00000000      00000000     222222222222  222222222222

                **START*****START*****START*****START*****START*****START*****START*****START***
                *                                                                              *
                *                                                                              *
                * JOBID          JOB00022               SEGMENT ID:   00001                    *
                * JOB NAME:      TESTJOB2                                                      *
                * USERID:        LANDER                                                        *
                * SYSOUT CLASS:  Y                                                             *
                * OUTPUT GROUP:  1        .00001.00001                                         *
                * TITLE:         Checking Account Balances                                     *
                *                                                                              *
                * DESTINATION:   LOCAL                                                         *
                * NAME:          David B. Allen                                                *
                * ROOM:          C2-4                                                          *
                * BUILDING:      High Tower                                                    *
                * DEPARTMENT:    Accounting  364                                               *
                * ADDRESS:       9363 Grand Boulevard                                          *
                *                San Juan, California                                          *
                *                90567                                                         *
                *                                                                              *
                *                                                                              *
                * PRINT TIME:    01:12:13 PM                                                   *
                * PRINT DATE:    11 SEP 1990                                                   *
                * PRINTER:       PRT17                                                         *
                * SYSTEM ID:     M410                                                          *
                *                                                                              *
                *                                                                              *
                **START*****START*****START*****START*****START*****START*****START*****START***
```

Figure 32. Sample Header Page. Additional distribution information is produced by the default APSUX01 exit supplied with PSF.

A sample header page containing the additional distribution information produced by the optional “fast printing” APSUX01 exit (APSUX01P) is shown in Figure 33 on page 86.


TESTJOB2
START

```
JOBID             JOB00018                SEGMENT ID:   00001
JOB NAME:         TESTJOB2
USERID:           LANDER
SYSOUT CLASS:     Y
OUTPUT GROUP:     1        .00001.00001
TITLE:            Checking Account Balances

DESTINATION:      LOCAL
NAME:             David B. Allen
ROOM:             C2-4
BUILDING:         High Tower
DEPARTMENT:       Accounting  364
ADDRESS:          9363 Grand Boulevard
                  San Juan, California
                  90567


PRINT TIME:       01:12:13 PM
PRINT DATE:       11 SEP 1990
PRINTER:          PRT17
SYSTEM ID:        M410
```

Figure 33. Sample Header Page. Additional distribution information is produced by the optional "fast printing" APSUX01 exit (APSUX01P).

Separator Page Resources

The form definition, page definition, and font used for printing the separator pages are specified on OUTPUT statements in the PSF startup procedure. These output statements are referenced by the JOBHDR and JOBTRLR parameters of the PRINTDEV statement. Examples of separator page OUTPUT statements are shown in "Sample PSF Startup Procedures" on page 75.

In the separator page OUTPUT statement, specify a fixed-space font that prints at 12 or more characters per inch, and specify a page definition that prints in the desired print direction. Use a form definition suitable for your printer. For example, for a cut-sheet printer, you can specify a form definition that selects job-header pages from the alternative paper bin (in which you keep colored paper).

PSF does not use inline resources to process separator pages. PSF obtains all resources used for separator pages from the system library or the security library.

The page origin for the job-header or job-trailer page is the same as the default media origin. For more information on default media origins, refer to *Advanced Function Presentation: Printer Information*. For information on the coordinate system used to define page and media origins, refer to *Intelligent Printer Data Stream Reference*.

Printing Copies of a Separator Page

To print more than 1 copy of a header or trailer separator page, use one of the following methods:

- If the header or trailer uses the default form definition, specify the number of copies in the COPIES parameter on the OUTPUT statement for the header or trailer page. For example, to print 3 copies of your trailer page, specify:

  ```
  //JOBTRLR  OUTPUT PAGEDEF=V06483,CHARS=GT15,COPIES=(,3)
  ```

- If the header or trailer does *not* use the default form definition, specify the number of copies in the form definition used by the header or trailer. Specify the name of the form definition in the FORMDEF parameter on the OUTPUT statement for the header or trailer page. For example, to use form definition F1MYSEP for a trailer page, specify:

  ```
  //JOBTRLR  OUTPUT PAGEDEF=V06483,CHARS=GT15,FORMDEF=MYSEP
  ```

Notes:

1. Include the OUTPUT statement for the header or trailer in the PSF startup procedure. Name the OUTPUT statement in the JOBHDR or JOBTRLR parameter of the PRINTDEV statement in the PSF startup procedure. See “Sample PSF Startup Procedures” on page 75 for sample PSF startup procedures.
2. Do *not* specify both the COPIES and FORMDEF parameters on the OUTPUT statement. If you do, PSF ignores the COPIES parameter and uses the number of copies specified in the form definition.
3. Do *not* use the COPIES=(*n*) form of the COPIES parameter. If you do, PSF ignores the COPIES parameter because this form does not apply to separator pages. Always use the COPIES=(,*n*) form as shown in the example.

For additional information on the COPIES parameter, refer to *Print Services Facility/MVS: Application Programming Guide*.

If forms marking is specified, the 3800 printer automatically prints multiple copies of trailer pages. See “Forms Marking” on page 88.

Distributed Print Function (DPF) Considerations

PSF identifies to DPF the data sets to be printed as a group. DPF keeps a data set group together as one print file on the spool.

JES2 4.1.0 and JES3 4.2.0 and above supply the group boundaries to PSF. Earlier levels of JES request a job header for the first data set of a group, if job separators have been specified.

When you attach DPF to PSF, if the JES level is earlier than JES2 4.1.0 or JES3 4.2.0, you must specify job-separator pages, using the following JES parameters:

- Under JES2, the SEP parameter specifies that job-header and job-trailer pages are to be printed.
- Under JES3, the HEADER=YES parameter specifies that job-header pages are to be printed, and the BURST=YES parameter specifies that job-trailer pages are to be printed.

Note: If you do not want PSF to generate separator pages, delete or modify the separator-page exits. For information on how to do this, see "Programming with Separator-Page Exits" on page 175.

For additional information about JES, see "Parameters for the JES2 Initialization Statement" on page 65 and "Parameters for the JES3 Initialization Statement" on page 70.

PSF and DPF on a JES3 System: When using DPF with JES3, make sure that all data sets grouped into a single DPF print file have the same forms requirements.

To ensure that JES3 schedules data sets with similar forms requirements together, use the SETUP=FORMS parameter on the PRINTDEV statement. When you do this, however, a change of forms will usually cause an intervention message to be sent to the JES3 console. In this environment, the intervention message is neither necessary nor desirable. Because DPF is a spooling device, no forms actually need to be changed when the message is displayed. By using the JES3 initialization deck, you can suppress the intervention message for an individual printer. You can issue a JES3 operator command to change the setting in the deck.

You can produce output that consists of multiple print data sets with different print classes between job separators. If this is how your installation is set up, all of the data sets between the job separators is grouped into one print file for DPF, but only the class of the first data set in the group is displayed by DPF.

Forms Marking

Forms marking produces marks on the edges of the trailer pages, as Figures 34 and 35 show. If separator pages are turned off, the marks appear on the edges of a blank page that PSF has inserted at the end of the job. If there is a job header page but no job trailer page, the marks appear on the edges of the header page.

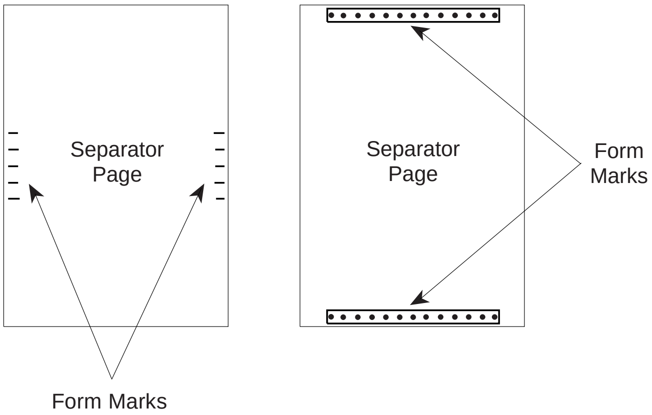


Figure 34. Example of Forms Marking for Cut-Sheet Paper. The actual mark depends on your printer.

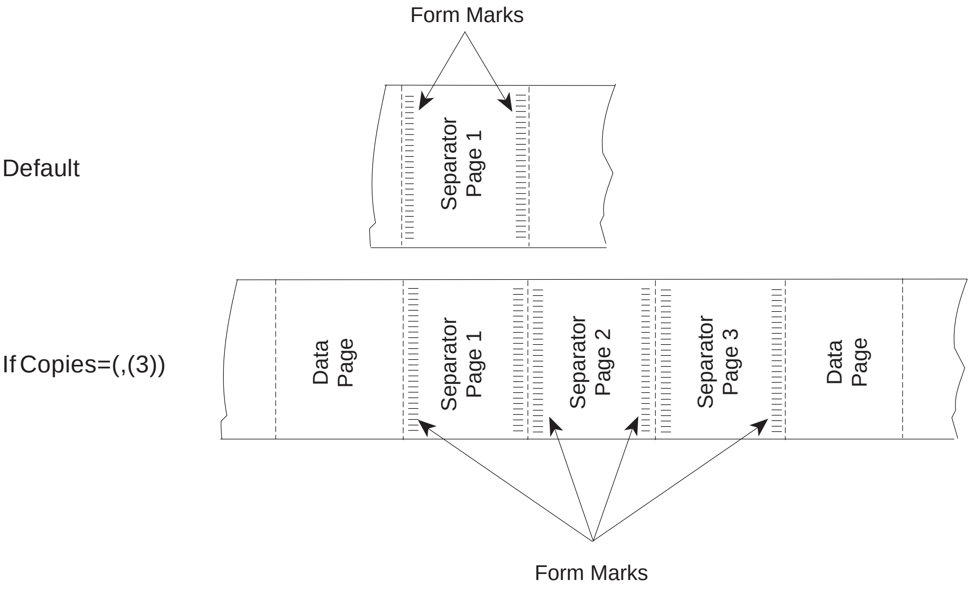


Figure 35. Forms Marking for Continuous Forms Not Printed on a 3800 Printer. Marks are not printed across perforations.

Forms marking is controlled by the following JES initialization options:

- For JES2, select the MARK option on the PRT*nnnn* statement.
- For JES3, select BURST=(YES,M) on the DEVICE statement.

For more information, see “Sample Initialization Statements for JES2” on page 64 and “Sample Initialization Statements for JES3” on page 69.

For a 3800 printer, forms marking extends across the perforation as shown in Figure 36.

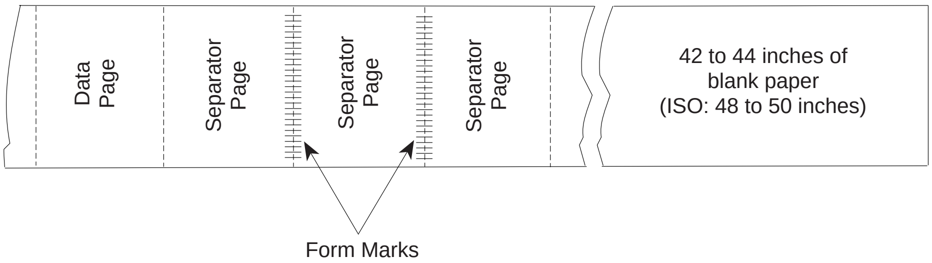


Figure 36. Example of Forms Marking. This example is for 7-inch, 8.5-inch, and 11-inch continuous forms.

If the form length is 7 inches, 8.5 inches, or 11 inches, three copies of the trailer page are printed. For shorter forms, 5 separator pages are printed and marked. Because the marks extend across the perforation, a length of blank paper is issued after the marked trailer pages to prevent the markings from stopping under the fuser roll and transferring to other pages.

Forms Marking of Multiple-Page Job Trailers

You can control the forms marking of trailer pages by using installation exits. You can mark particular pages of a multiple-page job trailer, or you can suppress marking of trailer pages altogether.

The ECAMFTRL flag in the Exit Communication Area (ECA) controls the forms marking of job trailer pages. Forms marking is done only if you set this field to 0 and specify forms marking in the JES initialization options. The default for the ECAMFTRL flag is 0. To suppress forms marking, set the ECAMFTRL flag to 1.

The ECAMFTRL flag can be set in any PSF exit except exit 5. Once set, the flag remains set until reset by an exit. However, PSF uses the flag only when trailer pages are being processed by exit 2 (APSUX02).

When you are processing multiple-page job trailers in APSUX02, and you want only the last job trailer page to be marked, set the ECAMFTRL flag to 1 for all but the last trailer page. When ECAMFTRL is set to 1, PSF does not send the MARK FORM order to the printer during the processing of the job trailer. APSUX02 should set ECAMFTRL to 1 before putting the first record of the first page of the multiple-page job trailer, and should set ECAMFTRL to 0 before putting the first record of the last page of the multiple-page job trailer.

Forms Marking of the Interrupt Message Page

You can control forms marking and offset stacking of the interrupt message page by using PSF installation exits. This support enables you to mark or offset-stack the interrupt message page(s) that PSF inserts into the printed output. For more information, see “Considerations on Forms Marking of an Interrupt Message Page” on page 199.

Carrier-Strip Marking

The 3800 printer also supports carrier-strip marking, diagrammed in Figure 37.

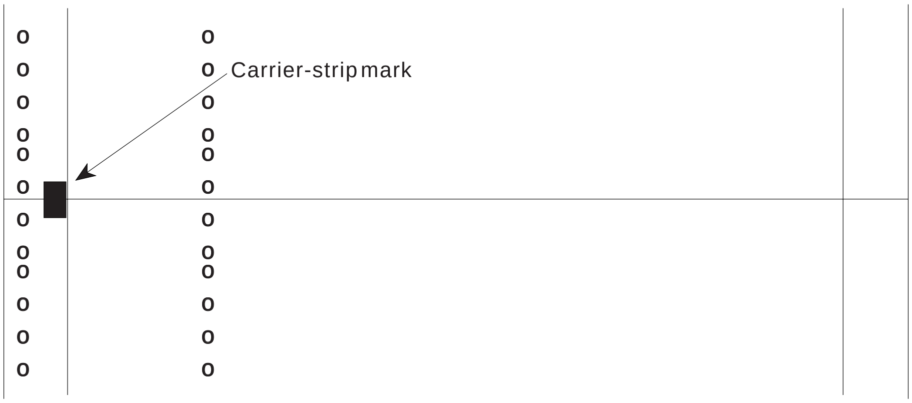


Figure 37. Example of Carrier-Strip Marking on the 3800 Printer

You can control carrier-strip marking by specifying JES initialization options or by using the PSF installation exits.

When carrier-strip marking is controlled by JES initialization options, the carrier-strip mark is printed on the job trailer of each job.

- For JES2, carrier-strip marking occurs on the 3800 printer if forms marking is turned off by use of the MARK=NO option of the PRT*nnnn* statement.
- For JES3, carrier-strip marking is specified in the DEVICE statement with STACKER=(YES,C).

For printers that support the mark form carrier strip (MFCS) command, you can use PSF installation exits to control carrier-strip marking. You can set field ECAMFCS to X'F5' to initiate marking, and to X'00' to disable marking. After ECAMFCS is set to X'F5', carrier-strip marking will be enabled until it is reset to X'00', or until the end of the job. ECAMFCS can be set in any exit except APSUX05. For separator page exits, set the field value before putting the first record. For basic information about exits, see Chapter 17, “Using Installation Exits.”

Note: On the 3800 printer, if you request both carrier-strip marking (MFCS) and forms marking on the trailer page, three copies of the trailer page are printed if the form length is 7 inches, 8.5 inches, or 11 inches, and the carrier-strip marking is printed. The forms marking across the perforation is not printed. For shorter forms, the only difference is that 5 separator pages are printed.

On the 3800 printer, if carrier-strip marking is requested, and the page with the carrier-strip mark stops in the transfer station because no other pages are to be printed (or for any other reason), the carrier-strip mark will not be printed. For more information about carrier-strip marking, refer to *Reference Manual for the IBM 3800 Printing Subsystem Models 3 and 6*.

Copy Marking and Offset Stacking

You can use copy marking and offset stacking to separate data sets within jobs or to separate jobs. Copy marking is used for continuous-forms printers. Offset stacking is used for cut-sheet printers or for continuous forms sent to a burster-trimmer-stacker.

Copy marking is illustrated in Figure 38.

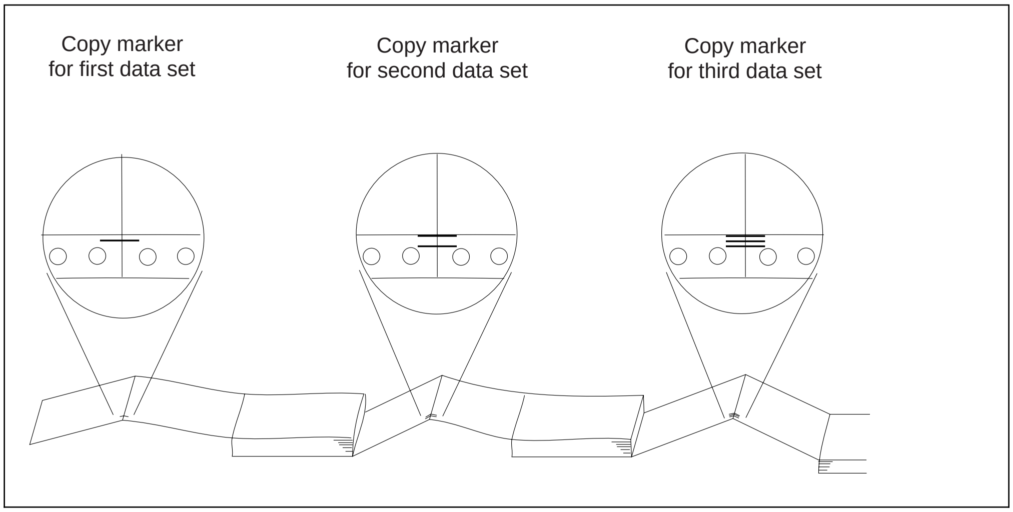


Figure 38. Example of Copy Marking to Simulate Offset Stacking

The printer marks the carrier strip of each page of the data set with 1, 2, or 3 black marks, incrementing the count at the end of each data set or at the end of each job. If the same job is sent to a cut-sheet printer, instead of printing copy marks, the printer offsets each data set or job in the output stacker.

Copy marking and offset stacking are controlled by the following JES initialization parameters:

- For JES2, the COPYMARK parameter of the PRT*nnnn* statement specifies when the copy mark is to be incremented: at the end of each data set or copy of a data set, or at the end of each job.

  For JES2, the COPYMARK parameter of the PRT*nnnn* statement specifies when the offset stacking is to occur: at the end of each data set or copy of a data set, or at the end of each job, or not at all.

  For the syntax of the COPYMARK parameter, see “Parameters for the JES2 Initialization Statement” on page 65.

- For JES3, the COPYMARK parameter of the DEVICE statement specifies when the copy mark increment is to occur: at the end of each data set or copy of a data set, or at the end of each job.

  For JES3, the COPYMARK parameter of the DEVICE statement specifies when the offset stacking is to occur: at the end of each data set or copy of a data set, or at the end of each job, or not at all.

  For the syntax of the COPYMARK parameter, see “Parameters for the JES3 Initialization Statement” on page 70.

The form definition resource can also specify separation by copy group. Then, each time a new copy group is activated, either the pages are offset, or the copy-mark is incremented. For more information on the form definition, refer to *Print Services Facility/MVS: Application Programming Guide*.

Separating Data Sets or Copies of Data Sets

Data sets within a print job, or multiple copies of the same data set within a print job, can be separated by:

- Data-set header pages
- Copy-marking or offset stacking

PSF creates and prints a data-set header page before each data set and before each copy of a data set within a print job. Whether data-set header pages are to be printed is specified by the following JES initialization statements:

- For JES2, the SEPDS parameter of the PRT*nnnn* statement.
- For JES3, the HEADER parameter of the DEVICE statement. This parameter controls both data-set headers and job headers.

For more information, see “Parameters for the JES2 Initialization Statement” on page 65 and “Parameters for the JES3 Initialization Statement” on page 70.

The content of the data-set header page is controlled by PSF installation exit APSUX03. PSF supplies a default version of this exit. You can also code your own version of the exit. For more information, see “Changing Exit Routines” on page 164 and “Separator-Page Exits: APSUX01, APSUX02, and APSUX03” on page 175.

The form definition, page definition, and font used for printing the data-set header pages are specified on an OUTPUT statement in the PSF startup procedure, and are referenced in the DSHDR parameter of the PRINTDEV statement. For information on the OUTPUT statements used with separator pages, see “Job Separator Pages” on page 82.

Copy marking or offset stacking is also used to separate data sets and copies of data sets within a job. For a description of these options, see “Copy Marking and Offset Stacking” on page 91.

System Management Facility Type 6 Records

In deferred-printing mode, PSF creates a system management facility (SMF) type 6 record for each data set that is processed. The information in the type 6 record depends on the device type.

- All printers
  - System ID (processor ID)
  - Logical output-device name
  - User ID
  - Job name
  - Job number
  - Stepname
  - Procedure stepname (procstepname)
  - Data-set name (dsname)
  - Data-definition name (DDNAME)
  - Process mode (PRMODE)
  - Time and date when PSF receives a data set from JES and when the SMF type 6 record is written
  - Data-input error indicator
  - Data-set control indicators (data-set terminated, interrupted, restarted by JES operator)
  - Logical input-record count
  - Copy groups
  - SYSOUT class
  - Physical-page (forms) count
  - Impression (side) count
  - Form name
  - CHARS fonts and UCS fonts
  - Number of fonts, overlays, and page segments loaded
  - Number of page definitions and form definitions loaded in virtual storage
  - Form definition names
  - Page definition names
  - For MVS 4.1.0 with JES2 4.1.0 or with JES3 4.2.1:
    - Segment ID
    - Scheduler work block text units (SWBTU)

      Note: The SWBTU contains the JCL values merged from the DD statement, the OUTPUT statement, and the JES defaults, according to the hierarchy in which the resources are arranged. You can use the SWBTUREQ macro to obtain the JCL parameters specified for the print data set from the SWBTU. The PSF-supplied separator exits (APSUX01, APSUX01P, APSUX01S, APSUX02, APSUX02P, and APSUX025) contain an example of how to invoke the SWBTUREQ macro. For more information about the SWBTUREQ macro, refer to the *Authorized Assembler Services Reference* for your operating system, listed in Appendix K, “Related Publications.” For more information about SWBs and how to use the SWB token to retrieve keywords, refer to the *Using the Functional Subsystem Interface* publication for your operating system, listed in Appendix K, “Related Publications.”

- Printers using continuous forms
  - Paper-footage count
  - Forms-flash name (3800 printer only)
  - Flash count (3800 printer only)
- Printers using cut-sheet paper
  - Cut-sheet indicator
  - Bin-selection indicator

    Note: If the bin selection is changed while a page with multiple copies is being printed, the SMF6PGE count and the SMF6BNCN bin count may be incorrect by the number of copies of pages printed before the bin selection was changed.
  - Page count per bin
  - Duplex indicators
- Printers using the PSF print labeling function
  - For more information, refer to *Print Services Facility: Security Guide*.

PSF supplies an installation exit point that you can use to provide an exit routine (module APSUX05) to modify SMF type 6 records. If you have not provided this routine, PSF creates a standard type 6 record and writes it to the SMF data set.

For information on the APSUX05 module used to modify SMF type 6 records, see “SMF Type 6 Record Exit: APSUX05” on page 188. For information on the structure (the mapping between fields in the record) of SMF type 6 records, refer to the publication on SMF.

The SMF type 6 accounting records written at the host do not reflect any actions taken by the DPF operator, such as changing the copy count, deleting a print file without printing it, or reprinting data because of errors.

Disabled Mechanisms

Some printers can function even if a mechanism, such as a paper source, becomes disabled. When the mechanism first becomes disabled and a job is sent to the printer, PSF abends, as it does for any permanent printer error. However, the operator can choose to restart PSF for a disabled printer, and the mechanism is bypassed as follows:

- If the duplex paper path is disabled, the printer prints in simplex.
- If a paper source is disabled, paper is selected from an alternative source.
- If the offset stacker is disabled, no offset stacking occurs between jobs or copy groups.

For information about printer support for disabled mechanisms, refer to *Advanced Function Presentation: Printer Information*.

You might want to define separate print output classes to assist the operator in determining whether to restart PSF for a disabled printer. One class could be defined for print data sets that are tolerant of disabled mechanisms. This class would be defined for print-job submitters who, for example, will accept simplex instead of duplex output. Another class could be defined for print data sets that are *not* tolerant of changes to the output caused by disabled mechanisms. If separate classes are defined in this way and a mechanism on a printer becomes disabled, the PSF operator can safely start a tolerant class of data sets on that printer and use the printer until the mechanism is enabled.

Chapter 8. Using Direct-Printing Mode

This chapter provides information on how to attach an AFP printer in direct-printing mode, describes what must be done to define PSF as an MVS subsystem, and provides examples of the JCL that must be coded when your application program is run.

PSF supports printers as system output devices for deferred printing under job entry subsystem (JES), and for direct printing under Direct Printer Services Subsystem (DPSS). DPSS is the access means by which the application program sends records directly to an attached printer or directly to PSF, bypassing the JES spool. In direct-printing mode, an application program allocates a host-connected, channel-attached (non-SNA) printer for the program's exclusive use. When you use direct-printing mode, make sure that the printer is not allocated to JES, but is online to MVS.

Notes:

1. Figure 1 on page 8 shows a sample configuration that supports direct-printing mode.
2. SNA-attached and TCP/IP-attached printers cannot use direct-printing mode.
3. Microfilm devices are not supported in direct-printing mode.
4. Direct-printing mode does not provide the following functions, which are provided only in deferred-printing mode mode:
   - User library
   - System-assisted restart
   - PSF repositioning

     Repositioning includes PAPER JAM recovery, PRINT ADJUST commands entered at the printer, and recovery for other intervention-required conditions at the printer.
   - Data-set checkpointing
   - Multiple data set processing
   - JES operator commands to control the printer
   - Job header and trailer pages
   - Data set header pages
   - Carrier-strip marking
   - Mandatory page labeling[4]
   - System management facilities (SMF) type-6 processing
   - Notification of job completion
   - Restartable abends
   - PSF installation exits 1–7

[4] For more information about page labeling, refer to *Print Services Facility: Security Guide.*

- Redirecting the message data set

5. If PSF is attached to a printer that does not support the selective deletion of page segments and overlays, DPSS will stop processing data sets if jobs require more than 127 page segments or more than 127 overlays. If you have IBM printer microcode that supports extended overlays and page segments, the maximum number of overlays is extended to 32 511.

6. Information about the installation exit for direct-printing mode is given in "Direct-Print Exit: APSUX16" on page 214.

7. For forms marking, see the APSUX16 begin-data-set call (BDSC), described in "Direct-Print Exit: APSUX16" on page 214.

8. For additional information, refer to the *Initialization and Tuning* publication for your operating system, listed in Appendix K, "Related Publications."

How to Use Direct-Printing Mode

The procedure for using PSF in direct-printing mode is as follows:

1. Define DPSS to MVS, using the procedures described in "Defining the DPSS" on page 99.

2. Code JCL to invoke direct-printing mode, using "Application Program JCL for Direct-Printing Mode" on page 100 as a sample.

3. Allocate the printer by either:

   - Coding a DD statement in the JCL that is used to invoke direct-printing mode
   - Coding a dynamic allocation invocation in the application program

4. In the application program, code an OPEN macro instruction for a data control block (DCB) that contains the DDNAME to be used to specify the printer in the JCL.

5. If you need more than one copy of a data set, follow the procedures in "Requesting Multiple Copies" on page 102.

6. Code a CLOSE macro instruction for the DCB of the DDNAME of direct-printing mode. PSF then detaches the FSA for the printer that is using direct-printing mode.

7. If the directly attached device is to be managed by PSF, DPSS Open establishes DPSS/PSF as the access method that the application program is to use. When the application closes the print data set, DPSS detaches the PSF subtask and returns to Data Management Close. Other job tasks that intervene through the use of exclusive ENQ/DEQ may result in system deadlock. You can prevent system deadlock by coding application level ENQ/DEQ around all system macros that may require EHQ on TIOT. Some examples of system macros that may require ENQ/DEQ are Open, Close, Dynalloc, and EOV. To avoid unexpected results or abend conditions, review the *Authorized Assembler Services Reference* for your operating system before coding the macro in your application.

Defining the DPSS

DPSS maintains communication between the application program and PSF. DPSS, in turn, uses the functional subsystem interface (FSI) to communicate with PSF and to deliver data to the printer.

To use direct-printing mode, you must first define DPSS as an MVS subsystem by modifying the PARMLIB member IEFSSN00 or by defining a new ISFSSN*xx* member. IEFSSN00 (or the newly created member) must contain the following statement:

```
DPSS,APSADPSS,n
```

where

- DPSS is the subsystem name.
- APSADPSS is the program name.
- *n* is the maximum number of FSAs (printers) to be used per address space.

PSF supports a maximum of 64 FSAs. The actual number of FSAs depends on several factors, such as the number and size of resources used in the print jobs and the amount of virtual storage available

Figure 39 shows a job stream that can create a new IEFSSN*xx* member of SYS1.PARMLIB. Note that if you create a new IEFSSN*xx* member, you must update the IEASYS*xx* member to point to IEFSSN*xx*, and an IPL is required.

```
//IEFSSN JOB ...,MSGLEVEL=1,MSGCLASS=A,CLASS=A
//STEP1 EXEC PGM=IEBUPDTE,PARM=NEW,REGION=256K
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=SYS1.PARMLIB,DISP=SHR
//SYSUT2 DD DSN=SYS1.PARMLIB,DISP=SHR
//SYSIN  DD  *
./ ADD NAME=IEFSSNBB,LIST=ALL
DPSS,APSADPSS,1
./ ENDUP
/*
```

Figure 39. Job Stream for Creating an IEFSSNBB Member of SYS1.PARMLIB

For MVS/ESA 5.2.0 and later, consider the following:

- If the old definition format is used for IEFSSNxx, the IEFSSNxx member must contain the old definition (DPSS,APSADPSS,*n*). If the new definition format is used, the IEFSSNxx member must contain the following:

  ```
  SUBSYS SYSNAME(DPSS) INITRTN(APSADPSS) INITPARM(n)
  ```

- You can dynamically add the subsystem after the system IPL by using the following operator command:

  ```
  SETSSI ADD,SUBNAME=DPSS,INITRTN=APSADPSS,INITPARM=n
  ```

- PSF/MVS DPSS does not support any other dynamic subsystem functions provided with MVS/ESA 5.2.0.

Application Program JCL for Direct-Printing Mode

Figure 40 shows sample JCL statements that enable direct-printing mode. JCL in this format can be incorporated into the application program job stream to start direct-printing mode. Be sure you have allocated the space required in the system library for the libraries you refer to in your JCL.

```
  //DIRECT   JOB...
⋮
  //STEP01   EXEC PGM=program,REGION=1750K
⋮
  //MSGDS    OUTPUT PAGEDEF=pdefname,FORMDEF=fdefname
  //FONT01   DD     DSN=SYS1.FONTLIB,DISP=SHR
  //PSEG01   DD     DSN=SYS1.PSEGLIB,DISP=SHR
  //OLAY01   DD     DSN=SYS1.OVERLIB,DISP=SHR
  //PDEF01   DD     DSN=INST.PDEFLIB,DISP=SHR
  //         DD     DSN=SYS1.PDEFLIB,DISP=SHR
  //FDEF01   DD     DSN=INST.FDEFLIB,DISP=SHR
  //         DD     DSN=SYS1.FDEFLIB,DISP=SHR
  //DA1      CNTL
  //DA1      PRINTDEV TRACE=YES,
  //             FONTDD=*.FONT01,
  //             OVLYDD=*.OLAY01,
  //             PSEGDD=*.PSEG01,
  //             PDEFDD=*.PDEF01,
  //             FDEFDD=*.FDEF01,
  //             MESSAGE=*.MSGDS,
  //             SETUP=FORMS,
  //             BUFNO=5,
  //             PAGEDEF=pdefname,
  //             FORMDEF=fdefname,
  //             CHARS=(fontname1,
  //             fontname2,
  //             fontname3,
  //             fontname4),
  //             PIMSG=(YES,16),
  //             DATACK=BLOCK
  //DA1      ENDCNTL
  //name    DD     UNIT=180,CNTL=*.DA1,...
⋮
```

Figure 40. Sample JCL to Enable Direct Printing

Following are descriptions of the statements and parameters shown in Figure 40 on page 100:

EXEC
Specifies the name of the direct print application program and establishes the region size and the PSF PARM parameters used for trace specifications. For more information on the PSF PARM parameters and tracing, refer to apsd000-> PSF/MVS: Diagnosis Guide and Reference LH40-0205 For more information on the REGION parameter, see “Selecting Region Sizes” on page 114.

MSGDS OUTPUT
Specifies the page definition and the form definition used to format and print messages.

DD
Identifies the system libraries that contain resources (fonts, page segments, overlays, page definitions, and form definitions) used for printing. For more information about libraries, refer to *Print Services Facility/MVS: Program Directory*.

CNTL
Precedes the PRINTDEV statement; ENDCNTL must follow the PRINTDEV statement. The CNTL and ENDCNTL statements have no parameters. The label on the CNTL statement must match the label on the corresponding ENDCNTL and PRINTDEV statements.

PRINTDEV
Specifies initialization parameters for a printer operating in direct-printing mode. Within the JCL for a single direct-printing job, you can code CNTL, PRINTDEV, and ENDCNTL statements for more than one printer. This parameter is required.

For a description of all the PRINTDEV parameters, see Chapter 6, “PRINTDEV Parameters.”

ENDCNTL
Specifies the end of a PRINTDEV statement.

name DD
Defines the printer to be used and refers to the CNTL statement associated with this DD statement. The *name* specifies the DDNAME used by the application program that was specified in the EXEC statement (EXEC PGM=*program*). The UNIT parameter specifies the device address for the printer in direct-printing mode. The CNTL and ENDCNTL sequence that includes the PRINTDEV statement must be within the current step containing the DD statement. This DD statement can also be allocated by using dynamic allocation instead of JCL. Refer to the *Authorized Assembler Services Guide* for your operating system for information about how to use dynamic allocation. With dynamic allocation, the direct print application program can allocate the printer while the program is running, and thus can control how and when the printer is allocated. For instance, the application program can determine such things as:

- Which JCL CNTL statement to reference for the allocation
- Which printer to allocate

- The particular point in the application's processing at which the allocation is to occur

Dynamic allocation provides greater flexibility than JCL DD allocation provides, as Table 3 shows.

Table 3. JCL DD Allocation vs. Dynamic Allocation

JCL DD allocation	Dynamic allocation
Must be defined before the job is submitted.	Can be defined while the application program is running.
Can be used only by authorized programs.	Can be used by both authorized and unauthorized programs.
Must be done during step allocation, just before the system passes control to the direct print allocation program.	Can be done at any point while the application program is running.

When using dynamic allocation, use the DALCNTL text unit key to refer to a CNTL statement in the JCL for the step.

Requesting Multiple Copies

Under JES, PSF retransmits a data set to provide more than one copy. In direct-printing mode, however, your application must retransmit the data set for each groupvalue if either of the following applies:

- More than one copy is required. For example, if you specify COPIES=14 in direct-printing mode, the number of copies defaults to 1. To get 14 copies, your application must retransmit the data set 14 times to PSF.
- More than one groupvalue is specified in a COPIES JCL parameter. For example, if you specify COPIES=(,(1,3,2)) in a direct-printing mode, the total number of requested copies (here 6) is ignored. The first groupvalue (1) determines the number of copies. If the first groupvalue is 3, three copies of each page are printed consecutively.

For information about using the COPIES parameter under JES, refer to *PSF/MVS: Application Programming Guide*.

3800 Considerations

For the 3800 printer, you can use the SETPRT macro instruction to initially set or dynamically change the printer control information. For DPSS, the following parameters can be changed by use of the SETPRT macro:

- BURST
- CHARS
- COPIES
- FCB
- FLASH
- UCS

For additional information about how to use the SETPRT macro, refer to *DFSMS/MVS Macro Instructions for Data Sets*.

Chapter 9. Using Cut-Sheet Emulation Mode

Cut-Sheet Emulation (CSE) is the ability of a continuous-forms printer to provide output similar to output from a cut-sheet printer.

Some continuous-forms printers[5] provide the option of printing in CSE mode. If the operator selects this option, the printer logically divides a continuous-forms sheet in half, parallel to the carrier strips. Then PSF/MVS 2.2.0, with APAR OW19103 applied, automatically prints jobs 2 up, treating each half of the sheet as if it were a single sheet of paper. If the printer is loaded with forms measuring 17 by 11 inches (not counting the carrier strips), PSF treats each 8.5- by 11-inch half-sheet the same as an 8.5- by 11-inch sheet on a cut-sheet printer. For a simplex job, page 1 is printed on the first half of the sheet, and page 2 on the second half. For a duplex job, page 1 is printed on the front of the first half, page 2 on the back of page 1, page 3 on the front of the second half, and page 4 on the back of page 3.[6]

The output, once it has been slit and collated by post-processing equipment, is identical with that printed on 8.5-inch by 11-inch cut-sheet forms.

How Do I Get Cut-Sheet Emulation Working?

To use cut-sheet emulation mode, the operator must select the CSE function on the printer. When the CSE function is selected, PSF automatically ensures that all data sets are printed 2 up unless N_UP is specified in the active copy group of the form definition. The N_UP specification in the form definition overrides CSE because the purpose of N_UP is to give the job submitter control over how the printer places multiple logical pages on a sheet.

Do I Have Any Control over Using Cut-Sheet Emulation?

For all copy groups in the form definition that do not specify N_UP, PSF defaults to CSE when that mode is enabled at the printer. By using installation exit APSUX07, however, you can request that PSF check page sizes to determine whether CSE should be used. APSUX07 gives you two options for checking page size: you can request that PSF check only the first printed page for which a new copy group is used, or you can request that PSF check each front side page. When either of these options is specified, PSF checks the Page Descriptor (PGD) structured field, either in the page definition or in the MO:DCA page, to see whether the page is small enough to fit 2 up on the sheet. If the page is not small enough, CSE is not used; the page is printed in normal 1_UP.

Note: Both APSUX07 options are used only for non-N_UP copy groups.

[5] For the printers and microcode level to support CSE, consult the Customer Engineer for your hardware.

[6] Some printers may offer options enabling the operator to control whether the first half of the sheet is the left or right side, and whether the front of the sheet is the top or bottom side of the physical form.

Will Cut-Sheet Emulation Work with 3800 Line-Mode Conversion?

If you are using the 3800 Line-Mode Conversion function and you have enabled CSE on the printer, PSF builds a page definition that takes advantage of CSE. However, PSF cannot determine whether your 3800 job is designed for wide or narrow paper. Installation exit APSUX04 has an option you can use to let PSF know which paper to emulate. For more information, see “Logical-Record Processing Exit: APSUX04” on page 183.

Things to Consider When Using Cut-Sheet Emulation

- In CSE mode, PSF treats each half of a sheet of paper as a single physical sheet. An operation that causes a new copy group to be invoked, and therefore would ordinarily would cause a sheet eject, now causes only a half-sheet eject. There are numerous such operations:

  – Change in medium overlay
  – Change to a different set of suppressions
  – Change in the number of copies of a page to be printed
  – Transition from simplex to duplex, or from offset to no offset

  Thus when two consecutive data sets are printed in CSE mode, the second starts on the next available half-sheet, without forcing a sheet eject. A single physical sheet of paper could contain the job trailer for job 1 and the job header for job 2. If you need to prevent this, you can force the header page to eject to a new sheet by assigning it a form definition that specifies N_UP (N_UP 1 will work).

- Form definition parameters **PRESENT** and **DIRECTION**, which PSF ordinarily ignores for cut-sheet printers, are honored in CSE mode. This enables CSE to accommodate a wide range of jobs in addition to those formatted specifically for cut-sheet paper, but it has an unfortunate side effect. Customers have sometimes inadvertently coded FORMDEF options other than the default PORTRAIT/ACROSS. Ordinarily, these options have no effect; but in CSE mode they may suddenly take effect, with unexpected and unwanted results.

- When PSF is running in CSE mode, the SMF 6 records reflect the actual number of physical sheets of paper moved through the printer for the SMF6FEET and SMF6PGE fields. SMF6IMPS reflects the number of sides of sheets.

- If your application relies on PSF to place pages on the same half-sheet after a reposition, PSF tries to aid you by forcing the number of interrupt message pages to be an even number. The separator pages, however, still require your attention. To handle them, consider one of the following options:

  – You can request an even number of copies, to force the user data to a new sheet. If the print job data contains an odd number of pages, however, this option may still cause the first reprinted page of the data set to be printed on a different half-sheet from the one the original page was printed on.

  – You can turn off separator pages, and mark forms instead. But if the print job contains an odd number of pages, this option too can cause trouble.

  – You can use N_UP 1 to force the separator pages to be on separate sheets of paper from the user's data.

- If your printer has an attached post-processor that can do left/right offset stacking, the offset stack (jog) command sent by the printer to the post-processor will indicate which half-sheet is to be jogged. The printer determines which half-sheet is to be jogged by simply recognizing which sheet-half was being processed at the time when a jog was requested. If your post-processor cannot handle left/right offset stacking, the printer will combine the jog commands so that if either half-sheet is to be jogged, the entire sheet is jogged.

Chapter 10. Enabling PSF 3800 Line-Mode Conversion

This chapter explains how you can use the PSF/MVS 3800 Line-Mode Conversion function to simplify your migration from a 3800 to a 3835 or 3900 AFP printing environment.

All 3800 printers can operate in "line mode" as non-AFP printers. In line mode, however, they do not use PSF or AFP resources such as page definitions and form definitions; rather, they use 3800 line-mode resources such as FCBs and GRAPHMODs. Without the PSF 3800 Line-Mode Conversion function, migrating to the AFP print environment could make it necessary for many of the resources for a 3800 Line-Mode job to be converted manually to AFP resources. The PSF 3800 Line-Mode Conversion function can reduce the number of required changes to resources, JCL, and application line data, and enable you to use a limited set of standard AFP functions that are not available on the 3800 Line-Mode printer.

This chapter explains what 3800 Line-Mode Conversion is, how it works, how to enable it, and some limitations or restrictions that you may encounter in your particular environment.

What is IBM 3800 Line-Mode Conversion?

The PSF conversion function emulates the operation of a JES-driven IBM 3800 Line-Mode printer. It processes JCL, resources, and line data in much the same way as a 3800 Line-Mode printer, and it detects many common error conditions. In other words, it enables you to print your old JES 3800 jobs with a 3835 or 3900 printer.

In one special case, 3800 Line-Mode Conversion even supports nonstandard OEM FCBs that were designed to print from perforation to perforation on non-IBM printers that are similar to the 3800 printer. PSF 3800 Line-Mode Conversion is *not* supported for a 3800 printer in AFP mode.

Line-mode conversion also makes available some standard AFP functions that are not offered in the 3800 Line-Mode printing environment:

- You can use one electronic overlay, instead of forms flash.
- You can mix AFP and 3800 Line-Mode jobs on the same printer.
- You can set the top and left margins to use the entire printable area, even for old 3800 Line-Mode jobs.

How Does 3800 Line-Mode Conversion Work?

The 3800 Line-Mode Conversion function enables PSF to dynamically build and use inline page definition and form definition resources for 3800 line-mode jobs. The resources are built from information in the job's FCB and JCL for the **LINECT**, **COPIES**, and **FLASH** parameters.

To build the inline page definition, PSF concatenates the member named FCB3 to the FCB value in the JCL, and accesses that member in SYS1.IMAGELIB. If no FCB is specified for the print job, PSF is unable to continue, and stops processing. The expected paper size is calculated from the FCB; it must match the size of the paper loaded in the printer. Page height, as specified in the page definition, includes the full printable page area, to allow room for character ascenders and descenders that a 3800 printer executing in line mode might truncate. The default margin at the top and bottom of the page is set to 1/2 inch (as on the 3800 printer), but can be changed to allow use of the entire printable area. If **LINECT** is specified in the JCL, the inline page definition contains either the number of lines specified by **LINECT** or the number specified in the FCB, whichever is smaller.

Information for the inline form definition is derived from several sources. To set the printer origin to match the origin on the 3800 printer, PSF uses information on the dimensions of the paper currently loaded in the printer. If a forms flash is used for the job, the software also extracts the **FLASH** name from the JCL, and indicates an electronic overlay of the same name. If PSF does not find a user-created electronic overlay with the same name (prefixed *O1*) in the PSF resource library, PSF issues an error message. Uncollated copies specified by the **COPIES** parameter in the JCL are used in conjunction with **FLASH** to determine how many copy subgroups should be specified in the inline form definition.

PSF also uses other information in the JCL, though that information is not integrated into the inline resources. Thus if multiple transmissions, or collated copies, are specified in the **COPIES** parameter, PSF produces multiple collated copies. Further, if 3800 GRAPHMODs and character sets are specified by the **CHARS** parameter, they are accessed as AFP fonts. If you use this method to specify fonts, be aware that unless like-named AFP fonts exist in the PSF/MVS font library before PSF is started, an error results.

How Can PSF 3800 Line-Mode Conversion Be Enabled?

The conversion function is activated through PSF/MVS Exit 4 (APSUX04). Exit 4 is the PSF/MVS record processing exit, which normally gains control each time PSF accesses a print record from the JES spool. To enable the conversion function, set flag XTP4CONV on for the first record of the transmission. For any other records of the transmission, PSF ignores XTP4CONV, although no error message is issued. As a default, PSF sets this flag to OFF before calling APSUX04 for the first record of the transmission.

If you use Exit 04 for any other modification of the records, they are processed normally, as though the 3800 Line-Mode Conversion function did not exist.

Note: For more information on PSF exit processing, see Chapter 17, “Using Installation Exits.”

You can set XTP4NCAL flag ON in Exit 4 to indicate that this exit should not be called again for the rest of the current transmission. Because the JCL is analyzed and inline resources are built only when the first record is processed, you will experience better overall performance if you set XTP4NCAL on. However, if you require Exit 4 functions on any record later in the current transmission, you should wait until you no longer want PSF to call Exit 4 for the current transmission before you set XTP4NCAL on.

When you program the logic for Exit 4, you should identify the 3800 jobs you want converted. Because the exit has access to all the JCL parameters for the job, you may choose to identify jobs by any of these parameters—for example, Job Name or

Job Class. To intermix AFP jobs and 3800 line-mode jobs, Exit 4 must identify each job as one kind or the other, and set the XTP4CONV flag off for AFP jobs and on for 3800 line-mode jobs.

You can also set up a table in Exit 4 that enables the user, by specifying any of the parameters available to the exit, to select special print offsets, XTP4XOFF and XTP4YOFF, for different 3800 line-mode jobs. This enables the conversion function to emulate the print offsets that a 3800 operator might enter in order to match print to special forms. Exit 4 can set the horizontal offset value XTP7XOFF and the vertical offset value XTP4YOFF. Using these fields, you can make adjustments that correspond to the adjustments allowed on the 3800 line-mode printer. The adjustments are based on 240 pels per inch; an adjustment of 2 inches is coded as 480. The defaults are set for a vertical adjustment of 1/2 inch and a horizontal adjustment of 0 inches. These bits are valid only when the XTP4CONV flag is on.

The conversion function provides line-mode processing only for jobs selected by Exit 4. However, not all jobs selected by Exit 4 are converted. Any job that has a PAGEDEF or FORMDEF parameter coded in its JCL is ignored for line-mode processing. PSF assumes that the job already has been designed for AFP printing, and processes it as a normal AFP job.

If you are using Cut-Sheet Emulation mode, you can set the XTP4WINA flag on in Exit 4 to indicate to PSF whether your 3800 line-mode job was designed for wide or narrow paper. If this flag is set OFF, PSF generates portrait pages to emulate narrow paper. If this flag is set ON, PSF generates landscape pages to emulate wide paper. To enable landscape pages, you should set flag XTP4WINA ON for the first record of the transmission. For any other records of the transmission, XTP4WINA is ignored, although no error message is issued. As a default, PSF sets this flag to OFF before APSUX04 is called for the first record of the transmission. The flag XTP4WINA is valid only when XTP4CONV is set ON and the printer is in Cut-Sheet Emulation mode.

Considerations and Limitations of the 3800 Line-Mode Conversion Function

Jobs with the following requirements cannot be printed in PSF/MVS until they have been manually converted for AFP printing:

- COPYMODs

  PSF cannot process jobs that use COPYMODs. If **MODIFY** is specified in the JCL for a job, and that job is identified in Exit 4 for 3800 Line-Mode Conversion processing, PSF issues an error message and requests that the system hold the job. If the same job had not been identified in Exit 4, it would print as a normal AFP job; the MODIFY parameter would simply be ignored. Jobs for which the MODIFY parameter is specified must be converted to AFP jobs manually.

- Customized fonts

  PSF does not convert 3800 character sets and GRAPHMODs. Like-named fonts must be available as AFP bounded-box fonts in the PSF resource library. PSF/MVS comes with AFP versions of all the character sets supplied with the 3800 printer. For any nonstandard character sets used in line-mode print jobs, equivalent AFP fonts must be created or substituted.

- Line-merge

  Jobs that use print-no-space to merge print records may not print the same under PSF 3800 Line-Mode Conversion as they do in 3800 line mode. Line merge is often used on the 3800 printer to mix different fonts on the same print line. In line mode, the 3800 printer merges character by character, regardless of the size of each character. In AFP mode, lines are merged pel by pel; a large character occupies more space than a smaller one. In that mode, therefore, merging characters of different sizes may lead to overprinting or different character spacing.

  Such jobs can be identified by either or both of the following parameters in their JCL:

  – **OPTCD=J**

    indicates that table reference characters (TRCs) are used to select different fonts for different records

  – **CHARS***(xxxx,yyyy,....)*

    Indicates multiple character sets

  If all the character sets listed are fixed-pitch and all of them have the same pitch (such as GT10 AND GB10), you do not have a line merge problem. However, if any of the character sets are mixed-pitch (such as the tri-pitch BITR), or if any two have different pitches (such as GT10 AND GT12), you do have a potential line-merge problem.

  If you do not know whether the application is merging lines of different character sizes, you need to examine the application in detail. If you discover line merge of different-sized characters, the application must be rewritten before it can be run on an AFP printer.

- Post-processing rotation

  You should also examine jobs that use a special post-processing routine and character set to create the 3800 version of rotated print. Although you may be able to convert the rotated character set for the AFP printer, removing the rotate post-processing step and using standard AFP fonts and AFP capabilities to rotate the job is easier.

The following restrictions apply to 3800 Line-Mode Conversion:

- 240 pel

  You can use the 3800 Line-Mode Conversion only to print 3800 line mode jobs on 240-pel printers.

- One input bin

  The 3800 Line-Mode Conversion assumes that the printer has only one input bin and will use paper only from the first input bin identified by the printer.

The following functions can be used with 3800 Line-Mode Conversion, subject to some limitations:

- Printing on cut-sheet AFP printers

  You can print 3800 line-mode jobs on cut-sheet AFP printers (such as the 3825 and 3829 printers) only if the print jobs will fit on the paper that these cut-sheet printers handle. Although 3800 jobs that printed on narrow paper should easily fit on cut-sheet paper, 3800 jobs designed for wide paper will probably not fit.

PSF does *not* rotate 3800 jobs on cut-sheet printers to try to make them fit the paper.

- Perf-to-perf printing when migrating from non-IBM 3800-like printers

  Support is provided by APAR OW08438 to allow migration from non-IBM printers that are similar in operation to the 3800 printer, but have the capability to print from perforation to perforation (no required half-inch margins on the top and bottom of the page). If the FCB contains a special `X'0B'` byte in the first byte of the line space/channel code descriptors, it is a nonstandard FCB (not created through the IEBIMAGE utility). In this case, PSF counts the top and bottom half-inch margins as a printable area when comparing expected and actual loaded paper size. If the half-inch vertical offset default is changed to 0 in ASPUX04, perf-to-perf printing is enabled. However, if character ascenders or descenders fall outside of the printable area, they are not printed.

An alternative for processing these jobs when migrating them to a 3900 printer is RPQ 8B3997—3800 Compatibility Mode for the 3900 printer. This RPQ provides a separate microcode load for the 3900 printer to enable it to run as a 3800-3 or 3800-6 printer, operating in 3800 line mode. This RPQ is intended to facilitate the migration of 3800 workloads to 3900 printers with no need to convert COPYMODs, GRAPHMODs, JCL, and so on, or to install higher levels of software. For more information about the RPQ, consult your IBM marketing representative.

Implementation Checklist for PSF/MVS 3800 Line-Mode Conversion

To use the PSF/MVS conversion function, take the following steps:

- Make sure that AFP fonts are available in the PSF resource library for all the 3800 character sets and GRAPHMODs used by the jobs being processed.
- Make sure that electronic overlays are created and available in the PSF resource library for all 3800 forms flashes used.
- Screen the 3800 jobs to identify those that cannot be processed by PSF—that is, jobs using COPYMODs or merging lines with different-sized characters.
- Modify PSF/MVS Exit 4 to indicate jobs that require the conversion function.

You are now ready to route 3800 line-mode jobs to the 3900 or 3835 printer.

Chapter 11. Managing Resources and Selecting Special Processing Parameters

Resources are collections of data and control information that PSF can use in printing a data set. Form definitions, page definitions, fonts, page segments, overlays, and object containers are all types of PSF resources.

PSF resources can be specified in:

- Job entry subsystem (JES) initialization parameters
- PSF startup procedures
- User job control language (JCL)
- Print data set
- Form definitions and page definitions

Form definitions and page definitions are loaded into PSF storage; all other resources are downloaded to the printer. PSF determines when to load or download a resource.

After processing a data set, PSF deletes all the page segments and overlays that were downloaded to printer storage, unless the resource exit specifies otherwise. Only after this deletion is done does PSF start to process another data set. For more information, see “Resource-Deletion Call (RDC)” on page 211. The most recently used fonts generally remain in the printer.

Resources can also be deleted from printer storage to allow space for a complex page and to maintain printer throughput. Complex pages normally contain many page segments and use a variety of fonts.

Default resources can be defined when PSF is started. For instructions, see “PSF Startup Procedures for AFP Printers” on page 74, and Chapter 8, “Using Direct-Printing Mode” on page 97. Default resources are loaded into printer storage as needed and are deleted as necessary, either when space is needed for a complex page or when a device error occurs. If printer resources are deleted, PSF retransmits those resources to the printer as required.

How PSF Selects Special Processing Parameters

This section describes the following special processing parameters:

- Selecting a region size
- Printing PSF error messages
- Specifying checkpoint intervals

Some processing parameters can be specified in:

- JES initialization parameters
- PSF startup procedures
- User JCL
- Active page definitions and form definitions

PSF selects these processing parameter values in the following order:

1. In a form definition named on a JCL OUTPUT statement, including any inline form definitions named on the OUTPUT statement
2. In the user JCL
3. In JES initialization statements
4. In the PRINTDEV statement
5. In a form definition named in the PRINTDEV statement, including any inline form definitions named on the PRINTDEV statement

Selecting Region Sizes

The region-size requirements for PSF depend on:

- Installation options in the PSF startup procedure
- The number of printers the region supports
- Requirements of the individual print job

During the initialization of the FSS and the first printer FSA, PSF code, control blocks, and various buffers have to be stored. Each additional FSA requires storage for control blocks and buffers. Storage acquired during initialization remains static and is not released until the FSA is drained.

Additional storage for certain FSA control blocks is required while print jobs are being processed. This storage is dynamic, is acquired as needed, and is released after the print job is processed. Some print-job characteristics require more storage and should be considered when you calculate the region-size requirements.

Table 4 shows the minimum storage required for PSF.

Table 4. Minimum Storage Required for FSS and FSAs. M = megabyte.

FSA	Channel-attached	SNA- or TCP/IP-attached
The **first** FSA in an FSS	1M below 16M 11M above 16M	0.5M below 16M 11M above 16M
Each **additional** FSA in the FSS	0.65M below 16M 0.25M above 16M	0.1M below 16M 0.25M above 16M

Installation Options: The following installation options can affect storage requirements during the initialization of PSF:

- Size of the trace table
  - PSF default is 128K per FSA.
  - For more information on tracking, refer to *Print Services Facility/MVS: Diagnosis Guide and Reference*.
- Requirements for the exit program:
  - Size of all installation exits (same in LINKLIB)
  - Per FSA:
    Size of the exit work area as defined in APSXUCOM
    Size of GETMAINS coded by the installation exits
- Resource buffer pool
  - The buffer pool is calculated by taking the smaller of the following and multiplying that value by the number of FSAs:
    - Largest block size of all resource libraries multiplied by 4

    - Largest block size of all resource libraries multiplied by 2 and added to the next largest block size multiplied by 4
  - DCB=(BUFNO=*n*) can be coded on the DD statement in the resource library. This will cause the allocation of an additional amount of virtual storage, calculated as follows:

    ```
    (BUFNO n) * resource library block size
    ```

- The printer output buffer is calculated by the following formula:

  ```
  (BUFNO parameter + 1) * 4k per FSA
  ```

  The value of BUFNO is taken from the PRINTDEV statement. The PSF minimum is 5 in fixed storage; for ESCON channel-attached printers and printers that support 16K I/O buffers; however, the minimum BUFNO value is 9 in fixed storage.

  Note: THE BUFNO PARAMETER IS IGNORED FOR TCP/IP-ATTACHED PRINTERS AND FOR SNA PRINTERS USING VTAM.

- If the Huffman compression algorithm is used, 38 000 bytes per FSA are required to perform the compression. For more information, see page 224.

For Each FSA: Following are some characteristics of print jobs that place additional storage requirements on each FSA:

- Number of resources required to print the document—0.5K for each resource
- Number of error messages generated while formatting and printing the document—0.2K for each message group
- Size of 6 installation-generated page definitions and 6 installation-generated form definitions—approximate size of the resource
- Size of the conditional processing page buffer—maximum number of text characters on a page, multiplied by 1.5

Redirecting the Message Data Set

The errors found during the printing of a user's data set are printed as a message data set at the end of the user's print data set. If APAR OW10066 has been applied, you can redirect this message data set as a SYSOUT data set to another CLASS or DEST for viewing or printing. This function does not include redirecting messages printed within a user's data set, caused by paper jams, operator commands, and so on. If you wish, you can use installation exit 7 to suppress messages printed within a user's data set. For details, see “Resource-Management Exit: APSUX07” on page 193.

If the system is MVS/ESA 4.2.0 or higher, message data set redirection can be activated for each FSA. (For MVS systems at lower levels, PSF ignores any attempts to activate message data set redirection.) After message data set redirection is activated, all messages for an FSA are redirected to the specified SYSOUT data set; tell your users where they can find the error messages for their print data sets.

To activate redirection of message data sets, specify PIMSG=NO on the OUTPUT statement for the message data set referenced by the MESSAGE= keyword in the PRINTDEV statement. When PIMSG=NO is specified on the OUTPUT statement for the message data set, all other keywords specified on this OUTPUT statement are also attributed to the generated SYSOUT data set. For example:

```
//MSGDS  OUTPUT PAGEDEF=A06462,FORMDEF=A10110,PIMSG=NO,FORMS=STD,
//       CLASS=M,DEST=PRT129,CHARS=GT10,DATACK=BLOCK
```

In this example, each message data set will be placed in a SYSOUT data set that is sent to a destination of PRS129 with a CLASS of M. All other specified attributes are used at the time the PRS129 device prints the message data set. Attributes specified in the PRINTDEV are not included. The PRS129 device processes the SYSOUT data set as if it were a user data set generated by the PSF FSS.

The SYSOUT message data set generated is formatted as 81-byte fixed-block ANSI line data records, and can be viewed with TSO/E or printed.

An alternative way to activate redirection of message data sets is to set XTP7MDSD with APSUX07. All the keywords specified on OUTPUT statement for the message data set referenced by the MESSAGE= keyword in the PRINTDEV statement will be attributed to the generated SYSOUT data set. For details, see “Considerations on Redirecting a Message Data Set” on page 197.

For the following example we assume that CLASS=H is a held class, and each message data set is held for viewing:

```
//MSGDS  OUTPUT PAGEDEF=A06462,FORMDEF=A10110,PIMSG=NO,
//       CLASS=H
```

Notice that a page definition and a form definition are specified, because some parameters on this OUTPUT statement are also used for printing messages generated in the middle of a user's print data set (unless such messages have been suppressed by instructions in installation exit 7). The specified page definition and form definition are not used in viewing the held message-data sets.

PSF generates a separate SYSOUT data set for each print data set that contains errors. The SYSOUT data set is generated and released after the last page (or trailer page) of the associated print data set is stacked.

Note: If you specify message data set redirection when multiple FSAs are active under a single FSS, overall printer throughput may be degraded, because fewer pages are printed in the same number of CPU cycles. To minimize such degradation, reduce the number of FSAs for each FSS.

Printing PSF Error Messages

The PIMSG parameter, which can be coded in the user JCL or in the PRINTDEV statement, controls the printing of PSF error messages at the end of the data set. The PIMSG parameter specified in the user JCL takes precedence over that specified in the PRINTDEV statement. For more information on the PIMSG parameter in the user JCL, refer to *Print Services Facility/MVS: Application Programming Guide*.

Note: Do not print the PSF messages data set with a page definition that uses record formatting, data suppression, or fixed text. Also, do not use a form definition that invokes an electronic overlay. The correct page definition for printing the message data set is specified in the PSF startup procedure. For more information, see “PSF Startup Procedures for AFP Printers” on page 74.

Specifying Checkpoint Intervals

Checkpoint intervals can be specified in the following places, and PSF uses the values in the following order:

1. CKPTPAGE or CKPTSEC parameter in the user JCL

   If both of these parameters are specified in the user JCL, JES2 or JES3 determines which is used. Refer to the *JCL Reference* for your operating system, listed in Appendix K, “Related Publications,” for information.

2. Applicable JES initialization parameters:

   - JES2 initialization CKPTPAGE, CKPTSEC, and CKPTMODE parameters. See “Parameters for the JES2 Initialization Statement” on page 65.
   - JES3 initialization CKPNTPG and CKPNTSEC parameters. See “Parameters for the JES3 Initialization Statement” on page 70.

How PSF Selects Resources

PSF resources can be specified in the following places:

- In the JCL for the print job
- In JES initialization statements
- In the PRINTDEV statement
- In a page definition or form definition named in the PRINTDEV statement
- In printer defaults (fonts only)

Tables 7 through 12 on pages 119 through 122 show how the page definition, the form definition, and the fonts for a print job are selected. PSF takes the user's JCL parameters from the DD statement and the first OUTPUT statement found; the search proceeds in the following order:

1. Explicitly referred to
2. Implicitly referred to at the step level
3. Implicitly referred to at the job level

If the same parameter appears both on the DD statement and on the OUTPUT statement, PSF uses the parameter from the DD statement.

An “explicitly referred to” OUTPUT statement is one named by the OUTPUT parameter of a DD statement. An “implicitly referred to” OUTPUT statement must contain a DEFAULT=YES parameter and must not be overridden by any OUTPUT keyword on a DD statement.[7] In the following example, OUT1 is an OUTPUT statement that is explicitly referred to:

```
//JOB01   JOB     MSGCLASS=A
//STEP01  EXEC    PGM=ENDMONTH
//OUT1    OUTPUT  CHARS=GT10
//DD1     DD      SYSOUT=A,OUTPUT=*.OUT1
```

In the following example, OUTA is an OUTPUT statement that is implicitly referred to:

[7] You cannot use an “implicitly referred to” OUTPUT statement on a printer in direct-printing mode.

```
//JOB02   JOB     MSGCLASS=A
//OUTA    OUTPUT  DEFAULT=YES,CHARS=GT10
//STEP01  EXEC    PGM=ENDMONTH
//DD1     DD      SYSOUT=A
```

For more information about the OUTPUT statement, refer to the JCL reference publication for your operating system.

Selecting a Forms Flash for a 3800 Printer

To select a forms flash for the IBM 3800 printer, you use control statements, in the order shown in these tables.

Table 5. Hierarchy of Flash Selection in Deferred-Printing Mode

User JCL	FLASH specified by FORMDEF= on OUTPUT statement (See notes 1 and 2.)	
	DD statement	FLASH= (See note 3.)
	OUTPUT statement	FLASH= (See note 3.)
JES default in the current printer setup		
FLASH specified by FORMDEF= in PRINTDEV statement. (See notes 2 and 3.)		

Table 6. Hierarchy of Flash Selection in Direct-Printing Mode

User JCL	FLASH specified by FORMDEF= on OUTPUT statement (See notes 1 and 2)	
	DD statement	FLASH= (See note 3)
	OUTPUT statement	FLASH= (See note 3)
	FLASH specified by FORMDEF= in PRINTDEV statement. (See notes 2 and 3.)	

Notes:

1. If an OUTPUT statement in the user JCL names a form definition, PSF uses the forms flash specified in that form definition. If no forms flash is specified in the form definition, the user JCL can specify the forms flash on the FLASH parameter.
2. If the user JCL does not include a FLASH parameter, the operator is told (in a SETUP message) to load the default JES forms flash, if one exists. If no default JES forms flash exists (and therefore no SETUP message is issued), PSF uses any forms flash that is currently loaded in the printer.
3. If the forms flash is specified only in the form definition from the PRINTDEV statement, the user must specify a forms flash with the FLASH parameter in the USER JCL; otherwise, PSF does not activate the forms flash. Nor will the forms flash currently loaded in the printer be used.

Selecting a Copy Group

A copy group is an internal object in a form definition that controls such items as data suppression, forms flash, overlays, media source, modifications to a form, and the number of copies of each page. PSF selects a copy group as follows:

- If the print data set contains an inline form definition, and the name of that form definition matches the name of the form definition specified either in the user's JCL or in the PRINTDEV statement, the inline form definition specifies the copy group that is to be used.
- If the print data set does not contain an inline form definition, and the user JCL specifies a form definition, that form definition specifies the copy group that is to be used.
- If the user JCL does not name a form definition, PSF determines the copy group from the default form definition in the PRINTDEV statement. However, the submitter of a print job can override the specification of number of copies in that copy group by use of a COPIES parameter in the JCL. (For more information about the COPIES parameter, refer to *Print Services Facility/MVS: Application Programming Guide*.)

Form Definitions in Deferred-Printing Mode

Table 7. Hierarchy of Form Definition Selection in Deferred-Printing Mode

User JCL	OUTPUT statement	Explicitly referred to (see notes 1 and 2)	FORMDEF =
		Implicitly referred to step-level	FORMDEF =
		Implicitly referred to job-level	FORMDEF =
First inline form definition (see note 3)			
PRINTDEV statement from the PSF startup procedure			FORMDEF =

Notes:

1. This includes inline form definitions, which are specified on the OUTPUT statement either by name or with the name DUMMY.
2. This includes a form definition in a user library, which must be specified by name on the OUTPUT statement.
3. If the user JCL does not specify the form definition parameter and there are inline form definitions, the first inline form definition in the print data set is selected.

Form Definitions in Direct-Printing Mode

Table 8. Hierarchy of Form Definition Selection in Direct-Printing Mode

User JCL	Explicitly referred to OUTPUT statement (see note 1)	FORMDEF =
First inline form definition (see note 2)		
User JCL	PRINTDEV statement	FORMDEF =

Notes:

1. This includes inline form definitions, which are specified on the OUTPUT statement either by name or with the name DUMMY.
2. If the user JCL does not specify the form definition parameter and there are inline form definitions, the first inline form definition in the print data set is selected.
3. PSF does not use user libraries in direct-printing mode.

Page Definitions in Deferred-Printing Mode

Table 9. Hierarchy of Page Definition Selection in Deferred-Printing Mode

User JCL	OUTPUT statement (see note 1)	PAGEDEF =
	DD statement	FCB =
	OUTPUT statement (see note 1)	FCB =
JES default in the current printer setup (see notes 2 and 3)		
First inline page definition (see notes 4 and 5)		
PRINTDEV statement from the PSF startup procedure (see note 3)		PAGEDEF =

Notes:

1. The OUTPUT statement used is the first one found from explicit, step-level implicit, and job-level implicit.
2. Under JES2, the JES default PAGEDEF can be specified by the FCB parameter in the PRT*nnn* statement or, if this parameter is not present, by the NIFCB parameter in the PRINTDEF statement.
3. Under JES3, a page definition specified in the PRINTDEV statement is never selected, because JES3 always has a default page definition. This default page definition is taken from the CARRIAGE parameter on the DEVICE statement or, if this parameter is not present, from the CARRIAGE parameter on the OUTSERVE statement. If the OUTSERV statement has no CARRIAGE parameter, a default of 6 lines per inch is used, which invokes page definition P16. To use inline page definitions in JES3, the name of the page definition should be specified in the PAGEDEF or FCB of the user JCL.
4. For JES2, if the user JCL does not specify the PAGEDEF or FCB parameter and inline page definitions exist, the first inline page definition in the print data set is selected unless JES has set a default.
5. If an inline page definition is included in the print data set, and either its name matches the name selected or the name from the JCL is DUMMY, the inline page definition is used.
6. A page definition from a user library is used if its name matches the name selected.

Page Definitions in Direct-Printing Mode

Table 10. Hierarchy of Page Definition Selection in Direct-Printing Mode

User JCL	Explicitly referred to OUTPUT statement (see note 1)	PAGEDEF =
	DD statement (see note 1)	FCB =
	Explicitly referred to OUTPUT statement (see note 1)	FCB =
First inline page definition (see note 2)		
User JCL	PRINTDEV statement	PAGEDEF =

Notes:

1. This includes inline page definitions, which are specified by name or with the name DUMMY, on the OUTPUT or DD statement.
2. If the user JCL does not specify the PAGEDEF or FCB parameter and inline page definitions exist, the first inline page definition in the print data set is used.
3. PSF does not use user libraries in direct-printing mode.

Fonts in Deferred-Printing Mode

When PSF is printing line-format data, Table 11 applies.

Table 11. Hierarchy of Font Selection in Deferred-Printing Mode

Fonts specified by the current page definition. This includes a JES default page definition, but not a page definition specified only by the PRINTDEV statement of the PSF startup procedure.		
User JCL	DD statement	CHARS=
	OUTPUT statement (see note 1)	CHARS=
	DD statement	UCS=
	OUTPUT statement (see note 1)	UCS=
JES default font in the current printer setup (see notes 2 and 3)		
PRINTDEV statement of the PSF startup procedure		CHARS=
If the PRINTDEV statement of the PSF startup procedure names the active page definition, any fonts specified in that page definition are loaded.		
Hardware default		

Notes:

1. The OUTPUT statement used is the first one found from explicit, step-level implicit, and job-level implicit.
2. Under JES2, you can select fonts with a SET command, the UCS parameter of the PRT*nnnn* statement, or the NIUCS parameter of the JES2 PRINTDEF statement. To prevent JES2 from sending a default font name to PSF, specify UCS=0 in the PRT*nnnn* statement. For more information on selecting fonts, refer to the JES2 publication for your operating system, or to *Print Services Facility/MVS: Application Programming Guide.*
3. Under JES3, you can select fonts with the CH parameter in the *START, *RESTART, and *CALL commands. You can also use the CHARS parameter in

the DEVICE, OUTSERV, and SYSOUT statements. Because a JES3 default font is always available, hardware default fonts or fonts specified in the PRINTDEV statement are never selected. For more information on selecting fonts, refer to the JES3 publication for your operating system, or to *Print Services Facility/MVS: Application Programming Guide.*

Fonts in Direct-Printing Mode

When PSF is printing line-format data, Table 12 applies.

Table 12. Hierarchy of Font Selection in Direct-Printing Mode

Fonts specified by the current page definition (unless the current page definition is specified only by the PRINTDEV statement)		
User JCL	DD statement	CHARS =
	Explicitly referred to OUTPUT statement	CHARS =
	DD statement	UCS =
	Explicitly referred to OUTPUT statement	UCS =
	PRINTDEV statement	CHARS =
If the PRINTDEV statement names the active page definition, any fonts specified in the page definition are loaded.		
Hardware default		

Searching for a Resource Specified for a Print Job

After PSF determines the name of the resource to be used, it must find that resource. It conducts a search, in the following order:

1. If inline resources were sent in the print data set, they are searched first.
2. If user libraries were specified, they are searched in the order in which they were specified.
3. Finally, the system libraries that were specified in the PRINTDEV statement are searched in the order in which they were concatenated.

On the first occurrence of a resource bearing the name specified, PSF stops the search.

Note: Only the security libraries (if PSFMPL is active) and the system libraries are searched for resources to be used on auxiliary data sets. For B1 security information, refer to *Print Services Facility: Security Guide*.

Using Form Definitions for Basic N_UP Printing

PSF/MVS supports basic N_UP printing, which enables the submitter of a PSF to place up to 4 pages in partitions on a sheet in simplex mode, and up to 8 pages on a sheet in duplex mode. The term N_UP comes from the Page Printer Formatting Aid/370 (PPFA/370) N_UP subcommand, which the job submitter specifies when creating a form definition.

N_UP printing is supported by some, but not all, AFP printers; for a list of the printers supported, refer to *Advanced Function Presentation: Printer Information*. To find the microcode level necessary for N_UP printing, check the documentation for your printer, or consult your IBM marketing representative.

When processing a form definition containing the basic N_UP subcommand, the PSF-supported printer divides a side of a sheet into 1 to 4 equal, fixed partitions. In basic N_UP printing (that is, printing in which the form definition does not include the explicit placement of pages in the partitions), the printer places page *n* in the first partition, page *n+1* in the second partition, page *n+2* in the third partition, and page *n+3* in the fourth partition. PSF processes the pages sequentially, in the order in which they occur in the data stream. PSF processes N_UP commands in both simplex and duplex printing mode, and applies both overlays and page segments to the pages.

Keep the following in mind when using N_UP printing:

- To create form definitions containing the N_UP subcommand, you must have PPFA/370 1.1.0 with APAR PN27884 applied.[8]
- Because the job submitter is placing, in effect, more pages of data on a sheet, the submitter must select appropriately small fonts.
- N_UP with guaranteed print labeling is not supported.
- PSF counts multiple pages on a single sheet and reports accurate accounting information.
- Although N_UP should not adversely affect the performance of PSF, the printer uses more buffer space when accumulating data for multiple pages than when accumulating data for a single page.
- PSF supports full error recovery and repositioning.
- With N_UP printing, PSF processes data suppression as before: PSF suppresses data for the entire side of a sheet, not for the individual pages in the partitions.
- With PSV/MVS APAR OW03243, the user can specify additional page eject actions when invoking a medium map.

Using Form Definitions for Enhanced N_UP Printing

With application of PSV/MVS APAR OW03243 and PPFA/370 APAR PN54401, N_UP printing can enable you to place the pages at any location on either side of the sheet. With enhanced N_UP printing on the 3900-0W1, 3900 Duplex, 3130, and 3935 printers, users can:

- Place a page relative to any partition origin on either side of the sheet, in any orientation, and of any size that fits on the sheet. Users can place multiple pages relative to the same origin, when the total number of pages does not exceed the N_UP limit for that sheet.
- Place overlays relative to any partition origin, or page origin, with or without variable page data from the application program.
- Specify a different rotation for each page.
- Specify one or more different overlays for each page.
- Specify a different offset for each page.
- Place pages of any size that fit the physical sheet. Page size is not limited by the partition size.

[8] APAR PN27884 has been available since January 28, 1994.

Using PSF, the Interface Attachment Feature, and the MICR Postprocessor

The MICR postprocessor enables magnetic ink character recognition (MICR) printing if:

- The device supports MICR printing.
- MICR printing is enabled.
- The font is a MICR font.

Because every host MICR font contains a list to indicate to PSF that the font is a MICR font, you do not have to do anything to prepare the system or PSF for MICR printing. Be aware, however, that if the user requests a MICR font for a device that either has disabled MICR support or does not support MICR printing, PSF issues a message and does not print the page.

Consider using a RACF-controlled user library to manage the MICR fonts. For more information, see “User Libraries in Direct-Printing Mode” on page 48.

For additional information about MICR printing, refer to *Advanced Function Presentation: Printer Information.*

Maintaining Resource Integrity

To maintain the integrity of resources, consider the following:

- Page definitions, form definitions, overlays, page segments, fonts, and object containers can be placed inline in the print data set, to ensure that the correct resources are used on the target system.

- You can update resource data sets, but you must make sure that PSF still has access to those data sets.

- You can use other AFP licensed programs—for example, Overlay Generation Language/370 and Page Printer Formatting Aid/370—to update resources in PSF libraries. A PPTNDSI bit in the MVS program-properties table specifies when resources can be updated. When this bit is set to 1, AFP programs can update resources in the PSF library while PSF is running. When this bit is set to 0, and DISP=SHR, AFP programs cannot update resources in the PSF library until either the FSA or the printer has stopped running.

- If a resource in a library is modified after it is loaded into virtual storage or into printer storage,[9] PSF might not have deleted the unmodified resource from storage.

  To make sure that the modified resource is used, do one of the following:

  – Put the resource in a user library.
  – Drain and then restart the FSA or the printer.
  – Change the name of the modified resource, and use the new name.
  – Code the resource inline.
  – Use the resource exit.

[9] Page definitions and form definitions are loaded into virtual storage; fonts, page segments, and overlays are loaded into printer storage.

- PSF deletes user libraries and inline resources when it finishes processing the data set for which they were downloaded.
- Do not compress PSF library data sets while PSF is active.
- Exclude PSF library data sets from HSM migration.

Chapter 12. Understanding PSF Data Protection and Security

PSF and the printers it supports protect data in the following ways:

- Data cannot be read back into the host processor by means of the available printer commands.
- Characters stored in the print buffer cannot be easily interpreted, because they have been translated from EBCDIC form.
- All raster character patterns stored in the printer (excluding the permanently resident fonts) can be erased. PSF can erase data remaining in the data buffers and in the accumulator, including page segments and electronic overlay data. PSF erases this data when a printer is started or stopped.
- No pages are left in the hardware buffer. PSF makes sure that all pages are in the stacker before responding to a STOP WRITER command. When a printer using continuous forms is in direct-printing mode, the nonprocess runout (NPRO) parameter must be specified to ensure that all pages are stacked. If NPRO is not specified, the operator must perform the NPRO.
- When a form change is requested, PSF provides a software NPRO to make sure that all printed pages are in the stacker before it allows the forms to be changed. For example, when a user specifies SYSOUT=(*class*[,, *formname*]) for printing on a continuous-forms printer, PSF causes the printer to perform an NPRO on the last page of one form before JES issues the message telling the operator to load the new form. This is done as a data security measure, to make sure that all the pages of the first data set are successfully printed before the operator loads a new form. Because of this software NPRO, additional blank pages can be printed to move the previously printed pages into the stacker.
- You have responsibility for specific protection for data sets in a shared environment such as Distributed Print Function (DPF) of PSF/2. Sensitive material stored in the PSF resource library should be marked PRIVATE. For more information, see “Using APSRMARK to Mark Resources” on page 153.
- Resources stored in libraries defined in the PSF startup procedure are available to all print jobs. These libraries should not be used for sensitive resources, nor should they be protected by read passwords. Consider using RACF-controlled user libraries to manage sensitive resources. For more information, see “User Libraries in Direct-Printing Mode” on page 48.
- PSF uses inline resources only for the data set with which they are sent inline; PSF ensures that the resources will not be used again after the processing of that data set is completed.

For specific printers, refer to the publications for those printers.

Security Labeling

Consider using security labeling to protect printed output. Security labeling can be made either voluntary or mandatory. For more information, refer to *Print Services Facility: Security Guide*.

Controlling Print Complete Notification

PSF can notify as many as four users when a print job is complete. To request that such a message be sent, use the NOTIFY JCL keyword in the OUTPUT statement. To read more about specifying the NOTIFY keyword, refer to *PSF/MVS: Application Programming Guide*.

To control these notices, profiles must be created in the SMESSAGE class, and the users who are to send notification messages must be given authority for access to the profiles. To use the security provided by these profiles, activate both the SMESSAGE class and the SETROPTS RACLIST processing for the SMESSAGE class. For more detailed information, refer to the *RACF Security Administrator's Guide*.

Chapter 13. Supporting Resident Fonts

Resident fonts are font resources that reside in a printer. Their use can significantly decrease the amount of data that PSF must send over slow-speed communication lines.

Not all printers support resident fonts, and of the ones that do support them, some support more than others. To learn which AFP printers support resident fonts, and how many each of them supports, refer to *AFP: Printer Information*.

Resident fonts for different printers are shipped on different media. For the 3812 printer, they are shipped on a diskette; for the 3820 printer, they are shipped as a hardware attachment; and for the 3916 printer they are shipped internal to the printer, installed and ready to use. For information about how resident fonts for a particular printer are shipped, refer to the publications for that printer.

Figure 41 is a diagram of a printer that supports resident fonts.

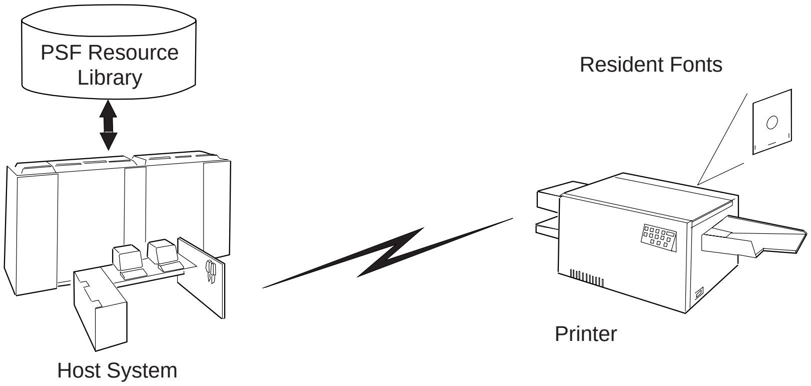


Figure 41. A Printer That Supports Resident Fonts

Notes:

1. If you are using a printer with resident fonts stored on a diskette, you can change the set of fonts by installing a different diskette. If you change diskettes, you must do a power-on reset (POR) on that printer to notify PSF of the font change.
2. The host version and the printer-resident version of a font may not be identical, especially if your installation has edited either version of the font. Also, printer-resident fonts and fonts stored in other libraries may differ, even though the fonts have the same name.
3. For printers attached through Workstation PrintManager (WPM, available in Japan) or the Distributed Print Function (DPF) of PSF/2, PSF does not support printer-resident fonts, even though the printer itself may support them. PSF Direct, however, does support printer-resident fonts.
4. Some printers, such as the 3130 printer, can capture downloaded fonts. For more information, see Chapter 14, “Marking Fonts for Font Capture” on page 141.

PSF provides two methods to access resident fonts:

- APSRFTBL (hard-coded table), used to identify resident fonts in printers that support only symbol sets—for example, the 4224, 4230, 4234, 4247, 6408, and 6412 printers. For more information, refer to *AFP: Printer Information*.
- APSRMARK (utility), used for all other printers that support resident fonts.

PSF Resident Font Table (APSRFTBL)

PSF uses the APSRFTBL module, supplied with PSF in the SYS1.SAMPLIB data set, to identify the symbol sets resident in the 4224, 4230, 4234, 4247, 6408, and 6412 printers. PSF does not download fonts to these printers. APSRFTBL contains tables that associate the font character set and code page for a particular host font with the identifier of the resident symbol set. APSRFTBL does not contain executable code. You can change, assemble, and install it (or link-edit it with the *replace* option) to provide your own version of this module if you want to change the resident symbol sets to be used for particular code pages and character sets.

The sample APSRFTBL module is found in Appendix B, “Font Resource Tables Supplied by PSF.”

Changing APSRFTBL

You can add, delete, and change any entries in the APSRFTBL font resource table except the last line, which contains the characters 'FFFFFFFF' and indicates the end of that particular table; all code for the table must be placed before the 'FFFFFFFF' line. For the contents shipped with PSF, see “APSRFTBL Font Resource Table” on page 267.

If the APSRFTBL resident-font tables do not contain an entry for a font requested in a print job, PSF issues an error message. Processing of the print job stops on the page where the font was requested, but continues on the succeeding pages. To prevent this error, you can edit the resident-font tables in APSRFTBL and associate the font character set or code page of any font referenced in a print job with the identifier of a resident font.

For example, suppose a user has specified Prestige 10-point font (character set C0S0PR10) for a print job being sent to a 4224 printer, but that font is not included in APSCS24V, the character set table in the APSRFTBL module for 4224 font character sets. You can decide to map that font to the Gothic 10-point symbol set that is resident in the 4224 printer.

Note: For a list of resident symbol sets, see the publications for your printer.

To change the APSCS24V table in the APSRFTBL module for a 4224 printer, duplicate the entry line for the Gothic 10-point font and change the character set name in the duplicated line to the character set name of the font specified in the print job, as in this example:

```
DC      CL8'C0D0GT10',XL2'001A',XL2'0000'
DC      CL8'C0S0PR10',XL2'001A',XL2'0000'
```

Because the resident-font table is searched sequentially, put the fonts used most often at the beginning of the table.

Note: Do not change or delete the last entry in the tables; the entry CL8'FFFFFFFF' marks the end of each table. Do not change the table labels.

After changing the table, you must assemble the APSRFTBL module and install (link-edit) APSRFTBL into the applicable LINKLIB (for example, the PSF LINKLIB).

Similarly, you can modify the relevant table in the APSRFTBL module as needed for your printer: the APSCS24V for the 4230 printer; the APSCS34V for the 4234; the APSCS47V for the 4247 printer; or the APSCS64X for the 64XX printer.

Similarly, you can modify the APSCS34V table in the APSRFTBL module for a 4230, 4234, 6408, or 6412 printer.

PSF Resident-Font Utility—APSRMARK

Host-font resources—code pages and character sets—can reside in a printer. To identify such resources to PSF as potentially printer-resident, use the APSRMARK utility, which is provided by PSF.

APSRMARK is a batch program invoked with control statements. You can use it to do the following:

- Mark all or individual host code pages and character sets of partitioned data sets, so that PSF will know that there is a printer-resident version of the resource.
- Report the marking status of all or individual members of partitioned data sets.
- Report the marking status of a font resource found in a sequential file that can be subsequently moved into a font library.

Fonts that have been updated by APSRMARK are called *marked* fonts. When you mark a font, you are essentially updating certain structured fields. For example, when you mark a code page, you are updating the Begin Code Page structured field.

To mark a font, you provide JCL statements to run the APSRMARK utility and identify fonts as either PUBLIC or PRIVATE. If the version of the font stored in the PSF resource libraries is marked PRIVATE, PSF downloads the font to the printer. A font stored in a PSF resource library must be marked PUBLIC so that the equivalent printer-resident font can be used; this prevents PSF from downloading the font to the printer. If a font is not marked, is treated as if it were marked PRIVATE. To prevent unnecessary downloading of fonts, mark PUBLIC any font that has an equivalent printer-resident font. Some fonts in the AFP Font Collection, and the 4028 Font Metrics, are shipped *already* marked PUBLIC. For more information, see page 133.

If someone updates a code page or character set on the host that was marked PUBLIC, you should run APSRMARK to change the marking of the host resources—for example, by entering new date and time values. Otherwise the change will not be reflected in the resources selected for printing a file. This is especially important when a resource is copied from a system library and modified for use in a private library. Unless you have run APSRMARK on the modified resource, you might use the old resource by mistake.

Notes:.

1. You cannot concatenate partitioned data sets as input to APSRMARK.

2. To identify 4224, 4230, 4234, 4247, 6408, or 6412 printer-resident font resources to PSF, see “PSF Resident Font Table (APSRFTBL)” on page 130.
3. To read about the sample jobs that PSF supplies for marking resources, see Appendix A, “APSRMARK Sample Jobs.”
4. To use APSRMARK to identify resources for use by DPF, see Chapter 16, “Marking Resources for Distributed Print Function.”

   You can use APSRMARK to mark the same resources for DPF and for printer-resident use; the latter is described in “Using DPF Fonts and Printer-Resident Fonts with APSRMARK” on page 161. For printers that support resident outline fonts, you can use APSRMARK to mark host outline fonts. If the printer is supported through WPM or DPF, however, you cannot use outline fonts.

APSRMARK Considerations

Before marking fonts for resident use, consider the following:

- Font character sets containing only font metrics are shipped with PSF as an aid in formatting documents that call for 4028 printer-resident fonts, for which PSF has no host equivalents. Because a 4028 metric-only font contains no raster data, PSF must use the printer-resident font that has the same name. Therefore the 4028 metric-only fonts shipped have been marked by use of the APSW4028 sample APSRMARK job. Code pages shipped with PSF have also been marked PUBLIC.

  Do not mark metric-only font character sets PRIVATE.

- Some printers do not support all combinations of inline direction and character rotation for resident fonts. If users require combinations that are not available for a resident font, mark the host version of the affected font PRIVATE, so that PSF uses it instead of the printer-resident font. Fonts marked PRIVATE are sent to the printer as needed. For more information on the print directions and character rotations supported, refer to *AFP: Printer Information*.
- You can use APSRMARK to mark resources that can be used as inline resources, or to mark user library resources.
- When using resident fonts, you may see some of the following differences:
  - Different character raster patterns
  - Different font metrics, such as character positioning
  - Different characters included in the character set for the font

  If any of the above differences occurs, consider a couple of alternatives. One is to mark any altered fonts residing in the PSF resource library as PRIVATE, so that the host version of the font is used instead of the printer-resident version (which has not been altered). The other is to provide a host version of the font that is identical with the printer-resident font being used.

- You can use a single font library for multiple 240-pel IPDS printers, even if different fonts are resident in each printer. You can mark the fonts resident in any printer as PUBLIC, and PSF will query the printer to determine whether the resident font is available. To determine which fonts each printer supports, refer to *AFP: Printer Information*.

  Note: Do not use a 240-pel library for a 300-pel printer, even if the resident fonts in the 300-pel printer are equivalent (same names) to those in the 240-pel library. Instead, create and mark a 300-pel version. For more information, see

“Using APSRMARK” on page 133. The single 300-pel font library can be used for multiple 300-pel IPDS printers. Mark the fonts resident in any printer as PUBLIC.

Using APSRMARK

Before PSF can use a font that is resident in a particular printer, you must complete the following procedure:

1. Choose the printer-resident fonts you want to use. For a list of the fonts resident on a particular printer, refer to *AFP: Printer Information*.
2. Determine the member names for both the code page and the font character set of the corresponding font in the system library. A list of code-page names and font character set names is given in *AFP: Printer Information*.

   Note: For 300-pel (300-dpi) printers such as the 3112, 3116, 3916, 3935, and 4028 printers, run the APSRCF30 utility to convert the 240-pel fixed fonts to 300-pel relative metric fonts *before* using APSRMARK. APSRCF30 is a font-scaling utility; it is described in Appendix F, “Font Conversion Program, 240-pel to 300-pel (APSRCF30).”
3. Using APSRMARK, mark the chosen code page and font character set. For more information on marking the fonts, see “Syntax of APSRMARK Control Statements” and “APSRMARK Control Statement Parameters for Resident Fonts.”
4. Make sure that the application generating the print job references the font correctly, using the member names marked in step 3.
5. Make sure that the marked member is the first member with that name found by PSF, which selects resources according to hierarchy. See “Searching for a Resource Specified for a Print Job” on page 122.

Syntax of APSRMARK Control Statements

The control statements that provide input to APSRMARK must be preceded by the SYSIN DD statement, and must use the following syntax:

- Each statement must be an 80-column record. The first 71 columns can contain valid APSRMARK parameters. APSRMARK ignores columns 73–80, which can contain a sequence number. A statement that does not fit on one line can be continued on the next line by entering a continuation character (any nonblank character) in column 72.
- Blank spaces are optional; they can be placed between parameters for readability.
- An asterisk (*) in column 1 denotes a comment statement.

If any syntax or I/O errors are found in the control statement parameters, APSRMARK reports them.

APSRMARK Control Statement Parameters for Resident Fonts

To mark a host font to indicate that a printer-resident font should be used in its place, if one is available, use the following control statements.

Note: Sample jobs are shipped with PSF; see Appendix A, “APSRMARK Sample Jobs.” For specific values of RRID, use the following formula:

RRID Specifies a resident resource identifier of 8 hexadecimal digits, as follows:

- For the font character set:

  ```
  RRID = FGID + FW
  ```

  where

  ```
  FGID = a font identifier of 4 hexadecimal digits
  FW   = a font width of 4 hexadecimal digits
  +      indicates concatenation
  ```

  Both FGID and FW must be specified as above, whether either or both fields contain only zeros or leading zeros.

- For the code page:

  ```
  RRID = GCSGID + CPGID
  ```

  where

  ```
  GCSGID = a graphic character set identifier
           of 4 hexadecimal digits
  CPGID  = a code page identifier
           of 4 hexadecimal digits
  +        indicates concatenation
  ```

  Both GCSGID and CPGID must be specified as above, whether either or both fields contain only zeros or leading zeros.

Refer to *AFP: Printer Information* for information such as:

- FGID and FW values for fonts on the 3270 IDS diskette[10]
- GSCGID and CPGID values for fonts on the 3270 IDS diskette
- FGID and FW values for fonts on the DCF/GML diskette[11]
- GSCGID and CPGID values for fonts on the DCF/GML diskette
- FGID and FW values for fonts on the 4028 font cards
- GSCGID and CPGID values for fonts on the 4028 font cards

RRDATE Consists of any value of 5 or 6 decimal digits.

Note: For a 3820 printer, this value is defined in the printer publications. For all other printers with resident fonts, this value can specify the date on which the font was marked.

RRTIME Consists of any value of 8 decimal digits.

Note: For a 3820 printer, this value is defined in the printer publications. For all other printers with resident fonts, this value can specify the time at which the font was marked.

{PUBLIC|PRIVATE}
Must be specified for both the code page and the character set. If a resource is marked PRIVATE, it is sent from the host to the printer when needed. If a resource is marked PUBLIC, and an equivalent resource is resident in the printer, PSF does not send the host resource from the

[10] 3270 Information Display Systems diskette.

[11] Document Composition Facility/General Markup Language diskette.

host to the printer; PSF uses the printer-resident resource. Figure 42 shows an example of marking resources.

Do not mark metric-only font character sets PRIVATE.

Using Control Statements to Mark a Font

Figure 42 shows an example of how you can use JCL control statements to mark a font. In this example, SYS1.FONTLIBB (TID0BASE) contains the code page, and SYS1.FONTLIBB (C0D0GT10) contains the character set. This JCL copies members T1D0BASE and COD0GT10 from SYS1.FONTLIBB to SYS1.FONT3812. The marked copies will be in SYS1.FONT3812.

```
//EX3812 JOB  (X,YY),MSGLEVEL=1,MSGCLASS=A,USER=SAM,PASSWORD=TODAY
//STEP1  EXEC  PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=A
//IN1    DD UNIT=3380,DSN=SYS1.FONTLIBB,DISP=SHR,VOL=SER=USR000
//OUT1   DD UNIT=3380,DSN=SYS1.FONT3812,DISP=SHR,VOL=SER=USR000
//*
//SYSIN  DD *
         INDD=IN1,OUTDD=OUT1,MEMBER=T1D0BASE,PUBLIC,RRID=000003EA,RRDATE=88044,RRTIME=00000000
         INDD=IN1,OUTDD=OUT1,MEMBER=C0D0GT10,PUBLIC,RRID=00280090,RRDATE=88044,RRTIME=00000000
/*
```

Figure 42. Sample JCL for Marking Resident Fonts. Your syntax may differ; for syntax rules, see “Syntax of APSRMARK Control Statements” on page 133.

APSRMARK Control Statement Parameters (General)

INDD=*DDname*

Specifies the name of the input data set to be used when you mark or report resources. The INDD parameter can name either a sequential data set or a partitioned data set. This parameter is required.

MEMBER=*name***|ALL**

This parameter is required for partitioned data sets. Do not specify this parameter for sequential data sets.

name

Specifies that resource marking or resource reporting is to be done for a single member (primary or alias) of a partitioned data set. When resources are marked, both the primary member and its alias are marked unless the primary has been updated or deleted since the alias was generated; in that case, the alias is not updated. An alias whose primary has been updated or deleted can be marked only by specifying the name of the alias and specifying the NEWNAME keyword.

All

Specifies that resource marking or resource reporting is to be done for all members of a partitioned data set. When resources are marked, all primary members and aliases whose primaries have not been updated or deleted are marked. Aliases whose primaries have been updated or deleted are not marked. When resources are reported, only primary members are listed. When marking multiple fonts for printer-resident use, do not specify MEMBER=ALL; rather, specify each member name in multiple control statements. When you specify the REPORT parameter, however, you can specify MEMBER=ALL.

NEWNAME=*name*

Specifies a new name for a member of a partitioned data set. This parameter is optional and has meaning only when a specific member is specified by MEMBER=*name*. The newly named member will be marked and will become a primary member without an associated alias.

OUTDD=*DDname*

Specifies the DD name of the output data set in which the marked resources are to be placed. You can use the REPLACE parameter to mark resources (or change resource marking) and replace them in the same data set. This parameter is required.

REPLACE

Specifies that existing members of the specified output partitioned data set are to be overwritten (replaced). REPLACE must be specified in order to overwrite members. Use REPLACE to mark resources initially, and then replace them in the same data set. REPLACE is also used to change resource markings. This parameter is optional, and applies only to partitioned data sets.

REPORT

Specifies that a report is to be formatted and printed. The report lists how resources are marked. The INDD and MEMBER parameters determine which resources are reported.

- INDD specifies the name of the input data set.
- MEMBER=ALL specifies that resource reporting is to be done on all primary members of a partitioned data set.
- MEMBER=*name* specifies that resource reporting is to be done on a single primary or alias member of a partitioned data set.

Sample Control Statements for APSRMARK

In the following examples, lowercase letters represent information you specify to customize the JCL.

Marking a Member with Resident Font Information

Figure 43 shows how to mark a member of a partitioned data set.

```
//example1 JOB  (X,YY),MSGLEVEL=1,MSGCLASS=A,USER=userid,PASSWORD=password
//*
//STEP1  EXEC  PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//IN1    DD UNIT=dasd,DSN=dsname1,DISP=SHR,VOL=SER=volser
//OUT1   DD UNIT=dasd,DSN=dsname2,DISP=SHR,VOL=SER=volser
//SYSIN  DD *
     INDD=IN1,OUTDD=OUT1,MEMBER=codepage,PUBLIC,RRID=rrid,RRDATE=rrdate,RRTIME=rrtime
     INDD=IN1,OUTDD=OUT1,MEMBER=charset,PUBLIC,RRID=rrid,RRDATE=rrdate,RRTIME=rrtime
/*
```

Figure 43. Example of the Use of JCL Statements to Mark a Member of a Data Set. Your syntax may differ; for syntax rules, see “Syntax of APSRMARK Control Statements” on page 133.

Figure 43 contains the following JCL statements:

IN1 DD

Defines the input partitioned data set that contains the member to be marked by APSRMARK.

OUT1 DD

Defines the output partitioned data set that will contain the member marked by APSRMARK.

Note: The data set attributes for OUT1 must match those for IN1. The data set name for OUT1 can match the data set name for IN1.

SYSIN DD *

Contains the marked member name (resource name) and the three specific values for the RRID, RRDATE, and RRTIME parameters.

Generating a Report of Resident Font Marking

Figure 44 shows how to generate a report of the results from Figure 43 on page 136.

```
//example2 JOB ,'IBM USER',MSGCLASS=A
//step1  EXEC PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//IN1      DD UNIT=dasd,DISP=SHR,DSN=dsname,VOL=SER=volser
//SYSIN    DD *
      INDD=IN1,MEMBER=name,REPORT
/*
```

Figure 44. Example of the Use of JCL Statements in Generating a Report. Your syntax may differ; for syntax rules, see “Syntax of APSRMARK Control Statements” on page 133.

Figure 43 on page 136 contains the following JCL statements:

IN1 DD

Defines the partitioned data set that contains the member that will be reported.

SYSIN DD

Precedes the control statements that specify the parameters for the report to be printed.

Figure 45 on page 138 is an example of a printed report.

```
APS580I *GET REPORT OF FONTS MARKED AS PRINTER-RESIDENT
APS580I  INDD=IN,REPORT,MEMBER=ALL
APS580I  C0D0GB10 PUBLIC  SAVE   89-07-21 17:58:25 95D6 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00270090  00000 00000000
APS580I  C0D0GB12 PUBLIC  SAVE   89-07-21 17:58:25 3E0D 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00450078  00000 00000000
APS580I  C0D0GI12 PUBLIC  SAVE   89-07-21 17:58:25 9CBF 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00440078  00000 00000000
APS580I  C0D0GT10 PUBLIC  SAVE   89-07-21 17:58:25 BE89 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00280090  00000 00000000
APS580I  C0D0GT12 PUBLIC  SAVE   89-07-21 17:58:25 32B5 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00420078  00000 00000000
APS580I  C0D0GT15 PUBLIC  SAVE   89-07-21 17:58:25 4717 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00E60060  00000 00000000
APS580I  C0D0GT20 PUBLIC  SAVE   89-07-21 17:58:25 8C38 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +01190048  00000 00000000
APS580I  C0D0GT24 PUBLIC  SAVE   89-07-21 17:58:25 10A7 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +01220036  00000 00000000
APS580I  C0L0KATA PUBLIC  SAVE   89-07-21 17:58:25 C4B9 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +002C0090  00000 00000000
APS580I  C0L00AOA PUBLIC  SAVE   89-07-21 17:58:25 10B6 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00130090  00000 00000000
APS580I  C0L00BOA PUBLIC  SAVE   89-07-21 17:58:25 10A7 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00030090  00000 00000000
APS580I  C0S0AE10 PUBLIC  SAVE   89-07-21 17:58:25 AB9E 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +002D0090  00000 00000000
APS580I  C0S0AE20 PUBLIC  SAVE   89-07-21 17:58:25 9EDC 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00270090  00000 00000000
APS580I  C0S0BITR PUBLIC  SAVE   89-07-21 17:58:25 EAEC 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +009B0078  00000 00000000
APS580I  C0S0CB10 PUBLIC  SAVE   89-07-21 17:58:25 3F2F 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +000B0090  00000 00000000
```

Figure 45 (Part 1 of 2). Printed Output of Printer-Resident Report

```
APS580I  C0S0CB12 PUBLIC  SAVE   89-07-21 17:58:25 C660 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +006C0078  00000 00000000
APS580I  C0S0CE12 PUBLIC  SAVE   89-07-21 17:58:25 533B 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00550078  00000 00000000
APS580I  C0S0CI10 PUBLIC  SAVE   89-07-21 17:58:25 071E 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00120090  00000 00000000
APS580I  C0S0CR10 PUBLIC  SAVE   89-07-21 17:58:25 8F81 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +000B0090  00000 00000000
APS580I  C0S0DOTR PUBLIC  SAVE   89-07-21 17:58:25 447F 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00AF0078  00000 00000000
APS580I  C0S0D224 PUBLIC  SAVE   89-07-21 17:58:25 C59D 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00CC006C  00000 00000000
APS580I  C0S0EBTR PUBLIC  SAVE   89-07-21 17:58:25 6BB3 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00A30078  00000 00000000
APS580I  C0S0EITR PUBLIC  SAVE   89-07-21 17:58:25 C660 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00A20078  00000 00000000
APS580I  C0S0ELTR PUBLIC  SAVE   89-07-21 17:58:25 D07A 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00AD0078  00000 00000000
APS580I  C0S0ESTR PUBLIC  SAVE   89-07-21 17:58:25 5766 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00A00078  00000 00000000
APS580I  C0S0LB12 PUBLIC  SAVE   89-07-21 17:58:25 3901 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +006E0078  00000 00000000
APS580I  C0S0LR12 PUBLIC  SAVE   89-07-21 17:58:25 6B64 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00570078  00000 00000000
APS580I  C0S0OB10 PUBLIC  SAVE   89-07-21 17:58:25 9453 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00260090  00000 00000000
APS580I  C0S0OR10 PUBLIC  SAVE   89-07-21 17:58:25 92C0 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00050090  00000 00000000
APS580I  C0S0PB12 PUBLIC  SAVE   89-07-21 17:58:25 963F 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +006F0078  00000 00000000
APS580I  C0S0PI12 PUBLIC  SAVE   89-07-21 17:58:25 92C0 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00700078  00000 00000000
APS580I  C0S0PR12 PUBLIC  SAVE   89-07-21 17:58:25 B9A9 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00560078  00000 00000000
APS580I  C0S0SR12 PUBLIC  SAVE   89-07-21 17:58:25 2C5E 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +00030090  00000 00000000
APS580I  C0T055B0 PUBLIC  SAVE   89-07-21 17:58:25 E6E1 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +1137004E  00000 00000000
APS580I  C0T05500 PUBLIC  SAVE   89-07-21 17:58:25 C300 660241123081 DSNAME VOLSER1.RESFONTS
APS580I +11370042  00000 00000000
```

Figure 45 (Part 2 of 2). Printed Output of Printer-Resident Report

Diagnosing Problems with Printer-Resident Fonts

Table 13 summarizes some of the problems resulting from incorrect use of APSRMARK, and provides possible solutions.

Table 13 (Page 1 of 2). Resident Font Diagnosis

Problem	Possible solution
PSF loads the host library version of the font into the printer and does not use the printer-resident font.	The code page, the font character set, or both are not marked correctly. See “Using APSRMARK” on page 133. If the resources are marked correctly, make sure that your MO:DCA-P document (or your page definition) is using the names of the resources that are marked as resident. Make sure that the marked resource is found by PSF before any unmarked version of the resource. For more information, refer to “Searching for a Resource Specified for a Print Job” on page 122.

Table 13 (Page 2 of 2). Resident Font Diagnosis

Problem	**Possible solution**
A printer-resident font is used, but not the one specified.	Some printer-resident fonts are simulated by the printer and may appear different from the version in the system library. For more information on font simulation done by your printer, refer to the publications for your printer. If a simulation is not the problem, make sure that the correct RRID values are specified. For more information on RRID values, see “APSRMARK Control Statement Parameters for Resident Fonts” on page 133.
Printing results in messages APS818I and APS717I, where the sense bytes 0, 1, and 19 of message APS717I contain the values `X'02'`, `X'18'`, and `X'02'`, respectively.	You may have specified a character rotation or print direction that your printer does not support. For information about the character rotations and print directions your printer supports, refer to the publications for your printer or to *AFP: Printer Information*.
Printing results in message APS279I.	Font character sets containing only font metrics are provided as an aid in formatting documents that reference certain printer-resident fonts that have no equivalent downloadable host font. These metric-only font character sets must be marked by the APSRMARK utility with the PUBLIC keyword specified, so that PSF can access the printer-resident font. Make sure that the font identified in the message is marked correctly. **Note:** APSW4028 is a sample APSRMARK job that contains control statements to correctly mark “4028 Font Metrics” character sets, which can be resident in printers such as the 3116, 3916, and 4028 printers. If the “4028 Font Metrics” character set has been marked correctly, make sure that the printer contains the desired resident font.
Printing results in message APS28OI.	If your installation is using 240-pel fonts to access 300-pel resident fonts, make sure that the font cartridge installed in the printer contains the 300-pel resident font that you are trying to access with a marked 240-pel resident font. If the correct 300-pel resident font does not exist in the printer, PSF attempts to download the host font. APS280I is issued when PSF attempts to download a 240-pel host font to a 300-pel printer. Make sure that the host font library is a 300-pel version (contains 300-pel character sets). For more information on creating and marking a 300-pel font library, see “Using APSRMARK” on page 133.

Chapter 14. Marking Fonts for Font Capture

Some printers, such as the 3130 printer, can capture downloaded fonts. The concept is similar to "dynamic caching" in DPF. For more information on DPF dynamic caching, see Chapter 16, "Marking Resources for Distributed Print Function" on page 153.

Both raster fonts and outline fonts can be captured. Captured fonts automatically become new temporary printer-resident fonts. This improves performance on future jobs that use the same fonts. If the fonts are ever deleted from the printer, PSF automatically downloads the host version the next time it is needed. The printer manages the captured font library.

A font character set or code page can be captured only if it is marked PUBLIC. Fonts marked PRIVATE are always downloaded, never captured. You can use the APSRMARK utility to mark a resource.

The following sections explain what needs to be done for different kinds of font.

Single-Byte Raster Font Capture

For raster fonts that were previously shipped as a part of the PSF product, run the sample APSRMARK jobs that were shipped with PSF. For more information, see Appendix A, "APSRMARK Sample Jobs" on page 265.

The single-byte raster fonts shipped in the IBM APF Font Collection (program number 5648-113) are already marked correctly both for activating resident fonts and for capturing fonts.

For single-byte raster fonts that you create or change, run APSRMARK to mark them PUBLIC, and specify the GRID part in RRID and the new date and time in RRDATE and RRTIME. For information on using APSRMARK, see "PSF Resident-Font Utility—APSRMARK" on page 131.

When a font should not be captured, mark it PRIVATE. For example, if there is a copyright restriction against having a copy of a font on the printer hard drive, mark the font PRIVATE.

Single-Byte Outline Font Capture

All single-byte outline fonts shipped in the IBM AFP Font Collection (program number 5648-113) are already marked correctly both for activating resident fonts and for capturing fonts.

Any single-byte outline fonts created or changed by the IBM Type Transformer program may need to be marked PUBLIC, but they contain all the other information needed for font capture. You do not need to specify RRID, RRDATE, or RRTIME when you run APSRMARK.

When a font should not be captured, mark it PRIVATE. For example, if there is a copyright restriction against having a copy of a font on the printer hard drive, mark the font PRIVATE.

Double-Byte Raster Font Capture

If you are migrating from a printer with resident fonts to a printer that captures fonts, the host libraries are already marked correctly.

For new installations of DBCS font product IDs 5771-AGB, 5771-AEK, 5771-AFZ, 5771-AFW, or 5771-AEN, run the sample APSRMARK job included with the product.

For raster double-byte sections that you create or change, run APSRMARK to mark them PUBLIC, and specify the GRID part in RRID and the new date and time in RRDATE and RRTIME. For information on using APSRMARK, see "PSF Resident-Font Utility—APSRMARK" on page 131.

When a font should not be captured, mark it PRIVATE. For example, if there is a copyright restriction against having a copy of a font on the printer hard drive, mark the font PRIVATE.

Double-Byte Outline Font Capture

The double-byte fonts in "IBM AFP CJK Metric-Only Fonts," RPQ 8A8080, are not suitable for capture, because they do not contain font data. However, they are correctly marked for activating the printer-resident version of the fonts. You do not need to run APSRMARK.

When you use the IBM Type Transformer program to create downloadable DBCS outline fonts, you can specify CAPTURE=YES|NO. CAPTURE=YES corresponds to PUBLIC, and CAPTURE=NO to PRIVATE. You need to run APSRMARK only if you want to change the setting.

Chapter 15. Supporting Outline Fonts

This chapter explains why IBM supports printers that use outline fonts, how outline fonts and raster fonts differ, and how you can tailor PSF so that print-job submitters can use outline fonts.

IBM's Font Strategy

IBM's long-range strategy is to enable you to print any kind of document on any printer in your enterprise. In a move toward fulfilling this strategy, IBM is supporting printers that can use outline font technology,[12] because outline font technology is resolution-independent.

The IBM AFP Font Collection contains a large selection of fonts in both outline and raster formats. (For descriptions of IBM's fonts, see Appendix K, “Related Publications.”) Say that a document for which an outline font is requested is sent to a printer on which the font libraries have been installed. If the printer supports outline fonts, the document is printed in the requested font. If the printer supports only raster fonts, the outline font is mapped to an equivalent raster font, and the document is printed in that raster font.

If a raster font is requested for a document, PSF handles the request in accordance with the specifications you make when you configure PSF. You can configure PSF to use the raster font or to map to an equivalent outline font. For information about how to use PSF Exit 7 (APSUX07) to map to an equivalent outline font, see “Considerations Related to Mapping to Outline Fonts” on page 202.

You can create fonts and add them to your font libraries by using Type Transformer. For information about creating new fonts, refer to *IBM AFP Fonts: Type Transformer User's Guide*.

Understanding Outline Fonts

For raster fonts, each character (bitmap) is made up of an arrangement of pels. Characters are stored in their full-size, bitmap forms, with the attendant characteristics of style, size, weight, height, width, and resolution.

For outline fonts, each character shape is represented by a mathematical expression, which, when combined with the font metrics, determines the style, shape, height, width, and space of the character. Outline fonts are stored in this format, along with vertical scaling information for determining the size of the characters, as well as information about the rotation of each character. When a printer that supports outline fonts needs to use font characters, it creates a rasterized bitmap for each character, at the printer's resolution, and rotates the characters as instructed.

Character sets and code pages for outline fonts are maintained as separate objects in the printer. Thus, outline font character sets can be shared among coded fonts.

[12] For example, the 3900-0W1, 3900 Duplex, 3900 wide/duplex, 3935, and 3130 printers.

This allows an application using outline fonts to use less printer storage, because it can use a character set or a code page that is already available in the printer and scale it to a different size.

Outline fonts are resolution-independent; that is, they can be used by any printer that supports outline fonts. When a printer needs to use a given character (and not until then), it creates a rasterized bitmap from an outline font, scaling the character larger or smaller as needed. See Figure 46. A raster font character, by contrast, is stored and used in its full, rasterized size.

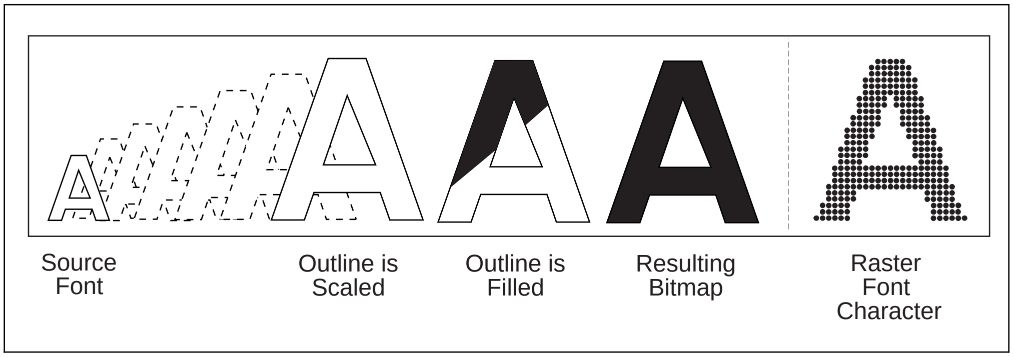


Figure 46. Comparison of an Outline Font Character with a Raster Font Character

Concatenating Outline Fonts with Other Library Resources

An outline font can be placed inline in the print data, in a user library, in a security library, or in a system library. Outline fonts can be stored in the same library as raster fonts, or they can be stored in a separate library. For more information, see "Concatenating Defined Libraries to System Libraries" on page 47.

Creating New Fonts

You can create new fonts by using, for example, Type Transformer. With Type Transformer, you can create or modify raster font character sets, outline font character sets, and code pages in FOCA format, and store them in host font libraries for use with PSF.

You can add entries to PSF's font mapping tables that identify new outline font character sets and their raster equivalents in different sizes. Then these fonts can be specified in print jobs with either outline font member names, raster font member names, or GRIDs. Working from the mapping tables, PSF selects either the raster version or the outline version of your new fonts, whichever can be used with your printer.

| To add a new single-byte outline font:

1. Using Type Transformer, create or modify a FOCA outline font character set and coded font from an Adobe Type 1 font. If you do not have a code page that can be used with the new font character set, you also need to use Type Transformer to create or modify a code page.
2. If it is possible that the jobs that reference outline fonts will need to be printed on a printer that supports raster font technology only, use Type Transformer to generate equivalent FOCA raster fonts in the sizes you need and in the resolution that your raster font printer supports (either 240 or 300 pels).

3. If the font is to be included as an inline font and it is possible that the job will be routed to printers that do not support outline fonts, generate the FOCA raster equivalent of this font (using Type Transformer) in the size and resolution you need, and include it inline.
4. Move the new FOCA outline font and raster font character sets, code page, and coded font to the MVS host font libraries.
5. Modify the Font-Name-to-GRID Mapping table (APSRNTGM) and the GRID-to-Font-Name Mapping table (APSRGTNM) to include the new outline font character set, the equivalent raster font character sets, and the new code page, as described below.
6. Assemble the user mapping tables, and link-edit them with PSF.
7. Make sure that the MVS host font libraries are concatenated in the PSF startup procedure.

Capturing New Outline Fonts

Some printers, such as the 3130 printer, can capture downloaded fonts. For more information, see Chapter 14, “Marking Fonts for Font Capture” on page 141.

Font Mapping

PSF sometimes needs to equate a font specified in a print job to another font. The process of equating fonts is called “font mapping.” The mapping process should not be confused with font substitution (which is done by use of APSUX07). The mapping process is used to verify that the specified font can be used in the current environment, or to find an equivalent font that can be used.

PSF maps fonts in any of three situations:

- When the technology of the font is incompatible with the technology that the printer supports. For example, PSF maps fonts when an outline font is specified in a print job but the printer supports only raster fonts. This process is called a *technology-compatibility map*.

  Note: Double-byte fonts can be technology-mapped only at the coded font prefix level. The are not mapped through the user mapping tables.

- When a GRID, rather than member names, is specified in the print data. PSF uses the mapping tables to find the corresponding code page and character set member names compatible with the technology of the attached printer and the setting of XTP7MTOF (see next bullet). This process is called a *GRID map*.
- When APSUX07 requests that PSF map fonts to outline fonts (XTP7MTOF = ON), and the attached printer supports outline fonts. This process is called an *outline map*.

The user does not select technology-compatibility mapping or GRID mapping; PSF performs these maps automatically.

The outline map function is an option that can be selected by using APSUX07. See “Considerations Related to Mapping to Outline Fonts” on page 202 for more information.

The different mapping functions are performed at different times during the processing of the font. The choice of mapping functions to be performed depends on how the font was specified in the print job. For example, a coded font is mapped by one process, and a pair of code page and character set names by another process.

Timing of Mapping Operations

The technology-compatibility map and the GRID map functions are performed before the Resource-Access call to APSUX07, if that call is enabled. This means that APSUX07 can be called with font names different from those specified in the job.

The outline map function happens after the Resource-Access call to APSUX07 and before the Resource-Load call to APSUX07, if those calls are enabled.

Mapping Algorithms

PSF uses different processes to map one technology to another; the process chosen depends on how the input font is specified.

- For coded font names, the mapping process consists of merely changing the prefix of the input font, X0 or X1–XG, to another prefix, XZ; the rest of the name remains the same. This is called "prefix mapping."
- For fonts specified as GRIDs, PSF uses the mapping tables to find the corresponding code page and character set names.
- For fonts specified as pairs of code page and character set names, PSF uses the mapping tables to find another pair of code page and character set names or to verify that the original names are of the desired technology.

Using the Font Mapping Tables

PSF has font mapping tables of two types: user tables that you can modify, and default internal system tables. These tables are used to map single-byte fonts of one technology to single-byte fonts of another technology.

The user mapping tables can be used to:

- Map input single-byte font names to equivalent raster font or outline font names. For example, if you have added new outline fonts or raster fonts to your host system, you can add entries to the user mapping tables to ensure that you can use these fonts when printing on printers that support either font technology.
- Selectively disable the mapping of the internal system tables. For example, if you want to use a particular raster font in a printer that supports both raster fonts and outline fonts, and if XTP7MTOF = ON in APSUX07, you can add an entry to the user mapping tables to disable the mapping of this particular font, so that it has to be printed as a raster font.

Appendix I, "Mapping Tables for Outline Fonts," contains two PSF font mapping tables:

- APSRGTNM, which maps *global resource identifiers* (GRIDs) to font names
- APSRNTGM, which maps font names to GRIDs

These tables are optional, in that they do not have to be link-edited with the rest of PSF. APSRNTGM is also used by the font library utility. For more information, see “Using the Font Library Update (FLU) Utility” on page 259.

When performing the mapping functions, PSF searches the user tables first. A mapping in the user tables takes precedence over a similar mapping in the internal system tables.

Each time you modify the mapping tables, you must compile them and link-edit them with PSF.

Adding a New Mapping of a Raster Font to an Outline Font

To add a new mapping of a raster font name to an outline font name:

1. To make sure that outline mapping is requested, set XTP7MTOF=ON in APSUX07.
2. Add a new entry to the Font-Name-to-GRID-Mapping table (APSRNTGM) to associate the raster font character set with its GRID parts, font technology, and vertical font size. For example:

   ```
   DC CL8'C0MYRAST'    CHARACTER SET NAME
   DC XL1'00'          RESERVED
   DC XL1'05'          TECHNOLOGY ID - RASTER
   DC XL2'1111'        FGID
   DC XL2'2222'        GCSGID
   DC XL2'0000'        FONT WIDTH
   DC XL2'00C8'        VERTICAL FONT SIZE
   ```

   You can find the FGID, GCSGID, and font size for all IBM-supplied AFP fonts in *IBM AFP Fonts: Font Summary*, G544-3810. For non-IBM supplied fonts or raster fonts created from OEM outlines using the IBM Type Transformer product, either determine the values that were used when the raster fonts were created or assign values yourself using FGID and GCSGID values in the customer-assigned range of 65280 to 65534 (decimal).

   Code the font technology for the raster font character set as XL1'05'.

   Specify a size parameter so that the printer knows what point size to use to render the selected font. You can specify Font Width, Vertical Font Size, or both. Because Vertical Font Size is more easily understood, it is preferable to use this value. Vertical Font Size is the point size multiplied by 20 (decimal).

   If you have created a specific code page for use with this font character set, use the GCSGID associated with that code page. Otherwise, code the GCSGID entry with a value of XL2'FFFF'.

3. If you are using a standard code page with this character set, you need not create a code page entry; PSF will use the internal system tables. If you have created a specific code page for use with your character set, add a new entry to the Font-Name-to-GRID Mapping table (APSRNTGM) to associate that code page with its GRID parts. For example:

   ```
    DC CL8'T1MYCDPG'    CODE PAGE NAME
    DC XL2'1234'        CPGID
    DC XL2'2222'        GCSGID
   ```

4. Add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM), to associate the FGID and GCSGID of the corresponding raster font character set entry in the Font-Name-to-GRID Mapping table (APSRNTGM) with the name of

the outline font character set. For example, an outline font character set that corresponds to the raster font entry in item 1 above can be coded as follows:

```
DC XL2'1111'        FGID
DC XL2'2222'        GCSGID
DC XL2'0000'        FONT WIDTH
DC XL1'00'          RESERVED
DC XL1'1F'          TECHNOLOGY ID - OUTLINE
DC XL2'0000'        VERTICAL FONT SIZE
DC CL8'CZMYOUTL'    CHARACTER SET NAME
```

Use the same FGID and GCSGID parameters that were assigned to the raster font. These values link the raster font to the outline font.

Code the font technology for the outline font character set as `XL1'1F'`.

This outline font should use the same code page as the corresponding raster font character set entry in the Font-Name-to-GRID Mapping table (APSRNTGM). If you have created a specific code page for use with this font character set, use the GCSGID associated with that code page. Otherwise, code the GCSGID entry with `XL2'FFFF'`.

This entry is also for GRID-to-outline font mapping.

5. If you are using a standard code page with this character set, you need not create a code page entry; PSF will use the internal system tables. If you have created a specific code page for use with your character set, add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM), to associate that code page with its GRID parts. For example:

```
DC XL2'1234'        CPGID
DC XL2'2222'        GCSGID
DC CL8'T1MYCDPG'    CODE PAGE NAME
```

Adding a New Mapping of an Outline Font to a Raster Font

To add a new mapping of an outline font to a raster font:

1. Add a new entry to the Font-Name-to-GRID Mapping table (APSRNTGM), to associate the outline font character set with its GRID parts, font technology, and vertical font size. For example:

```
DC CL8'CZMYOUTL'    CHARACTER SET NAME
DC XL1'00'          RESERVED
DC XL1'1F'          TECHNOLOGY ID - OUTLINE
DC XL2'1111'        FGID
DC XL2'2222'        GCSGID
DC XL2'0000'        FONT WIDTH
DC XL2'0000'        VERTICAL FONT SIZE
```

You can find the FGID, GCSGID, and font size for all IBM-supplied AFP fonts in *IBM AFP Fonts: Font Summary*, G544-3810. For non-IBM supplied fonts or outline fonts created from OEM outlines using the IBM Type Transformer product, either determine the values that were used when the raster fonts were created or assign values yourself using FGID and GCSGID values in the customer-assigned range of 65280 to 65534 (decimal).

Code the font technology for the outline font character set as `XL1'1F'`.

If you have created a specific code page for use with this font character set, use the GCSGID associated with that code page. Otherwise, code the GCSGID entry with a value of `XL2'FFFF'`.

2. If you are using a standard code page with this character set, you need not create a code page entry; PSF will use the internal system tables. If you have created a specific code page for use with your character set, add a new entry to the Font-Name-to-GRID Mapping table (APSRNTGM), to associate that code page with its GRID parts. For example:

   ```
   DC CL8'T1MYCDPG'    CODE PAGE NAME
   DC XL2'1234'        CPGID
   DC XL2'2222'        GCSGID
   ```

3. Add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM), to associate the FGID and GCSGID of the corresponding outline font character set entry in the Font-Name-to-GRID Mapping table (APSRNTGM) with the name of the raster font character set. For example:

   ```
   DC XL2'1111'        FGID
   DC XL2'2222'        GCSGID
   DC XL2'0000'        FONT WIDTH
   DC XL1'00'          RESERVED
   DC XL1'05'          TECHNOLOGY ID - RASTER
   DC XL2'00C8'        VERTICAL FONT SIZE
   DC CL8'C0MYRAST'    CHARACTER SET NAME
   ```

   Note: If you use the outline font at different point sizes, you need an entry in the GRID-to-Font-Name Mapping table (APSRGTNM) for the raster font in each size.

   Use the same FGID and GCSGID parameters that were assigned to the outline font. These values link the outline font to the raster font.

   Code the font technology for the raster font character set as `XL1'05'`.

   Specify a size parameter so that the printer knows what point size to use to render the selected font. You can specify Font Width, Vertical Font Size, or both. Because Vertical Font Size is more easily understood, it is preferable to use this value. Vertical Font Size is the point size multiplied by 20 (decimal).

   If you specify a non-zero font width, this entry is also used for GRID-to-raster font mapping.

   This raster font should use the same code page as the corresponding raster font character set entry in the Font-Name-to-GRID Mapping table (APSRNTGM). If you have created a specific code page for use with this font character set, use the GCSGID associated with that code page. Otherwise, code the GCSGID entry with `XL2'FFFF'`.

4. If you are using a standard code page with this character set, you need not create a code page entry; PSF will use the internal system tables. If you have created a specific code page for use with your character set, add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM), to associate that code page with its GRID parts. For example:

   ```
   DC XL2'1234'        CPGID
   DC XL2'2222'        GCSGID
   DC CL8'T1MYCDPG'    CODE PAGE NAME
   ```

Disabling the Mapping of a Raster Font to an Outline Font

To disable the mapping of a raster font to an outline font, do the following:

1. Add an entry to the Font-Name-to-GRID Mapping table (APSRNTGM), to associate the raster font character set name with an FGID of XL2'0000'.

   ```
   DC CL8'C0MYRAST'    CHARACTER SET NAME
   DC XL1'00'          RESERVED
   DC XL1'05'          TECHNOLOGY ID - RASTER
   DC XL2'0000'        FGID
   DC XL2'0000'        GCSGID
   DC XL2'0000'        FONT WIDTH
   DC XL2'0000'        VERTICAL FONT SIZE
   ```

Adding a New Mapping of a GRID to a Raster Font

If you have added a new mapping for an outline font to this raster font character set, you already have added an entry to map the GRID to a raster font. To use the GRID-to-raster font mapping, be sure that the font width is coded with a non-zero value.

If the GRID-to-Font-Name Mapping table (APSRGTNM) does not contain an entry for your raster font character set, do the following:

1. Add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM) to associate the FGID, the GCSGID, and the size to the raster font character set name. For example:

   ```
   DC XL2'1111'        FGID
   DC XL2'2222'        GCSGID
   DC XL2'0080'        FONT WIDTH
   DC XL1'00'          RESERVED
   DC XL1'05'          TECHNOLOGY ID - RASTER
   DC XL2'0000'        VERTICAL FONT SIZE
   DC CL8'C0MYRAST'    CHARACTER SET NAME
   ```

   You can find the FGID, GCSGID, and font size for all IBM-supplied outline fonts in *IBM AFP Fonts: Font Summary*, G544-3810. For non-IBM supplied fonts or outline fonts created from OEM outlines using the IBM Type Transformer product, either determine the values that were used when the raster fonts were created or assign values yourself using FGID and GCSGID values in the customer-assigned range of 65280 to 65534 (decimal).

   Code the font technology for the raster font character set as XL1'05'.

   For raster font character sets, specify a font width. You may also specify a vertical font size.

   If you have created a specific code page for use with this font character set, use the GCSGID associated with that code page. Otherwise, code the GCSGID entry with XL2'FFFF'.

2. If you are using a standard code page with this character set, you need not create a code page entry; PSF will use the internal system tables. If you have created a specific code page for use with your character set, add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM), to associate that code page with its GRID parts. For example:

   ```
   DC XL2'1234'        CPGID
   DC XL2'2222'        GCSGID
   DC CL8'T1MYCDPG'    CODE PAGE NAME
   ```

Adding a New Mapping of a GRID to an Outline Font

If you have added a new mapping for a raster font to this outline font character set, you already have added an entry to map the GRID to an outline font.

If the GRID-to-Font-Name Mapping table (APSRGTNM) does not contain an entry for your outline font character set, do the following:

1. Add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM) to associate the FGID and the GCSGID with the outline font character set name. For example:

   ```
   DC XL2'1111'       FGID
   DC XL2'2222'       GCSGID
   DC XL2'0000'       FONT WIDTH
   DC XL1'00'         RESERVED
   DC XL1'1F'         TECHNOLOGY ID - OUTLINE
   DC XL2'0000'       VERTICAL FONT SIZE
   DC CL8'CZMYOUTL'   CHARACTER SET NAME
   ```

   You can find the FGID, GCSGID, and font size for all IBM-supplied raster fonts in *IBM AFP Fonts: Font Summary*, G544-3810. For non-IBM supplied fonts or raster fonts created from OEM outlines using the IBM Type Transformer product, either determine the values that were used when the raster fonts were created or assign values yourself using the FGID and GCSGID values in the customer-assigned range of 65280 to 65534 (decimal).

   Code the font technology for the outline font character set as `XL1'1F'`.

   For outline font character sets, specify font width and vertical font size with XL2'0000'.

   If you have created a specific code page for use with this font character set, use the GCSGID associated with that code page. Otherwise, code the GCSGID entry with `XL2'FFFF'`.

2. If you are using a standard code page with this character set, you need not create a code page entry; PSF will use the internal system tables. If you have created a specific code page for use with your character set, add a new entry to the GRID-to-Font-Name Mapping table (APSRGTNM), to associate the code page with its GRID parts. For example:

   ```
   DC XL2'1234'       CPGID
   DC XL2'2222'       GCSGID
   DC CL8'T1MYCDPG'   CODE PAGE NAME
   ```

Chapter 16. Marking Resources for Distributed Print Function

Distributed Print Function (DPF), a part of the PSF/2 program, runs on an IBM PS/2 personal computer. DPF makes it possible to store PSF resources—fonts, overlays, and page segments—in the fixed-disk storage of the PS/2 personal computer. As Figure 47 shows, these stored resources can then be used by a printer that is attached to the PS/2 personal computer.

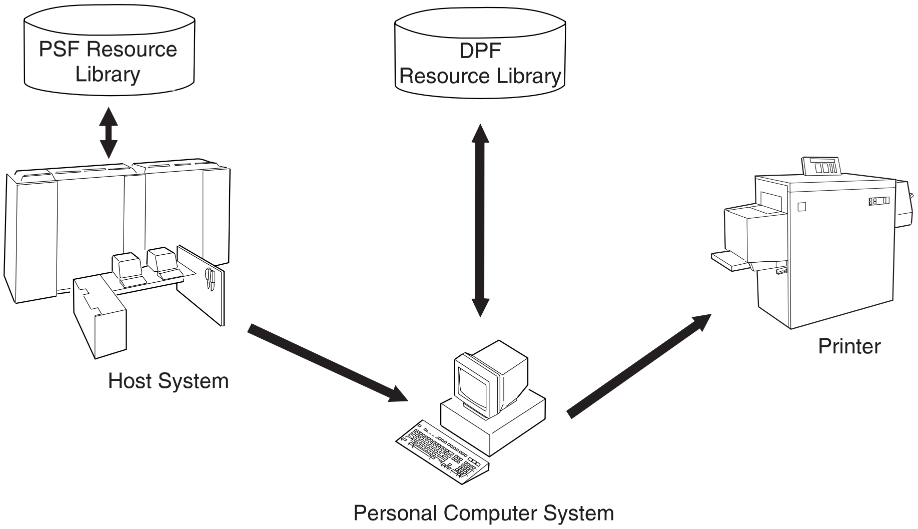


Figure 47. Access to Resources by DPF

Note: The PSF Direct function of PSF/2 and PSF for AIX does *not* store resources in the intermediate device.

If requested, DPF accesses these resources directly from the fixed disk and loads them into the printer at System/390 channel-emulation speeds, thus reducing the amount of data that PSF sends over slow-speed communication lines.

PSF provides the APSRMARK utility, which specifies how DPF is to manage resources stored in the PSF resource library. This chapter describes the JCL statements that are used to invoke the APSRMARK utility.

For more information about DPF, refer to the publications for PSF/2.

Note: You can mark a resource just once and then use it with DPF. You do not need to mark separate resources for each program.

Using APSRMARK to Mark Resources

Before you can store a PSF resource in the DPF resource library, you must mark (identify) that resource, using the APSRMARK utility.

APSRMARK is a batch program invoked with control statements that mark:

- Sequential data sets
- Individual members of partitioned data sets
- All members of a partitioned data set

To mark a resource, you provide JCL statements to run the APSRMARK utility. Marking a resource updates certain structured fields for that resource. For example, marking an overlay updates the Begin Overlay structured field.

Notes:

1. You cannot concatenate partitioned data sets as input to APSRMARK.
2. You can also use APSRMARK to identify resident fonts. See Chapter 13, “Supporting Resident Fonts.”
3. You can use APSRMARK to identify the same resources for use by DPF and printer-resident fonts. With DPF, however, you cannot use printer-resident resources; you must use the resources in the DPF resource library. For more information, see “Using DPF Fonts and Printer-Resident Fonts with APSRMARK” on page 161.
4. DPF does not support outline fonts.
5. You can use APSRMARK to mark inline resources and use them on printers attached through DPF or printers having resident fonts. PSF uses these marked resources just as it uses other resources marked with APSRMARK.

Control-Statement Syntax for APSRMARK

A **SYSIN DD** statement precedes the control statements that provide input to APSRMARK. These statements use the following syntax:

- Control statements must be 80-column records.
  - The first 71 columns can contain valid APSRMARK parameters.
  - Statements that do not fit on one line can be continued on the next line by using a continuation character (any nonblank character) in column 72.
  - APSRMARK ignores columns 73 through 80. If you wish, you can put a sequence number in these columns.
- Blanks are optional and can be placed between parameters to make the statement more readable.
- An asterisk (*) in column 1 denotes a comment statement.

APSRMARK reports any syntax errors and I/O errors found in the parameters of the control statement.

Control-Statement Parameters for APSRMARK

Each APSRMARK control statement can contain the following parameters:

INDD=*DDname*
Specifies the name of the input data set you should use when you mark or report resources. The INDD parameter can name either a sequential data set or a partitioned data set. This parameter is required.

MEMBER=*name***|All**
This parameter is required for partitioned data sets. Do not specify this parameter for sequential data sets.

name
Specifies that resources are to be marked or reported for a single member (primary or alias) of a partitioned data set. When resources are marked,

both the primary member and its aliases are marked unless the primary has been updated or deleted after the alias was generated, in which case the alias is not updated. An alias whose primary has been updated or deleted can be marked only by specifying the name of the alias and specifying the NEWNAME keyword.

ALL

Specifies that resources are to be marked or reported for all the members of a partitioned data set. When resources are marked, all primary members and aliases whose primaries have not been updated or deleted are marked. Aliases whose primaries have been updated or deleted are not marked. When resources are being reported, only primary members are listed. When marking multiple fonts for printer-resident use, do not specify MEMBER=ALL. Instead, specify each member name in multiple control statements. However, you can specify MEMBER=ALL when you specify the REPORT parameter.

NEWNAME=*name*

Specifies a new name for a member of a partitioned data set. This parameter is optional and has meaning only when a specific member is specified by MEMBER=*name*. The newly named member will be marked and will become a primary member without an associated alias.

OUTDD=*DDname*

Specifies the output data set where the marked resources are placed. You can use the REPLACE parameter to mark resources (or change resource marking) and replace them in the same data set. The OUTDD parameter is required.

{PRIVATE|PUBLIC}

Specifies the storage and use of PSF resources:

- Resources marked PRIVATE cannot be stored in the DPF resource library; instead, they are “passed through” to DPF for the one-time use of your printing application. (The default marking is PRIVATE, and PSF treats any unmarked resources as PRIVATE.)
- Resources marked PUBLIC are sent from the host system and stored in the DPF resource library just once; this eliminates the need for the host to send the resource to the printer each time it is printed.

You must code either PUBLIC or PRIVATE on each control statement.

REPLACE

Specifies that existing members of the specified output partitioned data set are to be overwritten (replaced). Use REPLACE either to mark resources initially and replace them in the same data set, or to change existing resource markings. This parameter is optional and pertains only to partitioned data sets.

Note: If you change a PSF resource, you must use APSRMARK to mark it again so that the changed version of the resource is managed correctly by DPF.

REPORT

Specifies that a report is to be formatted and printed. This report lists how resources are marked. The INDD and MEMBER parameters determine which resources are reported.

- INDD specifies the name of the input data set.

- MEMBER=ALL specifies that resources are to be reported for all the primary members of a partitioned data set.
- MEMBER=*name* specifies that resources are to be reported for a single primary or alias member of a partitioned data set.

Examples of Control Statements for APSRMARK

This section contains APSRMARK control statements for:

- Creating a PSF resource library with all members marked PUBLIC
- Creating a PSF resource library with some members marked PRIVATE
- Modifying an existing PSF resource library
- Printing a report of resource markings for a partitioned data set

In these four examples, lowercase letters represent JCL that you can customize for your organization.

PSF provides sample jobs used for marking resources. For more information, see Appendix A, “APSRMARK Sample Jobs.”

Creating a PSF Resource Library with All Members Marked PUBLIC

Figure 48 shows the APSRMARK control statements for creating a copy of each member in a partitioned data set (for example, the system font library) and marking them PUBLIC.

```
//example1 JOB ,'IBM USER',MSGCLASS=A
//STEP1  EXEC PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//IN1      DD UNIT=dasd,DISP=SHR,
//            DSN=dsname1,
//            VOL=SER=volser
//OUT1     DD UNIT=dasd,DISP=(NEW,KEEP),
//            DSN=dsname2,
//            SPACE=(same as IN1),
//            DCB=(same as IN1),
//            VOL=SER=volser
//SYSIN    DD *
  INDD=IN1,OUTDD=OUT1,PUBLIC,MEMBER=ALL
/*
```

Figure 48. Sample JCL for Creating a PSF Resource Library and Marking all Members PUBLIC

The sample JCL in Figure 48 uses the following statements:

IN1 DD
Defines the input partitioned data set that contains the members to be marked by APSRMARK.

OUT1 DD
Creates a new output partitioned data set that will contain the PUBLIC members.

Note: The data set attributes for OUT1 must match those for IN1.

SYSIN DD *

Precedes the control statements specifying that all members of IN1 are to be marked PUBLIC and placed in OUT1.

Creating a PSF Resource Library with Some Members Marked PRIVATE

Figure 49 shows the APSRMARK control statements for creating a partitioned data set that contains only PUBLIC members and for then marking some members in the same data set as PRIVATE. Notice that this task is accomplished in two separate JCL steps. Members in an existing partitioned data set can be processed more than once in the same JCL step.

```
//example2 JOB ,'IBM USER',MSGCLASS=A
//STEP1  EXEC PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//IN1      DD UNIT=dasd,DISP=SHR,
//            DSN=dsname1,
//            VOL=SER=volser
//INOUT1   DD UNIT=dasd,DISP=(NEW,KEEP),
//            DSN=dsname2,
//            SPACE=(same as IN1),
//            DCB=(same as IN1),
//            VOL=SER=volser
//SYSIN    DD *
  INDD=IN1,OUTDD=INOUT1,PUBLIC,MEMBER=ALL
/*
//STEP2  EXEC PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//INOUT1   DD UNIT=dasd,DISP=SHR,
//            DSN=dsname2,
//            VOL=SER=volser
//SYSIN    DD *
  INDD=INOUT1,OUTDD=INOUT1,PRIVATE,REPLACE,MEMBER=name1
  INDD=INOUT1,OUTDD=INOUT1,PRIVATE,REPLACE,MEMBER=name2
/*
```

Figure 49. Sample JCL for Creating a PSF Resource Library and Marking Some Members PRIVATE

The sample JCL in Figure 49 uses the following statements:

STEP1

IN1 DD

Defines the input partitioned data set that contains the members to be marked by APSRMARK.

INOUT1 DD

Creates a new output partitioned data set that contains the PUBLIC members.

Note: The data set attributes for INOUT1 must match those for IN1.

SYSIN DD *

Precedes the control statements specifying that all members of IN1 are to be marked PUBLIC and placed in INOUT1.

STEP2

INOUT1 DD

Defines the existing partitioned data set that contains members to be changed from PUBLIC to PRIVATE.

SYSIN DD *

Precedes the control statements specifying that members *name1* and *name2* will be marked PRIVATE and replaced in INOUT1.

Modifying an Existing PSF Resource Library

Figure 50 shows the APSRMARK control statements for changing one member in a partitioned data set from PRIVATE to PUBLIC. This example also shows how to take members from one partitioned data set and place them in a different partitioned data set.

```
//example3 JOB ,'IBM USER',MSGCLASS=A
//STEP1  EXEC PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//INOUT1   DD UNIT=dasd,DISP=SHR,
//            DSN=dsname1,
//            VOL=SER=volser
//IN2      DD UNIT=dasd,DISP=SHR,
//            DSN=dsname2,
//            VOL=SER=volser
//OUT2     DD UNIT=dasd,DISP=SHR,
//            DSN=dsname3,
//            VOL=SER=volser
//SYSIN    DD *
  INDD=INOUT1,OUTDD=INOUT1,PUBLIC,REPLACE,MEMBER=name1
  INDD=IN2,OUTDD=OUT2,PUBLIC,MEMBER=name2
  INDD=IN2,OUTDD=OUT2,PUBLIC,MEMBER=name3
/*
```

Figure 50. Sample JCL for Modifying an Existing PSF Resource Library

The JCL in Figure 50 uses the following statements:

INOUT1 DD

Defines the existing partitioned data set that contains a member to be changed from PRIVATE to PUBLIC.

IN2 DD

Defines the input data set that contains the members to be marked PUBLIC.

OUT2 DD

Defines the output partitioned data set that will contain the members from IN2 marked PUBLIC.

SYSIN DD *

Precedes the control statements that specify how APSRMARK will mark these members:

- The first INDD and OUTDD statements specify that member *name1* is to be marked PUBLIC and replaced in INOUT1.
- The second and third INDD and OUTDD statements specify that the members *name2* and *name3* in IN2 are to be marked PUBLIC and placed in OUT2.

Printing a Report of Resource Markings in a Partitioned Data Set

Figure 51 shows the APSRMARK control statements for requesting a printed report on the resource-marking status of all members of a partitioned data set.

```
//example4 JOB ,'IBM USER',MSGCLASS=A
//step1  EXEC PGM=APSRMARK
//*
//SYSPRINT DD SYSOUT=class
//IN1      DD UNIT=dasd,DISP=SHR,DSN=dsname,VOL=SER=volser
//SYSIN    DD *
  INDD=IN1,MEMBER=ALL,REPORT
/*
```

Figure 51. Sample JCL for Reporting Resource Markings in a Partitioned Data Set

The JCL in Figure 51 uses the following statements:

IN1 DD

Defines the partitioned data set that contains the members for which a resource-marking report is to be printed.

SYSIN DD *

Precedes the control statements that specify the parameters for the report.

Figure 52 on page 160 contains an example of a printed report.

```
APS580I  INDD=IN,REPORT,MEMBER=ALL
APS580I  C0D0GB10 PUBLIC  SAVE   89-07-21 12:50:56 95D6 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GB12 PUBLIC  SAVE   89-07-21 12:50:56 3E0D 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GI12 PUBLIC  SAVE   89-07-21 12:50:56 9CBF 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GT10 PUBLIC  SAVE   89-07-21 12:50:56 BE89 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GT12 PUBLIC  SAVE   89-07-21 12:50:56 32B5 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GT15 PUBLIC  SAVE   89-07-21 12:50:56 4717 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GT20 PUBLIC  SAVE   89-07-21 12:50:56 8C38 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0GT24 PUBLIC  SAVE   89-07-21 12:50:56 10A7 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0D0S012 ??????  ????   ??-??-?? ??:??:?? ???? ???????????? ?????? ???????.???????
APS580I  C0L0KATA PUBLIC  SAVE   89-07-21 12:50:56 C4B9 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0L00AOA PUBLIC  SAVE   89-07-21 12:50:56 10B6 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0L00BOA PUBLIC  SAVE   89-07-21 12:50:56 10A7 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0AE10 PUBLIC  SAVE   89-07-21 12:50:56 AB9E 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0AE20 PUBLIC  SAVE   89-07-21 12:50:56 9EDC 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0BITR PUBLIC  SAVE   89-07-21 12:50:56 EAEC 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0CB10 PUBLIC  SAVE   89-07-21 12:50:56 3F2F 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0CB12 PUBLIC  SAVE   89-07-21 12:50:56 C660 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0CE12 PUBLIC  SAVE   89-07-21 12:50:56 533B 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0CI10 PUBLIC  SAVE   89-07-21 12:50:56 071E 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0CR10 PUBLIC  SAVE   89-07-21 12:50:56 8F81 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0DOTR PUBLIC  SAVE   89-07-21 12:50:56 447F 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0D224 PUBLIC  SAVE   89-07-21 12:50:56 C59D 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0EBTR PUBLIC  SAVE   89-07-21 12:50:56 6BB3 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0EITR PUBLIC  SAVE   89-07-21 12:50:56 C660 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0ELTR PUBLIC  SAVE   89-07-21 12:50:56 D07A 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0ESTR PUBLIC  SAVE   89-07-21 12:50:56 5766 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0LB12 PUBLIC  SAVE   89-07-21 12:50:56 3901 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0LR12 PUBLIC  SAVE   89-07-21 12:50:56 6B64 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0OB10 PUBLIC  SAVE   89-07-21 12:50:56 9453 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0OR10 PUBLIC  SAVE   89-07-21 12:50:56 92C0 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0PB12 PUBLIC  SAVE   89-07-21 12:50:56 963F 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0PI12 PUBLIC  SAVE   89-07-21 12:50:56 92C0 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0PR12 PUBLIC  SAVE   89-07-21 12:50:56 B9A9 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0S0SR12 PUBLIC  SAVE   89-07-21 12:50:56 2C5E 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T055B0 PUBLIC  SAVE   89-07-21 12:50:56 E6E1 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T05500 PUBLIC  SAVE   89-07-21 12:50:56 C300 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T05580 PUBLIC  SAVE   89-07-21 12:50:56 F67F 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T075F0 PUBLIC  SAVE   89-07-21 12:50:56 8EBF 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T075N0 PUBLIC  SAVE   89-07-21 12:50:56 F27C 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T07500 PUBLIC  SAVE   89-07-21 12:50:56 46B6 660241123081 DSNAME VOLSER1.INFONTS
APS580I  C0T15500 PUBLIC  SAVE   89-07-21 12:50:56 B2EC 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1D0BASE PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1GDP500 PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1L0OCRB PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1L0OCR1 PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1L02773 PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1S0AE10 PUBLIC  SAVE   89-07-21 12:50:56 3D83 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1000420 PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
APS580I  T1000424 PUBLIC  SAVE   89-07-21 12:50:56 3F06 660241123081 DSNAME VOLSER1.INFONTS
```

Figure 52. Sample Output from a Resource-Marking Report

Note: A question mark with a member of a partitioned data set indicates that no marking data is available for that particular member. For an example, see C0D0S012 in Figure 52.

Using Page Segments

You can store a page segment in the DPF resource library only if the page segment is marked PUBLIC and is defined to PSF as a *hard page segment*.

Hard page segments are those named in a Map Page Segment (MPS) structured field of a MO:DCA-P page, in a page definition, or in an overlay. Hard page segments are loaded in the printer as separate objects at the start of either page processing or overlay processing, and they are retained until printing of the data set is finished. By contrast, *soft page segments* are segments that are not named in the MPS structured field. Soft page segments are sent to the printer each time they are invoked, as part of the page data. You cannot store these page segments in the DPF library.

In a line-format print data set, a page segment included by an IPS record is *hard* if it is defined in the page definition with a page segment list. If it is not defined in the page definition, the page segment is *soft* and cannot be stored in the DPF library.

If a page segment is called in an overlay, it is a hard page segment if it is named in the overlay resource MPS structured field. Most products that create overlays, such as OGL/370, automatically specify hard page segments. In addition, if an overlay that includes page segments is marked by the APSRMARK utility, PSF and DPF treats those page segments as hard page segments.[13]

If a page segment is called in a MO:DCA-P page, the page segment is hard or soft, depending on whether it is named in the MPS record of the Active Environment Group for the page. For MO:DCA-P pages created by DCF, an option to name or "map" the page segment is provided on the segment-include command. When PSF processes an MPS structured field, any page segment specified in that MPS is treated as a hard page segment by PSF and DPF. Similarly, when an overlay that includes page segments is marked by the APSRMARK utility, PSF and DPF treat those page segments as hard page segments.[13]

Using DPF Fonts and Printer-Resident Fonts with APSRMARK

In a PSF font library, you can define fonts for printers equipped with DPF and for printers equipped with resident fonts. You cannot, however, define outline fonts for use with DPF, because DPF does not support outline fonts. Also, a single PSF font library can be shared for printers equipped with DPF and for printers with resident fonts.

Using the APSRMARK utility, you can mark a font so that it is identified to PSF for use with:

- DPF only
- DPF and printer-resident fonts
- Neither of the above (the font is considered PRIVATE)

Marking a font for use with only DPF requires the following APSRMARK SYSIN control-statement parameters:

- INDD
- OUTDD
- PUBLIC
- MEMBER

Marking a font for use with DPF and printer-resident fonts requires the following APSRMARK SYSIN control-statement parameters:

- INDD
- OUTDD
- PUBLIC
- MEMBER
- RRID
- RRTIME
- RRDATE

[13] APSRMARK adds an appropriate MPS structured field for each soft page segment in an overlay that is marked for use with DPF.

Changing How Fonts Are Marked

You may want to use the APSRMARK utility to change marks for:

- Fonts already marked for use by DPF only
- Fonts already marked for DPF use and for printer-resident use
- Fonts already marked PRIVATE

Fonts Already Marked for Use by DPF Only

If the font is already marked for use only by DPF, and you want to also mark it for printer-resident use, follow the procedures described in Chapter 13, “Supporting Resident Fonts.” If you want to mark the font PRIVATE, follow the procedures described in this chapter and use the PRIVATE keyword in the APSRMARK SYSIN control statement.

Fonts Already Marked for DPF and Printer-Resident Use

If you want to mark a font for use by DPF only, follow the procedures described in Chapter 13, “Supporting Resident Fonts,” and use a value of 00000000 in the RRID keyword instead of the values given in *AFP: Printer Information*. If you want to mark the font PRIVATE, follow the procedures described in Chapter 13, “Supporting Resident Fonts,” and use the PRIVATE keyword in the APSRMARK SYSIN control statement.

Fonts Already Marked PRIVATE

If you want to mark a font for use by DPF and for use as a printer-resident font, and the font is already marked PRIVATE (or it is not marked at all), follow the procedures described in Chapter 13, “Supporting Resident Fonts.” If you want to mark the font for only DPF, follow the procedures described earlier in this chapter.

Chapter 17. Using Installation Exits

This chapter describes how to use the installation exits supplied with PSF. Some of the information in this chapter has to do with product-sensitive programming interfaces.

PSF provides installation exits for your use in coding and installing modifications to PSF functions. For example, with these exits, you can:

- Create your own separator pages, or modify separator-page formats supplied by PSF
- Modify, add, or suppress output records
- Modify system management facilities (SMF) type 6 records
- Inspect, redirect, or suppress PSF messages
- Manage resources

Note: PSF supports one exit, APSUX16, for direct-printing mode, and one exit, APSUX15, for MVS Download. The remaining installation exits are for use when PSF is running in deferred-printing mode.

Do's and Don'ts

Execution of the code in exit routines is an integral part of PSF and can affect performance. Keep the following points in mind when using exit routines:

- Exit routines should be coded by experienced programmers who are aware of overall system implications.
- Exit control blocks are provided in assembler language.
- Use only the common control blocks passed to this exit. For a description of the common control blocks, see “Common Control Blocks” on page 166.
- PSF runs above the 16-megabyte line. Exits can run in 24-bit or 31-bit addressing mode. However, 31-bit addressing is recommended, so that more functional subsystem applications (FSAs) can run in an address space. If any PSF exit is AMODE 24, PSF and JES allocate their control blocks, which are passed to the PSF exit below the 16-megabyte line.
- The exit code should reference a field in a common control block only by the name associated with that field in the DSECT control block. Do not use absolute displacements; they can change from one release of PSF to another.
- Do not write exit code that issues an explicit or implied WAIT, because it puts the entire PSF FSA into a WAIT state.
- Default exits APSUX01 and APSUX02 use the non-expanded area of the ECAWKBUF area in APSUECA. If you need to pass data between exits, extend the ECAWKBUF area. (For more information, see “Exit Table, APSUCOM” on page 179.) PSF and the DEFAULT exits do not use the expanded area. Because all exits have access to the same ECAWKBUF area, it is possible for one exit to overwrite an area used by another exit. You must be careful not to let this happen.
- If certain exit-routine functions are wanted for a subset of PSF-controlled printers, consider running your printers in multiple PSF address spaces and

using a STEPLIB statement to invoke selected exits only in those address spaces in which they are needed. The data sets on the STEPLIB statement must be APF-authorized. (The STEPLIB statement is highlighted in the example in Figure 53.)

If the data set containing the exit is in the MVS LINKLIST, the exit is available to all PSF address spaces. If the exit is to be used by a subset of PSF address spaces:

– Specify the data set containing the exit on the STEPLIB statement.

– Make sure that the data set containing the exit is not in the MVS LINKLIST.

```
//APSWPRO1 PROC
//STEP01 EXEC PGM=APSPPIEP,REGION=2048K
//STEPLIB DD DSN=SYS1.PSF.EXITLIB,DISP=SHR
//JOBHDR OUTPUT PAGEDEF=06080,FORMDEF=0101
//JOBTLR OUTPUT PAGEDEF=06080,FORMDEF=0101
//DSHDR OUTPUT PAGEDEF=06080,FORMDEF=0101
//MSGDS OUTPUT PAGEDEF=06080,FORMDEF=0101
//FONT01 DD  DSN=SYS1.FONTLIB,DISP=SHR
//PSEG01 DD  DSN=SYS1.PSEGLIB,DISP=SHR
//OLAY01 DD  DSN=SYS1.OVERLIB,DISP=SHR
//PDEF01 DD  DSN=SYS1.PDEFLIB,DISP=SHR
//FDEF01 DD  DSN=SYS1.FDEFLIB,DISP=SHR
//PRINTR1 CNTL
//PRINTR1 PRINTDEV  FONTDD=*.FONT01,
//    OVLYDD=*.OLAY01,
//    PSEGDD=*.PSEG01,
//    PDEFDD=*.PDEF01,
//    FDEFDD=*.FDEF01,
//    JOBHDR=*.JOBHDR,
//    JOBTRLR=*.JOBTLR,
//    DSHDR=*.DSHDR,
//    MESSAGE=*.MSGDS,
//    SETUP=FORMS,
//    BUFNO=5,
//    PAGEDEF=06080,
//    FORMDEF=0101,
//    CHARS=(GF10,GS10,TU10,GU10),
//    PIMSG=YES,
//    TRACE=NO,
//    DATACK=BLOCK
//PRINTR1 ENDCNTL
```

Figure 53. Use of STEPLIB Statement for Exit Routine in PSF Startup Procedure

Changing Exit Routines

During PSF installation, the PSF distribution tape is loaded into SYS1.SAMPLIB. PSF provides source code (in assembler language) for all the exit routines on the distribution tape. Each routine, then, is a member of SYS1.SAMPLIB. You can either view the source routines online or print them. You can then modify, assemble, and install (or link-edit with the replace option) any of these routines to provide your own version of the PSF exits. Table 14 lists these routines and summarizes their functions.

Table 14. PSF Exit Routines in SYS1.SAMPLIB

Sample Routine	Description
APSUX01.ASSEMBLE	Source routine for default job headers. See Figure 30 on page 82 for an example.
APSUX01P.ASSEMBLE	Source routine for job headers that print faster on communication-attached printers, except the 4224 and 4234 printers. See Figure 31 on page 83 for an example.
APSUX01S.ASSEMBLE	Source routine for job headers that use guaranteed print labeling (described in *Print Services Facility: Security Guide*).
APSUX02.ASSEMBLE	Source routine for default job trailers.
APSUX02P.ASSEMBLE	Source routine for job trailers that print faster on communication-attached printers, with the exception of the 4224 and 4234 printers.
APSUX02S.ASSEMBLE	Source routine for job trailers that use guaranteed print labeling (described in *Print Services Facility: Security Guide*).
APSUX03.ASSEMBLE	Source routine for default data set separators.
APSUX04X.ASSEMBLE	Sample (skeleton) source routine for processing line-data records.
APSUX04Y.ASSEMBLE	Sample (skeleton) source routine for processing structured fields.
APSUX05X.ASSEMBLE	Sample (skeleton) source routine for modifying SMF type 6 record processing.
APSUX06X.ASSEMBLE	Sample (skeleton) source routine for modifying PSF message processing.
APSUX07S.ASSEMBLE	Sample source routine for loading all page definitions and form definitions from DASD with each use.
APSUX07T.ASSEMBLE	Sample source routine for mapping a FORMS= value to a PAGEDEF and FORMDEF pair.
APSUX07U.ASSEMBLE	Sample source routine for altering the member-name prefix of resource objects (permitting use of objects with “nonstandard prefixes” for production or secure work).
APSUX07X.ASSEMBLE	Sample (skeleton) source routine explaining exit structure.
APSUX07Y.ASSEMBLE	Sample (skeleton) source routine demonstrating: • Exit involvement, based on printer name and type • Substitution of a page definition or a page segment • Issuance of a message for page segment substitutions • Retention of an overlay over data set boundaries.
APSUX07Z.ASSEMBLE	Sample (skeleton) source routine demonstrating: • Exit involvement based on printer name and type • Substitution of a job-header form definition.
APSUX15X.ASSEMBLE	Sample source routine to provide the printer device name and the output group name for MVS Download to transmit to its server on AIX.
APSUX16X.ASSEMBLE	Sample (skeleton) source routine for the direct printing exit.

Note: PSF provides object code for the default separator-page exits (modules APSUX01, APSUX02, and APSUX03). PSF does not provide object code for the other exits—logical-record processing, SMF type 6 record processing, PSF-message processing, resource processing, MVS download, and direct printing.

During PSF initialization, the exit subcomponent issues a BLDL request to locate any user-provided exit routines prefixed with APSUX*nn* (where *nn* is replaced with the exit numbers: 01, 02, 03, 04, 05, 06, 07, 15, 16). The exit routines may be in a data set pointed to in a STEPLIB DD statement in the PSF startup procedure, or they may be in a TASKLIB, STEPLIB, or JOBLIB concatenated with SYS1.LINKLIB. You can provide a different set of exit routines for each PSF address space in your system by using a unique STEPLIB for each PSF startup procedure, or you can provide a single set of exit routines for all PSF-controlled printers in the operating system by placing them in a LINKLIST data set.

Notes:

1. All exit routines must be re-enterable and must be link-edited with the RENT,LET attributes.
2. Exit routines must reside in an authorized library.
3. Code for installation exits can be written in 24-bit or 31-bit addressing mode.
4. PSF-program maintenance may require the reassembly of installation exits.
5. All exit routines run in supervisor state with a protection key of 1. All system services available in supervisor state are available with these exits.
6. The module name (in printable EBCDIC) should be placed in the object code, starting at the sixth byte. If a dump is run, the module name is printed in the dump. (ESTAE is in effect during the execution of exits.)

Common Control Blocks

When PSF calls an exit routine, register 1 points to a 1-word parameter list that contains the address of the APSGEXTP area. This area contains the address of the exit-communication area (APSUECA), pointers to JSPA, and pointers to the input records associated with exits APSUX04, APSUX06, APSUX07, and APSUX16. When the exit routine returns to PSF, PSF does not check register 15 for exits.

APSGEXTP Area

The format of APSGEXTP is shown in Figure 54 on page 167. Values in parentheses are in hexadecimal.

```
Offsets   Type     Length  Name Description

 0    (0) Structure    40  APSGEXTP       PSF-exit-parameter area

======================================================================
The following fields provide input to the installation exit processor
routines.
======================================================================
 0    (0) Address       4  XTPECAP        Exit-communication-area address.
 4    (4) Address       4  XTPJSPAP       Pointer to JES separator page
                                          area of the last data set
                                          (with a separator,
                                          might be zero).
 8    (8) Address       4  XTPRECP        Pointer to input record.
12    (C) Signed        4  XTPRECL        Input-record length.
16   (10) Character     4  *              Reserved.
======================================================================
The following fields are set by the installation exit processor
routines to provide an interface back to PSF.  On input to processors,
XTPRIXP points to a record index entry.  On output, if the processor
indicates “write index,” XTPRIXP must point to the newly created
index header.
======================================================================


20   (14) Address       4  XTPRIXP        Pointer to record table index.
24   (18) Unsigned      1  XTPPIND        Processing indicator.
25   (19) Bitstring     1  XTP4FLAG       APSUX04 end-of-transmission
                                          (EOT) flags.
      1...  ....           XTP4NTY        PSF will send EOT
                                          indicator to APSUX04.
      .1..  ....           XTP4NEOT       Normal EOT has occurred.
      ..1.  ....           XTP4ABNE       Abnormal EOT has occurred.
      ...1  ....           XTP4NCAL       Call APSUX04 one time.
                                          When this flag is set ON,
                                          Exit 04 will not be
                                          called again until EOT or
                                          the next copy of the data
                                          set or the next data set.
                                          This flag is reset when
                                          Exit 04 is called for the
                                          first record of the next
                                          copy of the data set or
                                          the next data set.
      ....  1...           XTP4CONV       Do line-mode conversion.
                                          This flag is reset when
                                          Exit 04 is called for the first
                                          record of the transmission.
      ....  .111           *              Reserved.
26   (1A) Character     4  *              Reserved.
30   (1E) Bitstring     1  XTP5FLAG       APSUX05 SMF 6 flags.
      1...  ....           XTP5NEWS       Processing JESNEWS data set.
31   (1F) Unsigned      1  *              Reserved.
32   (20) Unsigned      2  XTP4XOFF       Set horizontal offset-
                                          pels at 240 pels/inch.
                                          Valid only when
                                          XTP4CONV is set ON.
34   (22) Unsigned      2  XTP4YOFF       Set vertical offset-
                                          pels at 240 pels/inch.
                                          Valid only when
                                          XTP4CONV is set ON.
36   (24) Character     4  *              Reserved.
```

Figure 54 (Part 1 of 2). Format of the APSGEXTP Area

```
======================================================================
The following fields provide input to the installation exit 06
(message exit)
======================================================================

 0    (0) Character     8  XTP6PROC       FSA identifier:
                                           channel-attached printers:
                                             DDnnn, where nnn is the device-address
                                           non channel-attached printers:
                                             label of PRINTDEV statement
 8    (8) Character     7  XTP6MSG        Message ID (APSXXXI).
15    (F) Character     1  XTP6TYPE       Message TYPE.
                                          W=WTO:  Most WTO messages
                                          are asynchronous.  Async
                                          messages belong to the data
                                          set on the printer, which may
                                          be different from the one
                                          being processed by PSF.
                                          XTPJSPAP points to the JSPA
                                          related to the data set
                                          currently being processed
                                          by PSF.
16   (10) Character     1  XTP6FLG1       APSUX06 processing indicators
     .....  ....           XTP6TFSA       Terminate the FSA. PSF will
                                          complete the processing of the
                                          current data set (the one
                                          being processed by PSF, which
                                          may be different from the one
                                          on the printer), and terminate
                                          the FSA.
      .1..  ....           XTP6HOLD       Release the data set currently
                                          being processed by PSF to JES,
                                          and mark the data set as
                                          unprintable.
      ..11  1111           *              Reserved.
17   (11) Bitstring     1  XTP6ACT        Action to be taken by PSF:
                                          Set by PSF on input;
                                          can be altered by the exit.
                                          If XTP6ACT is set to 0 by the
                                          exit, the message is
                                          suppressed.
      1...  ....           XTP6WTO        Put msg to op. console.
      .1..  ....           XTP6SEC        Put msg to sec. console.
      ..1.  ....           XTP6PAP        Put msg to paper (msg data set).
      ...1  1111           *              Reserved.
18   (12) Character     6  *              Reserved.
```

Figure 54 (Part 2 of 2). Format of the APSGEXTP Area

For information on the contents of XTP7, see Figure 69 on page 280.

Exit-Communication Areas: APSUECA and APSUECE

APSGEXTP provides addressability to the exit-communication area (APSUECA, or simply ECA) and to its extension (APSUECE). These areas provide the interface among the exit routines.

The format of APSUECA is shown in Figure 55. Values enclosed in parentheses are in hexadecimal.

Offsets		**Type**	**Length**	**Name**	**Description**
0	(0)	Structure	*	APSUECA	Exit-Communication Area.
0	(0)	Character	*	ECABASE	Base section of ECA.
0	(0)	Character	4	ECAID	Control-block ID set to the character string UECA.
4	(4)	Signed	4	ECALEN	Length of exit-communication area.
8	(8)	Character	6	ECAPTF	Relative PTF date.
14	(E)	Character	2	ECAREL	Release level.
16	(10)	Signed	4	ECARCODE	RRB PUT return code.
20	(14)	Address	4	ECAUCOMP	Address of exit APSUCOM table.
24	(18)	Address	4	ECAPUTP	Address of exit PUT routine.
28	(1C)	Address	4	ECABLKP	Address of exit block-letter routine.
32	(20)	Address	4	ECASABP	Address of APSPFSAB.
36	(24)	Address	4	ECAECEP	Address of exit-communication extension.
40	(28)	Address	4	ECAMARP	3800 -3/8 migration aid pointer.
44	(2C)	Character	4	ECAFORM	Form name in output JCL (FORMS=).
48	(30)	Bitstring	1	ECAMFCS	Mark form carrier strip flag.
49	(31)	Character	6	ECARNDM	Separator page random number.
55	(37)	Character	1	ECAFLAG	ECA flags.
	1...			ECAMFTRL	Mark forms inhibit flag for job trailer. 0 = job trailers may be marked. 1 = do not mark job trailers.
	.1..			ECALSTDS	Last or only data set. 0 = Not the last or only data set of the job. 1 = Last or only data set of job.
	..11 1111			*	Reserved.
56	(38)	Address	4	ECARECAD	Address of a record to be put to PSF from an exit processor.
60	(3C)	Signed	4	ECARECLN	Length of exit record.
64	(40)	Bitstring	1	ECADRF	Record description flags.
	1...			ECADSR	Indicates data-stream record.
	.1..			ECALMR	Indicates line-mode record.
	..1.			ECAANSI	Record contains ANSI control characters.
	...1			ECAMACH	Record contains machine control characters.
	.... 1...			*	Reserved (PSF use only).
	.... .1..			*	Reserved (PSF use only).
	.... ..1.			*	Reserved (PSF use only).
	1			ECAOPJ	OPTCD=J used for record.

Figure 55 (Part 1 of 2). Format of Exit-Communication Area, APSUECA

```
  65   (41) Bitstring     1  *            Reserved.
  66   (42) Signed        2  ECATRNC      Transmission count.
  68   (44) Bitstring     1  ECAFLAGS     Exit-control flags.
        1...  ....           ECASLANT     Slant letters requested.
        .1..  ....           ECALJUST     Left justify block letter
                                          lines.
        ..1.  ....           ECAPBLK      Performance-block letters.
        ...1  1111           *            Reserved.
  69   (45) Bitstring     1  *            Reserved.
  70   (46) Signed        2  ECAGWRK      General-use work field.
  72   (48) Character     8  ECABLKIN     Input to block letter routine.
  80   (50) Character    72  ECAUSAVE     Exit processor's save area.
 152   (98) Character    72  ECAGSAVE     General-use exit save area.
 224   (E0) Character    72  ECABSAVE     24-bit interface return save
                                          area.
 296  (128) Character    72  ECAMSGSV     APSUX06 (message) exit save
                                          area.
 368  (170) Character    72  ECARSCSV     APSUX07 (resource) exit save
                                          area.
 440  (1B8) Character     *  ECAWKBUF     The work buffer (ECAWKBUF)
                                          length depends on a field in
                                          APSUCOM. Minimum length of
                                          the ECA, including the work
                                          buffer, is 2039 bytes.The
                                          The minimum length of the
                                          work buffer is 1599 bytes.
```

Figure 55 (Part 2 of 2). Format of Exit-Communication Area, APSUECA

JES2 4.1.0 and JES3 4.2.0 and above supply the job boundaries to PSF. Earlier levels of JES request a job header for the first data set of a job, if job separators have been specified.

If your JES level is earlier than JES2 4.1.0 or JES3 4.2.0, ECALSTDS is valid only if you specify job separator pages, using the following JES parameters:

- Under JES2, the SEP=YES parameter specifies that job-header and job-trailer pages are to be printed.
- Under JES3, the HEADER=YES parameter specifies that job-header pages are to be printed, and the BURST=YES parameter specifies that job-trailer pages are to be printed.

Note: If you do not want PSF to generate separator pages, delete or modify the separator-page exits. For information on how to do this, see “Programming with Separator-Page Exits” on page 175.

The format of APSUECE is shown in Figure 56 on page 171. Values enclosed in parentheses are in hexadecimal.

Offsets	Type	Length	Name	Description
0 (0)	Character	*	APSUECE	Exit extension.
0 (0)	Character	4	ECEID	Control Block ID set 'UECE'.
4 (4)	Signed	4	ECELEN	Length of ECE.
8 (8)	Character	6	ECEPTF	Relative PTF date.
14 (E)	Character	2	ECEREL	Release level.
16 (10)	Address	4	ECEFECAP	Addr of FSA ECA, used by exits 1-4, 6.
20 (14)	Address	4	ECENECAP	Addr of NST ECA, used by Exit 5.
24 (18)	Character	8	ECEPRMD	PRMODE defined in JCL output.
32 (20)	Character	4	ECEFLASH	Flash defined in JCL output.
36 (24)	Character	8	ECEDPDEF	PAGEDEF defined in PRINTDEV.
44 (2C)	Character	8	ECEDFDEF	FORMDEF defined in PRINTDEV.
52 (34)	Address	4	ECEFSSWK	FSS work area printer.
56 (38)	Character	8	ECEPDEVK	PRINTDEV SWB token.
64 (40)	Character	8	ECEOUTTK	Print data set SWB token.
72 (48)	Character	8	ECEJDVTN	Print data set JDVT name.
80 (50)	Character	2	ECEPRTT	Printer type (eg. X'3800') that the device is emulating (from the IPDS STM).
82 (52)	Character	1	ECEPRTM	Printer model number (e. g. X'03') that the device is emulating (from the IPDS STM).
83 (53)	Bitstring	1	ECEFLAG	Exit flags.
	1...		ECEJESS	JES supports Enhanced SYSOUT Support (ESS) JCL.
	.1..		ECERPM	RPM is attached.
	..1.		ECEDPF	DPF is attached.
	...1		ECEPSFD	PSF Direct is attached.
	 1111		*	Reserved
84 (54)	Character	8	ECEFORM	Form name in output JCL (FORMS=)
92 (5C)	Character	4	ECEPRTDT	Printer device type, stamped on the printer in EBCDIC (from the IPDS OPC, or IPDS STM if none in the OPC).
96 (60)	Character	3	ECEPRTMD	Printer model number, stamped on the printer in EBCDIC (from the IPDS OPC, or IPDS STM if not in the OPC).
99 (63)	Character	1	*	Reserved.
100 (64)	Address	4	ECEBTBL	Address of block letter tables. Do not modify ECEBTBL; it is used by APSUBLK.
104 (68)	Signed	4	ECEJOBLP	Logical pages in job.
108 (6C)	Signed	4	ECEJOBIM	Impressions (sides) in job.
112 (70)	Signed	4	ECEJOBPP	Physical pages in job.
116 (74)	Address	4	ECEWKPTR	Address of exit work area.
120 (78)	Signed	4	ECEWKLEN	Length of exit work area.
124 (7C)	Address	4	ECETUPTR	SWBTU address.
128 (80)	Signed	4	ECETULEN	SWBTU length.
132 (84)	Signed	4	ECEREC	Logical record count (does not include header or trailer page records).

Figure 56 (Part 1 of 2). Format of Exit-Communication Area Extension, APSUECE

136	(88)	Signed	4	ECELPCNT	Logical page count (includes header and trailer pages).
140	(8C)	Signed	4	ECEPPCNT	Physical page count (includes header and trailer pages).
144	(90)	Character	2	*	Reserved.
146	(92)	Signed	2	ECETXMS	Total transmission required.
152	(98)	Address	4	ECEX02P	Address of exit APSUX02.
156	(9C)	Address	4	ECEX03P	Address of exit APSUX03.
160	(A0)	Address	4	ECEX04P	Address of exit APSUX04.
164	(A4)	Address	4	ECEX05P	Address of exit APSUX05.
168	(A8)	Address	4	ECEX06P	Address of exit APSUX06.
172	(AC)	Address	4	ECEX07P	Address of exit APSUX07.
176	(B0)	Address	4	ECEX08P	Reserved.
180	(B4)	Address	4	ECEX09P	Reserved.
184	(B8)	Address	4	ECEX10P	Reserved.
188	(BC)	Address	4	ECEX11P	Reserved.
192	(C0)	Address	4	ECEX12P	Reserved.
196	(C4)	Address	4	ECEX13P	Reserved.
200	(C8)	Address	4	ECEX14P	Reserved.
204	(CC)	Address	4	ECEX15P	Address of exit APSUX15 (in MVS Download only).
208	(D0)	Address	4	ECEX16P	Address of exit APSUX16 (in direct print mode only).
212	(D4)	Signed	4	ECEIMPS	Impressions (sides) printed for the data set.
216	(D8)	Character	12	ECEDSID	FSI GETDS data set ID (in MVS Download only).
228	(E4)	Character	16	*	Reserved.

Figure 56 (Part 2 of 2). Format of Exit-Communication Area Extension, APSUECE

If repositioning is required, PSF can enter the exit routines for header pages (APSUX01 and APSUX03) more than once, but the header pages might not reprint unless the repositioning calls for the header pages to reprint. When PSF enters either exit APSUX01 or exit APSUX03, PSF sets ECATRNC to the current transmission count, which can be used to determine whether the exit has been reentered. An installation exit should not attempt to skip pages or duplicate pages if the exits are reentered; such an attempt might interfere when PSF is repositioning for error recovery.

ECEFSSWK Address of 32 bytes of functional subsystem (FSS) storage.

This address is passed to all exits in the same address space. When multiple functional subsystem applications (FSAs) are running in an FSS, the same address is passed to all the FSAs.

The exits can use this storage to communicate among different FSAs or to save data between FSA sessions. This storage is not freed until the FSS is terminated.

The FSS storage is for the exclusive use of the exits. PSF does not read or modify this storage.

The exit must serialize the use of the FSS storage, because more than one FSA subtask (or an FSA and the NST subtask) can be active at the same time in the FSS address space.

ECEPRTDT ECEPRTDT contains the printer device type, as provided in the IPDS Obtain Printer Characteristics Product Identifier Self-Defining Field; that field is in the XOA-OPC command. If the printer does not contain a device type in that field, ECEPRTDT matches ECEPRTT, and ECEPRTMD matches ECEPRTM.

ECEPRTMD ECEPRTMD contains the printer model, as provided in the IPDS Obtain Printer Characteristics Product Identifier Self-Defining Field. If the printer does not contain a model in that field, ECEPRTDT matches ECEPRTT, and ECEPRTMD matches ECEPRTM.

ECEPRTT ECEPRTT contains the printer device type that the printer is functionally emulating, as provided in the IPDS Sense Type and Model command. The functional emulation extends only as far as IPDS deviations and PSF software dependencies for the support of certain functions. Not all of the functions supported are identical.

ECEPRTM ECEPRTM contains the printer model that the printer is functionally emulating, as provided in the IPDS Sense Type & Model command. The functional emulation extends only as far as IPDS deviations and PSF software dependencies for the support of certain functions. Not all of the functions supported are identical.

ECEOUTTK The scheduler work block (SWB) token for the OUTPUT JCL statement of the print data set.

PSF uses ECEOUTTK to retrieve the keywords specified in the OUTPUT JCL statement of the print data set.

JES passes this token to PSF in the functional subsystem interface (FSA) GETDS parameter list (IAZFSIP).

For more information on SWBs and how to use the SWB token to retrieve keywords, refer to the *Using the Functional Subsystem Interface* for your operating system, listed in Appendix K, “Related Publications.”

Note: The SWB contains JCL values merged from the DD statement, the OUTPUT statement, and the JES defaults according to the resource hierarchy. JES merges the values before allocating the print data set to PSF. When the JCL values are returned, therefore, there is no way to determine whether a particular value was specified on the print data set or defaulted to by JES.

ECETUPTR The address of the SWBTU for the print data set's OUTPUT JCL statement.

With MVS 4.1.0 or higher, you can use the SWBTUREQ macro to obtain the JCL parameters specified for the print data set. The SWBTU address is input to the SWBTUREQ macro. ECETULEN contains the SWBTU length.

The separator exits supplied by PSF contain an example of how to invoke the SWBTUREQ macro.

For more information about the SWBTUREQ macro, refer to the *Authorized Assembler Services Reference* for your operating system, listed in Appendix K, “Related Publications.”

Note: The SWBTU contains JCL values merged from the DD statement, the OUTPUT statement, and the JES defaults according to the resource hierarchy. JES merges these values before allocating the print data set to PSF. When the JCL values are returned, therefore, there is no way to determine whether a particular value was specified on the print data set or defaulted to by JES.

ECEWKPTR The address of a local work area for the PSF exits.

ECEWKPTR can be used as a local work area by installation-written exits. The work area length is 1024 bytes.

The separator exits supplied by PSF exits use this area. If those exits are used, therefore, this area should not be used to retain data across exit invocations.

Sequence of Exit Invocation

Following is a typical flow of exit control in the sequence of exit interpretation:

1. APSUX01: Job-header page
2. APSUX03: Data-set separator page
3. APSUX04: Examine first spool record
4. APSUX04: Examine next spool record (until end of data set)
5. APSUX04: End-of-data-set call
6. APSUX02: Job-trailer page
7. APSUX05: Write SMF type 6 record

Note: After the trailer page prints, calls to APSUX05 are asynchronous. Calls to APSUX06 may be asynchronous also.

Control Flow with APSUX07

Following is a typical flow of exit control that includes the use of APSUX07 for resource management.

1. APSUX07: Begin-data-set call
2. APSUX01: Job-header page
3. APSUX07: Access and load calls for resources used on the job-header page
4. APSUX03: Data-set separator page
5. APSUX07: Access and load calls for resources used on the data set separator page
6. APSUX04: Examine first spool record
7. APSUX07: Resource-access call for page definition
8. APSUX07: Resource-access call for form definition
9. APSUX07: Access and load calls for resources identified in the page definition.
10. APSUX07: Access and load calls for resources identified in the form definition.
11. APSUX04: Examine next spool record (until end of data set)
12. APSUX07: Access and load calls for resources identified in the input data stream
13. APSUX04: End-of-data-set call
14. APSUX02: Job-trailer page
15. APSUX07: Access and load calls for resources used on the job-trailer page
16. APSUX07: Resource-delete calls

17. APSUX05: Write SMF type 6 record

Note: After the trailer page prints, calls to APSUX05 are asynchronous. Calls to APSUX06 may be asynchronous, also.

Separator-Page Exits: APSUX01, APSUX02, and APSUX03

PSF provides modules that create default separator pages of three different types: job headers, job trailers, and data set headers. As is shown in Figure 30 on page 82, Figure 32 on page 85, and Figure 33 on page 86, these separator pages are similar in format to the JES separator pages. Also, if data set printing is interrupted and then resumes, PSF provides a job-continuation page to show where printing continues. The formats of the job-header page and the job-continuation page are identical, except that the continuation page prints “CONT” at the bottom (instead of “START”).

Three exits are available for creating separator pages:

APSUX01	Job-header pages (including job-continuation pages)
APSUX02	Job-trailer pages
APSUX03	Data-set-header pages

PSF also provides assembler-language source routines in SYS1.SAMPLIB that you can modify, assemble, and install (or link-edit with the replace option) to create your own separator-page exit modules. These routines, listed in Table 14 on page 165, run in the same address space as PSF.

Input for Separator-Page Exits

When a separator-page exit is entered, it contains the following:

Register 1	Pointer to the address of parameter area APSGEXTP
Register 13	Pointer to caller's save area
Register 14	Return address
Register 15	Entry-point address

Output for Separator-Page Exits

When control returns from the exit routine to PSF, the contents of the registers should be the same as they were at entry to the exit routine. As is explained in “Service Routines” on page 181, if the exit-PUT service routine (APSUPUT) recognizes an error condition during record processing, the ECA return code (ECARCODE) is set. If this field is not zero, PSF issues message APS950I.

Processing Multiple-Page Job Trailers with Forms Marking

For details about processing multiple-page job trailers with forms marking, see “Forms Marking of Multiple-Page Job Trailers” on page 90.

Programming with Separator-Page Exits

The following considerations apply to the separator-page exits:

- When a separator-page exit routine is entered, register 1 points to a one-word parameter list with the address of the APSGEXTP area. APSGEXTP contains pointers to the exit-communication area (APSUECA) and to the JES separator-page area (IAZJSPA).

APSUECA contains pointers to:

- APSUCOM, the exit-communication table
- APSUBLK, the block-letter service routine
- APSUPUT, the exit-put service routine
- APSUECE, the exit-communication area extension

IAZJSPA provides information to produce JES separator pages—job headers, job trailers, and data set headers. You can use IAZJSPA to duplicate this information for your own separator-page exit routines. Figure 57 on page 178 shows the format of IAZJSPA.

- When a separator-page exit routine is entered, register 13 points to an 18-word save area that can save the caller's registers. The APSUECA control block contains a save area, ECAUSAVE, for the exit routine.
- Use register 12 as a base register for exit routines. If register 12 points to the beginning of an exit routine, and the module name begins in the sixth byte of the routine, the module name appears in the SDUMP title page if an abend occurs.
- Service routines APSUBLK and APSUPUT require that register 5 contain the address of APSUECA. For details on the interface to these routines, see “Service Routines” on page 181.
- Before calling APSUBLK, set the ECAPBLK field of APSUECA if you want to generate an optional (faster-running) separator page. For more information, see “Block-Letter Routine, APSUBLK” on page 181.

  If you use an optional separator page, the job-separator-page OUTPUT statement in the PSF startup procedure should specify a page definition having a page size no larger than 7⅛ × 7 inches and a font no larger than 12 pitch.[14]

  Note: A 12-pitch font produces text that is centered on the separator page.

- When your printer is attached to the Distributed Print Function (DPF) of PSF/2, and your JES level is earlier than JES2 4.1.0 or JES3 4.2.0, you must enable job headers and job trailers, because headers and trailers cause JES to tell PSF where the job boundaries are. You do *not*, however, have to transmit the data to create the job headers or trailers. To suppress the pages, you can delete or modify the separator exits. If some of the PSF-controlled printers on the system are not attached through DPF, you can add code to the separator exits. Place the following in the PROLOG section of the code, after the “SAVE CALLERS SAVE AREA ADDR”:

```
USING     APSUECE,@02           Get addressability
L         @02,ECAECEP             to ECE
TM        ECEFLAG,ECEDPF        DPF-attached printer?
BO        OUT                   Yes--exit immediately
DROP      @02                   Drop address to ECE
```

- As is explained in “Exit Table, APSUCOM” on page 179, the APSXUCOM macro provides a table for converting the Julian date to a year/month/day format (for example, 87 Jun 03).

[14] Pitch refers to character width and is based on the number of characters that can be placed in a linear inch. For example, a 12-pitch font prints 12 characters per inch.

- As is explained in “Block Letter Tables, APSUBTBL” on page 180, APSUBTBL provides the translate tables for the block letters created by APSUBLK for the separator pages.
- The separator-page lines can be either line-format records or structured fields. These lines can be composed of information obtained from the following control-block fields:
  - JSPAJBNM for job name
  - JSPAJBID for job number
  - JSPJSOCL for SYSOUT class
  - JSPJGRP1 for output-group identifier 1
  - JSPJGRP2 for output-group identifier 2
  - JSPJGRPN for output-group name
  - JSPJGRPD for output-group destination name
  - JSPJPNAM for programmer name
  - JSPJRMNO for room number
  - JSPADEVN for device name
  - JMRCPUID for system name

  You can also use the following additional informational fields for the data-set-header exit routine:

  - JSPJDSSN for step name
  - JSPJDSDD for data set DD name
  - JSPJSOCL for SYSOUT class
  - JSPJPRIO for data set priority
  - JSPJDSPN for procedure name

JES Job-Separator-Page Areas—IAZJSPA and IAZJSEP

IAZJSPA contains job-level information for a data set that generates JES separator pages—job headers, job trailers, and data set headers. You can use IAZJSPA to duplicate this information for your own separator-page exit routines. If the separator-page exit requires information that is not provided in IAZJSPA, the information may be available in a JES exit.

A pointer to the modified IAZJSPA is passed to the PSF separator-page exit. To modify IAZJSP in JES2, use exit 23; in JES3, use exit IATUX45.

Figure 57 on page 178 shows the format of IAZJSPA. Values inside parentheses are in hexadecimal.

Offsets	Length	Name	Description
0 (0)	4	JSPAID	Control block identifier set to JSPA
4 (4)	2	JSPALEN	Length of JSPA
6 (6)	1	JSPAFLG1	Flags
7 (7)	1	*	Reserved
8 (8)	8	JSPAJBNM	Job name
16 (10)	8	JSPAJBID	Job identifier
24 (18)	8	JSPADEVN	Device name
32 (20)	4	JSPADEVA	Device address
36 (24)	4	JSPAJMR	JMR address
40 (28)	8	JSPJGRPN	Output-group name
48 (30)	2	JSPJGRP1	Output-group identifier 1
50 (32)	2	JSPJGRP2	Output-group identifier 2
52 (34)	8	JSPJGRPD	Output-group destination name
60 (38)	4	JSPJRMNO	Room routing number
64 (40)	20	JSPJPNAM	Programmer's name
84 (54)	24	JSPJDSNM	Data set name, fully qualified
	8	JSPJDSPN	Data set procedure name
	8	JSPJDSSN	Data set step name
	8	JSPJDSDD	Data set DD name
108 (6C)	1	JSPJSOCL	SYSOUT class
109 (6D)	1	JSPJPRIO	Data set priority
110 (6E)	4	JSPAUSR1	Reserved for user
114 (72)	4	JSPAUSR2	Reserved for user

Figure 57. Format of JES Job-Separator-Page Area, IAZJSPA.

JES determines the fields, starting with JSPJGRPN at offset 40 (X'28'). The JSPA is mapped by the IAZJSPA macro. For more information, refer to your JES publications.

A job-separator-page area extension, IAZJSEP, is available with either JES2 3.1.3 or JES3 3.1.3. Figure 58 shows the format of IAZJSEP. Values inside parentheses are in hexadecimal.

Offsets	Length	Name	Description
0 (0)	2	JSPEXNUM	Number of extensions
2 (2)	2	JSPEXLEN	Length of extensions
4 (4)	16	*	Reserved
20 (14)	2	JSPCELEN	Length of common extension
22 (16)	2	JSPCEVSN	Version number
24 (18)	4	JSPCECID	Common extension identifier
28 (1C)	8	JSPCEUID	User ID
36 (24)	8	JSPCESEC	Security label
44 (2C)	53	JSPCEDSN	Data set resource name
97 (61)	3	*	Reserved
100 (64)	4	JSPCESEG	Segment ID

Figure 58. Format of JES Job-Separator-Page Area Extension, IAZJSEP

Exit Table, APSUCOM

The exit table, APSUCOM, contains:

- A 4-byte length field
- A table for converting the Julian date to a year/month/day format

The length field represents the length of the exit-communication area, APSUECA. It is built once during the initialization of PSF. Increasing the value given for this length field increases the work buffer at the end of APSUECA. The work buffer can store information that is passed from exit to exit for use by several PSF installation-exit routines.

The APSXUCOM macro can be used to create a new APSUCOM exit table. Not only can you change the length field, but you can also set input parameters to change the 3-character month names in the Julian conversion table.

Note: As is explained in “Block-Letter Routine, APSUBLK” on page 181, APSUBLK passes block letters to the exit in the work buffer, ECAWKBUF. To prevent APSUBLK from overwriting data, expand the buffer and pass the data in the expanded portion of ECAWKBUF.

To create a new APSUCOM exit table:

1. Code the APSXUCOM macro.

   The variable names for the months of the year are alphanumeric, with a maximum length of 3 characters. With default values underscored, the syntax for the APSXUCOM macro is as follows:

   [SYMBOL] APSXUCOM [BUFF=*nnnnnnn*|1599],
   [JAN= {*ccc*|JAN}], [FEB= {*ccc*|FEB}], [MAR= {*ccc*|MAR}],
   [APR= {*ccc*|APR}], [MAY= {*ccc*|MAY}], [JUN= {*ccc*|JUN}],
   [JUL= {*ccc*|JUL}], [AUG= {*ccc*|AUG}], [SEP= {*ccc*|SEP}],
   [OCT= {*ccc*|OCT}], [NOV= {*ccc*|NOV}], [DEC= {*ccc*|DEC}]

   BUFF= {*nnnnnnn*|1599}
   The size of the work buffer, ECAWKBUF, in the exit-communications area, APSUECA. *nnnnnnn* is a decimal number between 1599 and 1048576. (For more information on APSUECA, see Figure 55 on page 169.)

   JAN= {ccc|JAN}
   The variable name for January

   FEB= {ccc|FEB}
   The variable name for February

   MAR= {ccc|MAR}
   The variable name for March

   APR= {ccc|APR}
   The variable name for April

   MAY= {ccc|MAY}
   The variable name for May

   JUN= {ccc|JUN}
   The variable name for June

JUL= {ccc|JUL}
The variable name for July

AUG= {ccc|AUG}
The variable name for August

SEP= {ccc|SEP}
The variable name for September

OCT= {ccc|OCT}
The variable name for October

NOV= {ccc|NOV}
The variable name for November

DEC= {ccc|DEC}
The variable name for December

2. Assemble the following module:

   ```
            START  0
   APSUCOM  CSECT
            APSXUCOM BUFF=nnnnnnn,.....
            END    APSUCOM
   ```

   PSF requires 1599 bytes of buffer space; so increase the value *nnnnnnn* by the additional space you need. For example, if you need 100 bytes, code BUFF=1699.

3. Link-edit APSUCOM either into the library that contains the PSF exit routines that will use this module or into any authorized LINKLIST library.

Upon the next initialization of the functional subsystem, FSS, the new APSUCOM exit table takes effect.

Block Letter Tables, APSUBTBL

The block letter tables, APSUBTBL, consist of:

- A translate table
- An index table
- A performance block letter table
- A default block letter table

You can use APSUBTBL to create new block letters, add new block letters, and change existing block letters. The default tables produce block characters from the following input:

- Uppercase letters, A–Z
- Symbols, $, #, and @
- Numerals, 0–9
- Lowercase letters, a–z, which are translated to uppercase letters for printing
- Superscript EBCDIC numerals, which are translated to numerals (0–9) for printing
- All other characters, which are translated to blanks

PSF provides the assembler language source code in SYS1.SAMPLIB for the block letter tables, APSUBTBL. You can modify, assemble, and install, or link-edit with the *replace* option, APSUBTBL to create your own block letter tables.

If installation separator page exit APSUX01, APSUX02, or APSUX03 is installed, APSUBTBL must also be installed. If APSUBTBL is not installed, PSF issues message APS959I, and processing continues without the separator page exits.

Service Routines

PSF provides two service routines: a block-letter routine, APSUBLK and an exit-put routine, APSUPUT. The exit-communication area, APSUECA, provides input to these service routines.

Block-Letter Routine, APSUBLK

The block-letter service routine, APSUBLK, builds block letters for separator pages. Block letters can be straight or slanted, and centered or left-justified. They are 12 characters high (using 12 printed lines) and can be either 12 or 8 characters wide.

The default separator page uses a block-letter width of 12 characters. However, using a smaller block-letter width causes separator pages to print faster. To specify a block-letter width of 8 characters, set ECAPBLK on. (To see printed examples of these header pages, see Figure 30 on page 82 and Figure 31 on page 83.)

The print lines created for separator pages are put in ECAWKBUF, beginning at the second print position. The first print position is reserved for the printer-control characters for the first block-letter print line.

Input for APSUBLK

When the block-letter service routine, APSUBLK, is entered, it contains the following:

General Register 5
The address of the exit-communication area, APSUECA

ECABLKIN
The name (ranging from 1 to 8 characters) to be used by the block letters

ECALJUST
If set on, used to left-justify the block letters

ECASLANT
If set on, used to slant the block letters

ECAPBLK
If set on, used to build performance characters

Output for APSUBLK

When leaving the service routine, APSUBLK passes the block letters to the exit in the work buffer, ECAWKBUF. The format of the output in this buffer is shown in Figure 59 on page 182.

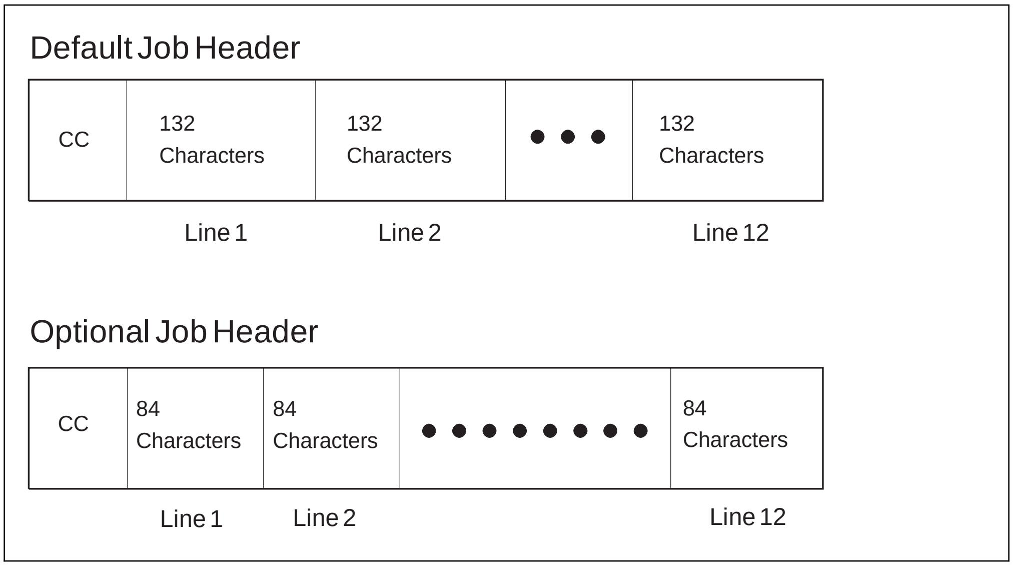


Figure 59. Format of the Output in the Work Buffer, ECAWKBUF

Programming Considerations for APSUBLK

The service routine produces corresponding block characters from the tables in APSUBTBL.

The separator routines supplied by PSF do not generate slanted block letters. You can, however, set the ECASLANT field in APSUECA to produce block letters that are slanted to the right. For the location of this field, see Figure 55 on page 169.

The separator routines supplied by PSF generate block letters that are centered on the twelve output lines. You can set the ECALJUST field in APSUECA to produce left-aligned block letters, starting in the first print position.

Note: During centering calculations, ECALJUST ignores blanks to the right of the last character in the input field.

APSUBLK saves the caller's registers in the area addressed by register 13. When register 5 is entered, it must provide APSUBLK with the address of APSUECA. APSUBLK uses ECAGSAVE as its own save area. When control is returned from APSUBLK to PSF, register 15 is reset to 0.

Exit-Put Routine, APSUPUT

The exit-put service routine, APSUPUT, enables the separator-page exit routines to put records into the PSF print-record stream.

When APSUPUT is entered, it expects the address of APSUECA in register 5. APSUPUT saves the caller's registers in the area pointed to by register 13. APSUPUT uses ECAGSAVE as its own save area.

APSUPUT uses a return-code field, ECARCODE, to identify previous failures of APSUPUT during the creation of a single separator page. Separator-page exit routines should not modify the ECARCODE field.

When APSUPUT is entered, register 15 is reset. If an error occurs during a record PUT, register 15 and ECARCODE are set, and APSUPUT does not run again for this particular separator page.

Input for APSUPUT

When APSUPUT is entered, it contains the following:

General Register 5 The address of the exit-communications area (APSUECA).

APSUECA Before running APSUPUT, the following fields must be set:

- The address of the record in ECARECAD.
- The record length in ECARECLN.
- The record-description flags in ECADRF, which identify the kind of record that APSUPUT is to process. These flags are also described in the IDXFLAG1 field, shown in Figure 61 on page 188.

Output for APSUPUT

Upon leaving the service routine, APSUPUT passes the ECARCODE and the contents of register 15.

Logical-Record Processing Exit: APSUX04

PSF provides an exit you can use to create an APSUX04 routine for modifying records in the printed output, adding records, and suppressing records. This routine runs in the same address space as does PSF.

You can modify, assemble, and install (or link-edit) the following sample (skeleton) source routines in SYS1.SAMPLIB to create an exit for logical-record processing:

- APSUX04X (for processing line data)
- APSUX04Y (for processing data already composed into pages)

For more information about line-data records, refer to *PSF/MVS: Application Programming Guide*. For more information about structured fields composing page data, refer to *Mixed Object Document Content Architecture Reference.*

Input for APSUX04

When the logical-record processing exit is entered, the contents of the registers are as follows:

Register 1	Pointer to the address of parameter area APSGEXTP
Register 13	Pointer to caller's save area
Register 14	Return address
Register 15	Entry-point address

For a logical record call:

XTPRIXP Pointer to index table entry for the print record

XTPRECP Pointer to print record

XTPRECL Length of record

For a line-mode conversion call at the beginning of each transmission of a data set:

XTP4NCAL Call Exit 04 only for the first record of the data set transmission.

XTP4CONV Do conversion flag

XTP4XOFF Horizontal offset value

XTP4YOFF Vertical offset value

XTP4WINA Wide or narrow paper flag

Note: XTP4CONV, XTP4XOFF, XTP4YOFF, and XTP4WINA are valid only if this is the record call for the first record of a data set transmission.

For an end-of-transmission call:

XTP4ABNE Abnormal end

XTP4NEOT Normal end

Output for APSUX04

XTPPIND The exit may set the XTPPIND indicator to:

- Write the current print-data record without change.
- Skip the current print-data record.
- Write the print records as described in new print-index entries constructed by the exit.

  Note: These print records may or may not include the original print record.

The APSGEXTP macro, shown in Figure 54 on page 167, provides the values used to set XTPPIND.

XTP4NCAL Call Exit 04 for the first record of the data set transmission only.

The fields are used only at the first record of the data set transmission.

XTP4CONV Do conversion flag

XTP4XOFF Horizontal offset value

XTP4YOFF Vertical offset value

XTP4WINA Wide or narrow paper flag

Other fields:

XTP4NTY Request exit notification at end of transmission

XTPRIXP Pointer to index

Programming Considerations for APSUX04

The following considerations apply to APSUX04:

- When APSUX04 receives control, no processing has occurred. When APSUX04 sees the first record of a new data set, no resource-access processing has taken place. For the first record, therefore, APSUX04 does not recognize other resources that the current data set may required.
- When the exit routine is entered, the contents of register 13 point to a standard 18-word save area that can be used to save the calle r's registers. The APSUECA control block contains a save area, ECAUSAVE, for the exit processors.

- When the exit routine is entered, register 1 points to a 1-word parameter list with the address of the parameter area, APSGEXTP. The XTPRECP field in APSGEXTP contains a pointer to the current print-data record. The XTPECAP field in APSGEXTP contains a pointer to the exit-communication area, APSUECA.

  The APSUECA work buffer, ECAWKBUF, can be used to build records and indexes. If this work buffer is too small, see "Exit Table, APSUCOM" on page 179 for information on expanding ECAWKBUF. If you expand ECAWKBUF, PSF does not use the additional storage (that exceeds the APSUECA default size) between successive exits. The PSF exits can therefore use this storage to maintain information between successive exits.

- Information describing the current print record is in the JES index table, IAZIDX, not in the APSUECA control block.

- Use register 12 as a base register for exit routines. If register 12 points to the beginning of the exit routine and the module name starts at the sixth byte in the routine, the module name appears in the PSF SDUMP, "Symptoms for This Dump," if the routine has an abend. The PSF component name and the subcomponent names can also appear on the SDUMP title page.

- This routine can set the record-processing indicator XTPPIND in the parameter area APSGEXTP to show the following:

  X'00' Write the current record. This is the default when the processing indicator is not explicitly set.

  X'01' Skip the current record.

  X'02' Write additional records as defined in a new print-index table constructed by the exit. If the current record is to be printed, it must be referenced by the new index table. The format of this index table is described in Figure 61 on page 188.

  Note: When split records are processed, and the record is not changed, the default X'00' should be coded.

- To request 3800 line-mode conversion of JES print resources to AFP print resources:

  - Determine whether line-mode conversion is required for the current data set. Set the XTP4CONV flag appropriately.

  - Set the horizontal and vertical offsets, XTP4XOFF and XTP4YOFF. These fields cause adjustments that can correspond to the adjustments allowed on the 3800-1 printer. The adjustments are based on 240 pels per inch; an adjustment of 2 inches is coded as 480. The defaults are set for a vertical adjustment of 1/2 inch and a horizontal adjustment of 0 inches.

  - | If the printer is in Cut-Sheet Emulation mode, set XTP4WINA. Set the flag ON for landscape printing, or OFF for portrait printing (which is the default). |

  - Set the line-mode flag XTP4NCAL to ON, to request that the logical-record processing exit be called only once for each time a collated copy of the data set is processed. The default setting is OFF, meaning that the exit is called for every logical record.

    Note: Line-mode conversion requires a call to the exit only at the beginning of each data set transmitted.

- To add records or to change the length of the current record, consider that:

– Information describing the current record is in the index table entry for this record. On entry to APSUX04, XTPRIXP (the pointer to the record-table index) in the APSGEXTP block will contain the address of the index-table entry for this record.

  Note: When split records are processed, the index points to the last section of the record when APSUX04 is entered. XTPRECP points to the beginning of the entire record.

– An index table must be built, using the JES IAZIDX format that describes the records to be written instead of, or in addition to, the current record. If the current record is to be included, it must also be referenced in the new index table.

  The format of IAZIDX is shown in Figure 61 on page 188. Values inside parentheses are in hexadecimal.

  Note: IAZIDX is an MVS control block, not a PSF control block.

  When control is returned to PSF, the XTPRIXP field in the APSGEXTP area must contain the address of the newly created index table. If the record being added is a control record whose first byte is the X'5A', the record-description flags—IDXANSI and IDXDSR—in the index entry must be set to show that the record is a data-stream record containing ANSI control characters.

– The IAZIDX index table, and the print-data records it describes, can be placed in the ECAWKBUF area of the APSUECA. If this area is too small, see “Exit Table, APSUCOM” on page 179 for information on expanding ECAWKBUF. If you expand ECAWKBUF, PSF does not use the additional storage (that exceeds the APSUECA default size) between successive calls. The PSF exits can therefore use this storage to maintain information between successive calls.

– When a record is being added, the record that was being processed during this exit is not printed unless the newly created index table has been updated to point to this record.

– If the exit saves (buffers) records or data across page boundaries, the exit must be able to handle PSF repositioning. PSF repositions to a page boundary. After PSF repositions and passes the next record to the exit, the exit must return the same data to PSF as it returned when the record was originally passed to the exit. For example, record 3 was the last record on page 10, and the exit returned ‘ABC’ to PSF. Record 4 was the first record on page 11, and the exit returned ‘DEF’ to PSF. If PSF repositioned to page 11 and passed record 4 to the exit again, the exit would have to return ‘DEF’ to PSF. If this was not done, data could be lost.

– For APSUX04, PSF provides the end-of-transmission (EOT) indicators (first three bits in Figure 60 on page 187) and the 3800 line-mode conversion indicators in field XTP4FLAG in APSGEXTP, as is shown in Figure 60.

Bit	Field	Meaning
1...	XTP4NTY	APSUX04 sets this bit to request an EOT notification from PSF. For notification to be received, this bit must be on every time APSUX04 is entered.
.1..	XTP4NEOT	PSF sets this bit on to show that a normal EOT has occurred (no more logical records will be processed in this transmission). APSUX04 should set the processing indicator (XTPPIND) to 2 and pass any remaining records in the buffer back to PSF. When this bit is set, no records will be passed. If no more records remain, XTPPIND does not need to be set.
..1.	XTP4ABNE	PSF sets this bit on to show that an abnormal EOT has occurred, either because of a printer I/O error or because of a JES operator command (reposition, cancel, or interrupt). APSUX04 should discard any records remaining in the print buffer, and return to PSF. (PSF will not process any returned records if an abnormal EOT has occurred.)
...1	XTP4NCAL	PSF sets this bit OFF when APSUX04 is called for the first record of the transmission; this maintains the current processing, allowing the logical record exit to be called for every record. APSUX04 may set this bit ON to request that the logical record exit be called only for the first record of each data set transmission (next copy of the data set or the next data set). This bit does not affect the EOT call.
.... 1...	XTP4CONV	PSF sets this bit OFF when APSUX04 is called for the first record of the transmission. APSUX04 may set this bit ON for the first record of the transmission to request that conversion for 3800 line-mode printing be performed. If APSUX04 sets this bit ON for any other records of the transmission, the bit is ignored.
.... .1..	XTP4WINA	Valid only if CSE and 3800 Line-Mode Conversion. PSF sets this bit OFF when APSUX04 is called for the first record of the transmission; this causes portrait pages to be generated. APSUX04 may set this bit ON for the first record of the transmission to request that landscape pages be generated. If APSUX04 sets this bit ON for any records of the transmission, the bit is ignored.

Figure 60. APSUX04 Indicators in XTP4FLAG Field

In Figure 61, the first 12 bytes represent a fixed-length header. The information starting at field ADXENTRL at offset 12 (X'0C') is repeated for each index entry. This control block is mapped by macro IAZIDX. For more information, refer to your JES publications.

Offsets	Length	Name	Description
	Fixed-Length Index Header		
0 (0)	4	IDXID	Index-table identifier set to IDX
4 (4)	2	IDXNUM	Number of table entries
6 (6)	2	IDXTOK	JES-supplied token
8 (8)	4	*	Reserved
	Repeating Index Entry		
12 (0C)	2	IDXENTRL	Length of this index entry
14 (0E)	2	IDXRECL	Length of print data record associated with this index entry
16 (10)	1	IDXFLAG1	Flags describing print data record
	1...	IDXDSR	Indicates AFPDS record
	.1..	IDXLMR	Indicates line-mode record
	..1.	IDXANSI	Indicates that the record contains ANSI control characters
	...1	IDXMACH	Indicates that the record contains machine control characters
	.... 1...	IDXSRS	Split record, start of record
	.... .1..	IDXSRM	Split record, middle of record
	.... ..1.	IDXSRE	Split record, end of record
	1	IDXOPJ	OPTCD=J used for record
17 (11)	3	*	Reserved
20 (14)	4	IDXRADR	Address of the print data record associated with this index entry
24 (18)	8	IDXRECID	Record identifier token

Figure 61. Format of JES Index Table, IAZIDX

SMF Type 6 Record Exit: APSUX05

You can use this exit to modify system management facilities (SMF) type 6 records. SMF is an optional control program that provides a way for you to collect information to aid in evaluating system use. If you have not provided an APSUX05 routine, PSF creates a standard type 6 record and writes it to the SMF data set.

Note: You can use this exit to suppress SMF type 6 records for JESNEWS data sets.

PSF provides a sample (skeleton) SMF type 6 record-processing routine that you can use to create, assemble, and install (or link-edit) an exit that modifies SMF type 6 records. (For more information, see Table 14 on page 165.) This routine runs in the same address space as does PSF.

For a description of the SMF type 6 records that PSF creates, see “System Management Facility Type 6 Records” on page 93. For detailed information about SMF type 6 records, refer to *System Management Facilities*.

Input for APSUX05

When the SMF type 6 record-processing exit is entered, the contents of the registers are as follows:

Register 1	Pointer to the address of parameter area APSGEXTP
Register 13	Pointer to caller's save area
Register 14	Return address
Register 15	Entry-point address

XTP5NEWS	JESNEWS data set indicator
XTPRECP	Pointer to SMF type 6 record
XTPRECL	Length of record

Output for APSUX05

XTPPIND The exit may set the XTPPIND indicator to:

X'00'	Write the SMF type 6 record
X'01'	Skip the SMF type 6 record

Programming Considerations for APSUX05

The following considerations apply to APSUX05:

- When APSUX05 is entered, register 13 points to a standard 18-word save area that can be used to save the caller's registers. The APSUECA control block contains a save area, ECAUSAVE, in case APSUX05 needs to make external calls.
- When the exit routine is entered, register 1 points to a one-word parameter list with the address of the parameter area, APSGEXTP. APSGEXTP contains a pointer to the exit communications area, APSUECA.

  APSUX05 can use the APSUECA work buffer, ECAWKBUF. If this work buffer is too small, see “Exit Table, APSUCOM” on page 179 for information on expanding ECAWKBUF. If you expand ECAWKBUF, PSF does not use the additional storage (that exceeds the APSUECA default size) between successive calls. The PSF exits can therefore use this storage to maintain information between successive calls.
- You should use register 12 as a base register for exit routines. If register 12 points to the beginning of the exit routine, and if the module name starts at the sixth byte in the routine, the module name appears in the PSF SDUMP, “Symptoms for This Dump,” if the routine has an abend. The PSF component name and subcomponent names can also appear on the SDUMP title page.
- Macro IFASMFR describes the SMF type 6 record. For more information, refer to your publication on System Management Facilities (SMF).

  Note: Fields in IFASMFR provided by SVC 83 are not available to APSUX05.
- PSF can be processing a new data set during the same time that APSUX05 is processing the SMF type 6 record for a preceding data set. This may result in APSUX05 being called for the preceding data set at the same time one of the other exits is being called for the current data set.
- APSUX05 is called from the NOTIFY subtask; the following exits are called from the FSA subtask:
  - APSUX01
  - APSUX02

  - APSUX03
  - APSUX04
  - APSUX06
  - APSUX07

  APSUX05 can be running during the same time that another exit in the FSA subtask is running for a new data set.

- Two exit communication areas (ECAs) are created for each FSA. ECEFECAP points to the ECA that is passed to the following exits:

  - APSUX01
  - APSUX02
  - APSUX03
  - APSUX04
  - APSUX06
  - APSUX07

  ECENECAP points to the ECA that is passed to APSUX05.

- You can change the contents of the SMF type 6 record, but you cannot change its length.
- You cannot change the address of the SMF type 6 record in XTPRECP. If you do change the address, it is ignored.
- Each time a different rotation of an outline font is activated, the SMF record is updated; that is, the number of fonts is incremented by 1). With raster fonts, however, a coded font is counted only once, regardless of the number of rotations of the font that are activated.

PSF Message Exit: APSUX06

APSUX06 defines a message exit that is given control before most of the messages generated by PSF FSA APS*xxx*I and APS*xxx*A, are issued.

You can use this exit to suppress or redirect messages. You can redirect messages to the operator console, the security console, or both. To suppress or redirect messages, use the XTP6ACT flag. The exit cannot request that a message destined for the operator or the security console be printed with the data set.

You can use this exit to release the current data set to JES and request that the system hold it. To do this, set the XTP6HOLD flag. After the data set runs to completion, PSF marks the data set as one that cannot be selected.

You can also use this exit to terminate a printer functional subsystem application (FSA). To do this, set the XTP6TFSA flag. After the data set runs to completion, PSF terminates the FSA.

If you set both flags, PSF terminates the FSA and the system does *not* hold the data set.

PSF provides a sample (skeleton) routine for inspecting PSF messages. You can use this routine to create, assemble, and install (or link-edit) an exit to inspect messages. (For more information, see Table 14 on page 165.) This routine runs in the same address space as does PSF.

Input for APSUX06

When the PSF message-inspection exit is entered, the contents of the registers are as follows:

Register 1	Pointer to the address of parameter area APSGEXTP
Register 13	Pointer to caller's save area
Register 14	Return address
Register 15	Entry-point address

XTP6PROC FSA identifier:

channel-attached printers: DD*nnn*, where *nnn* is the device-address
non channel-attached printers: label of PRINTDEV statement

XTP6MSG Message ID (APS*xxx*I or APS*xxx*A)

XTP6ACT Message action to be taken by PSF:

XTP6WTO	Put message to the operator console.
XTP6SEC	Put message to the security console.
XTP6PAP	Print message with the data set.

Output for APSUX06

XTP6HOLD Request that the system hold the data set. PSF marks the data set as one that cannot be selected.

XTP6TFSA Terminate FSA.

XTP6ACT Message action requested by the exit. You can redirect the message to any or all of the destinations by setting the correct combination of flags.

XTP6WTO	Put the message to the operator console.
XTP6SEC	Put the message to the security console.
XTP6PAP	Print the message with the data set.
XTP6SUP	Suppress the message.

Programming Considerations for APSUX06

The following considerations apply to APSUX06:

- When the exit routine is entered, register 13 points to a standard 18-word save area that can be used to save the caller's registers. APSUX06 might receive control while one of the other exits is active. If APSUX06 needs to make an external call, the code should use ECAMSGSV, a unique save area provided for this exit in the ECA.
- When the exit routine is entered, register 1 points to a one-word parameter list with the address of the parameter area, APSGEXTP. APSGEXTP contains a pointer to the exit-communications area, APSUECA.

  APSUX06 can use the APSUECA work buffer, ECAWKBUF. If this work buffer is too small, see “Exit Table, APSUCOM” on page 179 for information on expanding ECAWKBUF. If you expand ECAWKBUF, PSF does not use the additional storage (that which exceeds the APSUECA default size) between successive exits. The PSF exits can therefore use this storage to maintain information between successive exits.

Notes:

1. The transmission-count field, ECATRNC, is not updated before this exit is invoked.
2. ECEOUTTK and ECEJDVT are not updated before this exit is invoked.

- Use register 12 as a base register for exit routines. If register 12 points to the beginning of the exit routine, and the module name starts at the sixth byte in the routine, the module name appears in the PSF SDUMP, “Symptoms for This Dump,” if the routine has an abend. The PSF component name and subcomponent names can also appear on the SDUMP title page.
- XTP6, a dsect in APSGEXTP, contains information that APSUX06 needs to have in order to determine its action. The pointer XTPRECP within APSGEXTP provides the address of XTP6. XTP6 is mapped within macro APSGEXTP. For more information on APSGEXTP, see Figure 54 on page 167.
- The message action, XTP6ACT, can be changed by the exit as follows:
  - If the exit does not alter XTP6ACT, PSF takes the action that the message originally called for.
  - If the exit sets XTP6ACT to 0, PSF suppresses the message.
  - If the exit sets XTP6ACT to one or more actions, PSF takes those actions.
- The following restrictions apply to XTP6ACT:
  - Action messages, noted by the format APS*xxx*A, cannot be suppressed. If the exit requests that an action message be suppressed, PSF ignores the request and sends the message to the operator.
  - The exit cannot request that a message be printed with the data set, XTP6PAP, unless that was the original destination set by PSF.
  - If the message is suppressed, PSF still traces the message.
  - PSF issues some messages twice, once to the operator and once for printing with the data set. When a message is to be redirected, you might want to code a check for duplicate messages.
- Consider the following restrictions:
  - This exit is entered before the FSA part of PSF issues a message from PSF. The printer runs asynchronously to this exit, and is not affected by its actions. While this exit is running, the FSA part of PSF is not running.

    If any of the following messages is issued, execution is not transferred to the exit:

    APS000I APS001I APS002A APS003A APS004I APS005I
    APS006I APS022I APS023I APS025I APS038I APS042I
    APS043I APS044I APS045I APS046I APS047I APS048I
    APS049I APS050I APS051I APS052I APS053I APS054I
    APS055I APS060I APS062I APS063I APS064I APS065I
    APS066I APS067I APS068I APS226I APS516I APS517I
    APS551I APS605I APS607I APS608I APS610I APS922I
    APS955I APS957I

    Execution is not transferred to this exit in any of the following conditions:

    - For any message generated by the APSRMARK utility—messages APS580I to APS599I.
    - For any message generated by the trace formatter utility, APSTRFMT—messages APS960I to APS979I).

    - For any message generated by the APSRFLU utility—messages APS1700I to APS1711I.
    - For any message generated by the operator interface—messages APS620I through APS699I, with the exception of message APS691I.
  - The error that the message reports is not necessarily related to the data set that PSF is currently processing. The JSPA also provides information about the data set that PSF is processing (not necessarily about the data set to which the message belongs).
  - When this exit is called, the JSPA pointer, XTPJSPAP, may be set to 0.

Resource-Management Exit: APSUX07

PSF can call APSUX07 at various times while processing a print data set during an FSA session. APSUX07 controls these calls by setting the appropriate flags in the XTP7NOTY field of APSGEXTP. (For a listing of the XTP7 section in APSGEXTP, see Figure 69 on page 280.) This set of calls enables you to manage resources to do the following:

- Specify whether fonts should be mapped to outline fonts, if the target printer supports outline fonts.
- Specify that PSF should verify the outline font exits in the host library when raster fonts are mapped to outline fonts.
- Specify whether a resource should be loaded from the host library, even if it already resides in virtual storage or in the printer.
- Specify which resources of a given type should be saved in the printer across data set boundaries.
- Substitute one resource for another.
- Enforce the security standards of your installation by terminating jobs that attempt to access resources for which they are not authorized.

You can select these resource-management functions for all types of resource or for specific types of resource. For example, you can use the exit to manage fonts but not to manage overlays.

Invoking APSUX07

The call to APSUX07 can be initiated from different places in PSF. A call to APSUX07 is defined by the following fields in the XTP7 section of APSGEXTP:

XTP7ETYP	Exit type
XTP7RTYP	Resource type
XTP7DSTY	Data set type
XTP7DSAT	Auxiliary data set type

Each call provides access to a specific area of PSF resource management. For a listing of APSGEXTP, see Figure 54 on page 167.

Sequence of Events

1. If APSUX07 is present at PSF initialization,[15] PSF sets the XTP7INIT flag in the XTP7ETYP field to 1, to indicate that this is the first call to the exit, and then calls the exit. At this time, APSUX07 may set flags in the XTP7NOTY fields to specify the resources and events for which the exit wants control.
2. Additional calls to the exit depend on the status of the XTP7NOTY flags that the exit set during the INIT.
3. If the exit is not installed, no calls or references to the exit occur.

APSUX07 Subroutine Calls

The input and output for the APSUX07 subroutine calls (summarized in Table 15) are described in terms of their associated processing indicators within each control block. Table 19 on page 302 references the input and output for each call, and Appendix C, “Field Descriptions for APSUX07,” provides a listing of the control blocks.

Table 15. Summary of APSUX07 Subroutine Calls

Subroutine call and page number	**Description**
“Initialization Call (INIT)” on page 203	Defines the control parameters of the resource exits.
“Begin-Data-Set Call (BDSC)” on page 205	Controls the processing of the data set.
“Resource-Access Call (RAC)” on page 205	Controls processing for individual resources—for example: • Specifying whether a resource should be loaded from the host library, even if it already resides in virtual storage or in the printer. • Substituting one resource for another.
“Resource-Load Call (RLC)” on page 210	Requests or releases control of serially reusable resources, using ENQ or DEQ. (ENQ and DEQ are MVS system macros. For more information on system macros, refer to the macro publication for your version of MVS.) You can also use RLC to collect statistics.
“Resource-Deletion Call (RDC)” on page 211.	Controls resource deletion at data set end (DSE).
“Termination Call (TERM)” on page 213.	If needed, restores processing states at FSA-termination time.

[15] For a description of how PSF determines that an exit is present, see “Changing Exit Routines” on page 164.

Programming Considerations for APSUX07

The following general considerations apply to APSUX07:

- Upon entry to this routine, register 1 points to a one-word parameter list that contains the address of APSGEXTP. APSGEXTP contains a pointer to the exit-communications area, APSUECA. The work buffer, ECAWKBUF, in the ECA control block can be used to build records and indexes. This provides a method for maintaining information between successive exits. (For more information on ECAWKBUF, see Figure 55 on page 169.)
- APSUX07 can receive control at the same time that one of the other exits is active. APSUX07 should always use ECARSCSV, a unique save area provided for this exit in the ECA.

  Additional pointers are provided for data areas not specific to PSF. For example, the JES job-separator page data area, IAZJSPA, is pointed to by the XTPJSPAP field in the APSGEXTP control block.
- Upon entry to this routine, register 13 points to a standard 18-word save area that can be used to save the caller's registers.
- Use register 12 as a base register for all exit routines. If register 12 points to the beginning of an exit routine, and the module name begins in the sixth byte of the routine, the module name appears on the SDUMP title page if an abend occurs.

  Note: PSF components and subcomponents also appear on the SDUMP title page.

Page Segment Considerations

PSF can convert a page segment before loading it into the printer, to ensure that the page segment is compatible with the units of logical measurement (L-units) of the page. Unlike an overlay, a page segment does not specify its own L-units. A page segment included on a page inherits the L-units of that page.

Because PSF supports multiple L-units, multiple copies of the same page segment can be loaded in the printer. Although they appear to be the same page segment, they are actually different versions of the page segment, loaded into the printer with different L-units.

Special Performance Considerations

- XTP7NOTY flags are located in the APSGEXTP parameter area. See “XTP7 Parameter Area (DSECT)” on page 279 for more information. XTP7NOTY flags should not be turned on unless a specific function is intended.
- Retaining an often-used overlay or page segment over data set boundaries can improve printer throughput by eliminating the need to load the resource for each data set. Do not, however, retain resources arbitrarily. Page segments can take up substantial amounts of storage, and retaining overlays can force PSF to retain unneeded resources.

3800-Compatible Media Origin Considerations

APSUX07 can aid in migrating from a 3800 printer to a 3835 or 3900 printer by setting the media origin of the destination printer to match that of the 3800 printer. The location of the media origin is one of the main differences between the 3800 and other AFP continuous-forms printers. For an explanation of these differences, refer to the appendix describing “Compatibility, Conversion, and Performance” in *AFP: Printer Information*.

When migrating 3800 print jobs, you can provide a compatible media origin by including the proper presentation options in the form definition for each job. Alternatively, you can tailor APSUX07 to enforce a 3800-compatible form origin for any designated job.

Note: This compatibility of APSUX07 should be used only for purposes related to migration. Jobs designated by this exit bit should not specify capabilities that are not supported by the 3800 printer, such as N_UP, page modification controls, or post-processing controls. If, for example, N_UP is specified in the form definition of a job designated as 3800-compatible, the 3800-compatible bit is ignored. PSF processes the job, using the N_UP capabilities of the 3900 or 3835 printer to which the job is sent, but issues a warning message noting the conflict between the exit bit and the form definition.

If the exit bit is set for a job that is routed to a cut-sheet printer or routed to a continuous-forms printer that does not support changeable media origin, the exit bit is ignored, but no message is issued.

You can use APSUX07 to control whether to set the media origin to the 3800 media origin.

XTP738MO Controls setting the media origin for the user's data set on continuous-forms printers to the 3800 media origin.

XTP7JHMO Controls setting the media origin for the job header on continuous-forms printers to the 3800 media origin.

XTP7JTMO Controls setting the media origin for the job trailer on continuous-forms printers to the 3800 media origin.

XTP7DHMO Controls setting the media origin for the data set header on continuous-forms printers to the 3800 media origin.

XTP7MDMO Controls setting the media origin for the message data set continuous-forms printers to the 3800 media origin.

These flags default to 0. The default specifies that PSF is not to set the media origin to the 3800 media origin. If any of the flags is set to 1, PSF sets the media origin to the 3800 media origin for that particular page or pages. Any or all of the flags can be set to 1 during a call to APSUX07. These flags can be set during the INIT call or the BDSC call, and remain set between calls; only APSUX07 can change the state of these flags.

Considerations on Suppressing an Interrupt Message Page

You can use APSUX07 to control whether the interrupt message page that PSF inserts in your printed output is to be printed:

XTP7IPSU Controls the suppression of the interrupt message page.

This flag defaults to 0. The default specifies that PSF is to print the interrupt message page. If the flag is set to 1, PSF suppresses the interrupt message page. The flag can be set during the INIT call, and it remains set between calls; only APSUX07 can change the state of this flag.

Note: If printing to microfilm, PSF automatically suppresses the printing of interrupt messages, and this flag has no effect.

Considerations on Redirecting a Message Data Set

With APSUX07 you can control the capability of redirecting a message data set as a SYSOUT data set to another CLASS or DEST for viewing or printing.

XTP7MDSD Controls the redirection of a message data set

This flag defaults to the PIMSG keyword setting on the OUTPUT statement for the message data set. If the flag is set to 0, PSF will not send the message data set to a SYSOUT data set. If the flag is set to 1, PSF will send the message data set to a SYSOUT data set. The keywords specified on the OUTPUT statement for the message data set referenced by the MESSAGE= keyword in the PRINTDEV statement will be attributed to the generated SYSOUT data set. The flag can be set during the INIT call or during the BDSC call. This flag does *not* remain set between calls; it is set to the value based on the PIMSG keyword.

Overlay Considerations

Before an overlay containing fonts or page segments can be printed, the overlay and its associated resources must be loaded in the printer. Fonts and page segments must be defined in one of two ways:

- In the Active Environment Group (AEG) of the current page definition. PSF makes sure that the resources are loaded before the page using the resources begins to print.
- In the AEG of the overlay itself. These resources are identified as *nested* resources, and must remain in the printer if the overlay remains loaded. A nested resource is a resource mapped in the overlay.

For more information about the AEG, refer to *Mixed Object Document Content Architecture Reference.*

If APSUX07 is not installed, PSF deletes all overlays at data set end. If APSUX07 is installed, you can retain overlays over data set boundaries by modifying the PSF resource-deletion algorithm. (For more information on deleting resources, see “Full Control of Resource Deletion” on page 212.)

Considerations on Converting a Multiple-Celled IM1 Image

With APSUX07, you can control whether a multiple-celled IM1 is converted to a single IOCA image or to multiple IOCA images:

XTP7C2SI Controls the conversion of a multiple-celled IM1 image.

This flag defaults to 0. The default specifies that PSF is to convert a multiple-celled IM1 image to multiple IOCA images. If the flag is set to 1, PSF converts a multiple-celled IM1 image to a single IOCA image. The flag can be set during the INIT call or the BDSC call, and it remains set between calls; only APSUX07 can change the state of this flag.

This request is honored only if the printer does not support IOCA replicate and trim.

This change improves performance only if the image is a 240-pel IM1 celled image that requests replication. For any other case, this change may degrade performance.

Considerations on Eject-to-Front-Facing Function

If your continuous-forms printer supports the eject-to-front-facing (ETFF) function, PSF issues an ETFF command before the job-header page and before the start of a new document. With APSUX07, you can control the ETFF by use of two flags:

XTP7ETFF Controls the ETFF issued between documents in a data set.

XTP7EFFH Controls the ETFF issued before the job-header page.

Both of these flags default to 0. The default specifies that PSF is to issue the ETFF. If the flags are set to 1, PSF does not issue the ETFF. The flags can be set during the INIT call or the BDSC call and remain set between calls; only APSUX07 can change the state of these flags.

Considerations on Restartable Abends

PSF automatically restarts when certain abend reason codes occur. See "Understanding Restartable Abends" on page 249 for more information on restartable abends.

Because restarting after an abend may cause duplicate pages to print without notification to the operator or the job submitter, you may want to prevent PSF from automatically restarting after abends. With APSUX07, you can prevent PSF from automatically restarting by using the following flag:

XTP7RSTR Controls restarting of the abend

PSF sets XTP7RSTR to 1 when an abend occurs that is restartable; otherwise XTP7RSTR is set to 0. You can prevent PSF from restarting after a restartable abend, by setting the flag to 0 in APSUX07 during the TERM call. You *cannot* cause PSF to restart after an abend by setting the flag to 1. This flag does not remain set between calls.

XTP7ABRC contains the abend reason code. For example, if the abend reason code is E19, then XTP7ABRC contains X'0E19'. If no abend occurred, then XTP7ABRC contains X'0000'.

Considerations on Cut-Sheet Emulation Mode

With APSUX07, you can control how PSF checks the pages for Cut-Sheet Emulation mode. If a page is checked and found not to fit 2 up on the sheet, the page is printed in normal 1_UP. For more information, see Chapter 9, "Using Cut-Sheet Emulation Mode" on page 103.

XTP7ACSE Controls the page checking for cut-sheet emulation mode

This field defaults to 0(XTP7NCK). The default specifies that PSF is not to check to see whether the page fits 2 up on the sheet. If this field is set to 1(XTP7K1P), PSF checks only the first page printed for a new code page to see if it fits 2 up on the sheet. If the field is set to 2(XTP7CKAP), PSF checks all the front side pages to see

| if they fit 2 up on the sheet. The field can be set during the INIT call or the BDSC
| call, and it remains set between calls; only APSUX07 can change the state of this
| field.

Considerations on Forms Marking of an Interrupt Message Page

If your printer supports forms marking or offset stacking, you can identify the location of an interrupt message page that PSF inserts into your printed output. When an error such as a paper jam or a JES repositioning command occurs, PSF inserts an interrupt message page into your printed output. When PSF is marking the interrupt message page, you can request multiple copies of the page. Requesting multiple copies causes the markings to be more visible from the side of the stack of paper.

The offset stacking request is separate from the mark forms request; that is, you can request offset stacking but not request forms marking.

The following fields are used to control this function:

XTP7IPMF Controls forms marking for the interrupt message page.

XTP7IPOS Controls offset stacking for the interrupt message page.

XTP7IPCC Controls the number of copies printed for the interrupt message page.

Both of the flags, XTP7IPMF and XTP7IPOS, default to 0. The default specifies that PSF is not to issue any mark forms or offset stacking request for the interrupt message page. The flags and the number of copies can be set during the INIT call, or the BDSC call and remain set between calls. Only APSUX07 can change these fields. If the copies field, XTP7IPCC, is not specified, or if an incorrect value is specified, it defaults to 1. XTP7IPCC is used only when XTP7IPMF is set to 1.

Considerations Related to Page Definitions and Form Definitions

The following page definitions and form definitions are useful when you are working with APSUX07:

User-specified A page definition specified in the PAGEDEF or FCB parameter in the JCL; or, a form definition specified in the FORMDEF parameter in the JCL. These definitions are stored in a user library.

Auxiliary A page definition or a form definition specified in the PSF startup procedure for use in an auxiliary data set. Auxiliary data sets are:

- Job headers
- Job trailers
- Data set headers
- Message data sets

JES default A page definition specified in the JES2 or JES3 initialization statements.

Inline A page definition or a form definition within the print data set.

Modified-default A page definition or form definition that was the default specified in the PRINTDEV statement of the PSF startup procedure but that was modified by a JCL OUTPUT statement, the PRINTDEV statement, or the JES default font. A page definition is modified by a font list specified in the CHARS or USC parameter. A form definition is modified by *groupvalue* in the COPIES parameter, or by the FLASH parameter.

Processing Considerations: The following considerations apply to the processing of page definitions and form definitions:

- An inline page definition or form definition is read into PSF directly from the user's data set. PSF uses the inline page definition or inline form definition only for the data set that contains it, and deletes it after the processing of that data set is complete.
- The name of a user-specified page definition or form definition is passed to PSF by JES. If unable to find or use an existing version of that resource, PSF finds the resource in the host library and loads it into virtual storage. If the resource is in a user library, it is loaded into virtual storage from the user library, even if an existing system version of the resource could be used. PSF uses the user library resource only for the data set that specifies it. For more information, see "User Libraries in Direct-Printing Mode" on page 48.
- A JES default page definition appears to PSF as if it were a user-specified page definition. PSF does not identify the page definition as a default page definition. No JES default form definition can be specified in JES.
- Device-default and auxiliary page definitions and form definitions are loaded at the beginning of the FSA session, and remain loaded until the session ends.
- When you are using a modified-default page definition or form definition, PSF copies the default resource specified in the PRINTDEV statement and modifies it with the respective parameters from the JCL OUTPUT statement or from the PRINTDEV statement. PSF modifies the default resource at the same time it is setting up the environment for the data set. No new resource object is created; therefore, the resource exit is never called with an APSURLST entry for a modified-default page definition or form definition.

Selection Hierarchy: For information on the selection hierarchy of page definitions and form definitions, see "How PSF Selects Resources" on page 117.

Data-Set Termination Considerations

By setting the XTP7TDS flag to 1, you can use APSUX07 to request termination of the current data set. You can set this flag during the BDSC, the RAC, or the RLC call.

A terminated data set is usually purged. To request that a terminated data set be returned to JES, set the XTP7HQUE flag to 1. PSF then marks the data set as one that cannot be selected.

When a data set is terminated, PSF adds APS954I to the message data set. To add text to this message, supply:

- A pointer to your text in XTP7MSGP
- The length of the text in XTP7MSGL—up to 64 bytes

Note: If either XTP7MSGP or XTP7MSGL is 0, PSF does not add text to APS954I. XTP7TDS, XTP7HQUE, XTP7MSGP, and XTP7MSGL are in the XTP7 section of APSGEXTP. For more information on XTP7, see Figure 69 on page 280.

3800 Considerations

You can use APSUX07 to select the 3800 hardware default font. Initially, the XTP7HDF flag is set to 0, indicating that the 3800 should use the hardware default font. APSUX07 can set XTP7HDF to 1, to specify that the printer is to take the first font in the current font list as the hardware default.

To increase the amount of raster-pattern storage available, you can specify the first font in the current font list as the hardware default. Consider this option when a data set is terminated with PSF message APS921I.

Note: The hardware default font may differ from one printer to another and from one operating system to another. When the same application is printed on a different printer or under a different operating system, therefore, using this option can produce output incompatibilities.

3800 Forms Marking

The 3800 prints 3 to 5 copies of the marked form; the number is a function of the printer. The marking of forms is the same for data pages (for example, the interrupt message page) as it is for trailer pages. For the 3800 printer, PSF ignores the copies field, XTP7IPCC; the 3800 device controls the number of copies printed of the interrupt message page. For more information, see “Forms Marking” on page 88.

Considerations Related to User Libraries

The following processing considerations apply to resources stored in a user library:

- APSUX07 cannot substitute user library resources. PSF ignores a request to do so.
- APSUX07 cannot keep a user library resource across data set boundaries. User library resources are deleted before the Resource-Deletion Call (RDC). For details, see “Resource-Deletion Call (RDC)” on page 211.
- APSUX07 is passed a concatenation retrieval list, APSLCRL, for user library resources.

Considerations Related to Inline Resources

PSF/MVS supports inline resources—form definitions, page definitions, overlays, page segments, character sets, code pages, and coded fonts—as well as inline object containers. The following processing considerations apply to inline resources:

- APSUX07 cannot substitute inline resources. PSF ignores a substitution request for an inline resource.
- APSUX07 cannot keep an inline resource across data set boundaries. PSF deletes inline resources before the Resource-Deletion Call (RDC).
- APSUX07 is not passed a concatenation retrieval list (APSLCRL) for inline resources.
- APSUX07 cannot force a reload of an inline resource. When PSF is unable to reload a resource, RLSTRRI is set on. If PSF requests that the resource be reloaded, the request is ignored; no error occurs. If APSUX07 requests substitution, and PSF determines that the substitution request should be honored, PSF attempts to retrieve the substituted resource only from the library from which the original resource would have been retrieved.

Considerations Related to Outline Fonts

PSF/MVS 2.2.0 supports outline fonts as well as raster fonts. When the printer does not support outline fonts, PSF maps an outline font to an equivalent raster font, as is explained in Chapter 15, “Supporting Outline Fonts.” This mapping function is always active; it occurs before the resource-access call. For more information, see “Font Mapping” on page 145.

For performance considerations, see “Using Outline Fonts” on page 223. For information on changing the mapping, see “Using the Font Mapping Tables” on page 146.

The following processing considerations apply to outline fonts:

- If Exit 7 is invoked and a font substitution request is made, but the printer does not support the type of font substituted, PSF terminates the data set and issues error messages. PSF does not map the font to an equivalent raster font or outline font, because the substitution by Exit 7 overrides all other actions.

  When you request that an outline font be reloaded, you may experience some performance degradation if several coded fonts are using the same code page or character set. All the fonts sharing the code page or character set are deactivated and then reactivated, all from the same reload request. (Deactivating and reactivating an outline coded font does not make it necessary to send all the font data to the printer again; only the commands necessary to associate the code page and the character set are sent.)

- When Exit 7 deletes fonts, it deletes them at the coded font level; that is, Exit 7 deletes both the code page and the character set. Because different outline coded fonts may use the same code pages and character sets, other instances of these font members are deleted (and made unusable) at the same time.

Considerations Related to Mapping to Outline Fonts

If your printer supports outline fonts and you have existing applications that use raster fonts, and you want to use outline fonts without changing the applications, you can request that PSF map fonts to outline fonts.

XTP7MTOF controls whether PSF maps fonts to outline fonts. The XTP7MTOF flag defaults to zero. The default specifies that PSF is *not* to map fonts to outline fonts. If the flag is set to 1, PSF attempts to map the font to a corresponding outline font, using user and system mapping tables. For more information, see “Font Mapping” on page 145 and “Using the Font Mapping Tables” on page 146. The flag can be set during the INIT call, and it remains set between calls. Only APSUX07 can change the state of this flag.

Note: PSF does *not* map a font to an outline font when:

- The font is used on an auxiliary data set.
- The font is an inline font or a font from a user library.
- A double-byte raster font section is requested by character set and code page name.

  Note: Double byte raster coded fonts are prefix-mapped to equivalent outline coded fonts.

- PSF senses that the font is already an outline font.
- The target printer does not support outline fonts.

For performance considerations, see “Using Outline Fonts” on page 223.

Considerations Related to Recover-from-Font-Not-Found Flag

If your printer supports outline fonts and you have requested that PSF map to outline fonts (set XTP7MTOF to 1), but many pages in your jobs are terminated because the outline font identified through the mapping algorithms did not exist on the host, you may want to turn on the Recover-from-Font-Not-Found function.

The XTP7RFNF flag can be set during the INIT call, and remains set between calls. Only APSUX07 can be used to change the state of this flag. The XTP7RFNF flag defaults to 0, which specifies to PSF that it does *not* need to make sure that the mapped font exists before proceeding. If the mapped font does not exist, PSF issues an error message and terminates the page.

To specify that PSF is to make sure that the outline font derived from the mapping algorithms (for example, the mapped font) exists before proceeding, set the XTP7RFNF flag to 1. If XTP7RFNF is set to 1, PSF performs library queries to determine whether the mapped font exists. If PSF determines that the mapped font exists, it proceeds to load the mapped font. If XTP7RFNF is set to 1 and the mapped font is not found inline or in any specified user library or system library, PSF does *not* map the font, but instead loads the raster font.

Note: Because there are additional library queries for every mapped font resource, **setting XTP7RFNF to 1 degrades the performance of PSF**.

Considerations Related to Huffman Compression

By setting the XTP7HCA flag in the initialization call, you can use APSUX07 to specify that PSF is to compress the data by use of the Huffman compression algorithm before sending it to the printer. Data compressed by this means can be sent only to SDLC-attached printers that have the decompression algorithm—that is, AFCCU printers; the algorithm cannot be used if the AFCCU printers are attached to DPF or PSF Direct. For a description of the algorithm and information about performance, see “Transmission and Compression of Data” on page 223.

The XTP7HCA flag defaults to B'0'. The default specifies that PSF is not to compress the IPDS data. If the flag is set to B'1' and the printer is SDLC-attached, PSF compresses the IPDS data by use of the Huffman compression algorithm. The flag can be set during the INIT call, and it remains set between calls; only APSUX07 can change the state of this flag.

Initialization Call (INIT)

The INIT call is made once, after the FSA has been initialized for each device. By modifying XTP7NOTY flags (listed in Figure 69 on page 280), you can use the INIT call to set up for future calls to APSUX07. As defaults, XTP7NOTY flags are set inactive; however, you can modify these flags to:

- Control resource access by type[16]; set any of the XTP7NACC flags.
- Load resources by type[16]; set any of the XTP7NLD flags.
- Delete resources by type; set any of the XTP7NDEL flags.

[16] The resource-*access* event occurs whenever the job refers to the resource by name. The resource-*load* event occurs when the resource is loaded either into the printer or, for page definitions and form definitions, into PSF virtual storage. A resource can be loaded into the printer more than once during the processing of a data set.

- Receive notification when a data set begins; set XTP7EBDS=1.
- Receive notification when an FSA terminates; set XTP7ETRM=1.

INIT also provides pointers in XTP7CRLP to concatenation retrieval lists (CRLs). The CRLs contain information about system resource libraries as they were set up by use of the PSF startup procedure.

This call has several other uses:

- To control ETFF. For more information on ETFF, see “Considerations on Eject-to-Front-Facing Function” on page 198.
- To control whether to set the media origin to the 3800 media origin. For more information, see “3800-Compatible Media Origin Considerations” on page 196.
- To control the conversion of a multiple-celled IM1 image. For more information, see “Considerations on Converting a Multiple-Celled IM1 Image” on page 197.
- To control the marking and offset stacking of the interrupt message page. For more information, see “Considerations on Forms Marking of an Interrupt Message Page” on page 199.
- To modify the “reasonable resource loading values” (RRLV). PSF uses the RRLV to determine the number of resources saved in a printer or in virtual storage at the end of each data set. For more information, see “Partial Control of Resource Deletion” on page 211.
- To control the suppression of the interrupt message page. For more information, see “Considerations on Suppressing an Interrupt Message Page” on page 197.
- To control the message data set redirection. For more information, see “Considerations on Redirecting a Message Data Set” on page 197.
- To enable the compression of the data sent to the printer by use of the Huffman compression algorithm. For more information, see “Considerations Related to Huffman Compression” on page 203.
- To control how PSF checks the pages for Cut-Sheet Emulation mode. For more information, see “Considerations on Cut-Sheet Emulation Mode” on page 198.

Input and Output Indicators for INIT

For a complete cross-reference of the XTP7 and RLST fields used as input and output, see Table 19 on page 302.

Programming Considerations for INIT

With APSUX07 you can run the system ENQ or DEQ macros on the resource library that contains the resource about to be loaded at the resource-load call. PSF supplies the following information:

- CRLs of all partitioned data set names.
- Pointers to the CRL for each type of resource.
- A flag that enables INIT to retain a specific CRL.

  On returning from the exit, PSF frees the storage for each CRL unless XTP7CRLF is set. CRL flags are supplied for all exit calls; however, only at the INIT call can the CRL storage be freed. See Figure 69 on page 280.

Note: To use PSF storage efficiently, retain CRLs only if you need them. If you are using the ENQ or DEQ system macros with selected partitioned data sets, you can store information, such as VOLSERs, in the exit work buffer, ECAWKBUF, so that PSF can free CRL storage.

Begin-Data-Set Call (BDSC)

The begin-data-set call (BDSC), which is optional, is controlled by the XTP7EBDS flag. This call is made before the processing of the data set begins—that is, before any separator-page exits begin. The begin-data-set call is made for all new data sets and for data sets that are restarted after an interrupt. Fields in the Exit Communications Area APSUECA and its extension APSUECE, which describe the data set (that is, the user's JCL), have not yet been initialized. You can use this call to do the following:

- Control ETFF. See "Considerations on Eject-to-Front-Facing Function" on page 198.
- Request termination of the data set (XTP7TDS). See "Data-Set Termination Considerations" on page 200.
- Modify XTP7RRLV and XTP7NOTY. For XTP7RRLV, see "Partial Control of Resource Deletion" on page 211. For XTP7NOTY, see "Initialization Call (INIT)" on page 203.
- Control the marking and offset stacking of the interrupt message page. See "Considerations on Forms Marking of an Interrupt Message Page" on page 199.
- Control the setting of the media origin to the 3800 media origin. See "3800-Compatible Media Origin Considerations" on page 196.
- Control converting a multiple-celled IM1 image. See "Considerations on Converting a Multiple-Celled IM1 Image" on page 197.
- Control the page checking for Cut-Sheet Emulation mode. See "Considerations on Cut-Sheet Emulation Mode" on page 198.

Input and Output Indicators for BDSC

For a complete cross-reference of the XTP7 and RLST fields used as input and output, see Table 19 on page 302.

Programming Considerations for BDSC

None.

Resource-Access Call (RAC)

The resource-access call (RAC), which is optional, is controlled by the XTP7NACC flags. This exit call, which is selected on the basis of resource type, occurs when PSF determines that it needs a resource in order to continue printing. The resource access call is made once for each resource in each map, any time the map is processed. You can use this call to do the following:

- Request termination of the data set (XTP7TDS). For information on messages issued when data sets are terminated, see "Data-Set Termination Considerations" on page 200.

  Note: Terminating a data set overrides other processing.

- Substitute a resource by changing the resource name in the RLSTNAME field of the APSURLST control block. Setting the RLSTSMSG flag in the RLSTAFLG field causes message APS563I to be issued when a resource is substituted. Text can be inserted and is pointed to by RLSTMSGP. The insert length is contained in RLSTMSGL. The maximum length is 64 bytes.

  If you want to substitute for a resource, you must substitute the resource name each time the map is processed. For example, the exit must substitute a page segment name at the Map Page Segment (MPS) resource access call and the Include Page Segment (IPS) resource access call.

  Notes:

  1. If the exit requests a substitution message (APS563I) each time the resource access call is invoked for the resource, multiple substitution messages can be issued for a resource.
  2. No substitution can be made for inline or user library resources.
  3. If APSUX07 requests substitution, and PSF determines that the request should be honored, PSF attempts to retrieve the substituted resource only from the library from which the original resource would have been retrieved.

- Force a load or reload[17] of the resource by setting the RLSTLOAD flag in the RLSTAFLG field of the APSURLST control block.

  – If the resource has not been loaded, the action is ignored.
  – If the resource has been loaded, the resource is deleted and loaded from the library.
  – When processing certain map structured fields, PSF cannot reload the resource. For example, when the exit is called for the Map Page Segment (MPS), the page segment can be reloaded. When the exit is called for the Include Page Segment (IPS), PSF is not in a state in which it can reload the page segment.

  When unable to reload the resource, PSF sets RLSTRRI (resource-reload-ignored) ON before calling the exit. PSF ignores any request to reload the resource; no error occurs. PSF continues processing the resource as though the reload had not been requested.

  Note: If a substitution is made and a load is requested, the substituted resource is loaded.

- Mark a resource as a priority-level resource (PLR). For information on PLRs, see “Partial Control of Resource Deletion” on page 211.

- Modify the XTP7NOTY and XTP7RRLV fields. For information on XTP7NOTY, see “Initialization Call (INIT)” on page 203. For information on XTP7RRLV, see “Partial Control of Resource Deletion” on page 211.

[17] To have the exit request that a resource be “reloaded,” set RLSTLOAD to 1. PSF does not use an existing version of the resource, but loads the resource from the host library.

Input and Output Indicators for RAC

For a complete cross-reference of the XTP7 and RLST fields used as input and output, see Table 19 on page 302.

Programming Considerations for RAC

- APSUX07 is called to process a mapped resource each time PSF encounters a structured field for the resource:

  - Map medium overlay (MMO) structured field
  - Map page segment (MPS) structured field
  - Map coded font (MCF) structured field
  - Include page segment (IPS) structured field
  - Include page overlay (IPO) structured field
  - Map page overlay (MPO) structured field

  Even if the resource containing the structured field remains loaded, subsequent references to that structured field result in calls to APSUX07.

  Substitutions made to resources in that structured field do not remain in effect. Each time the resource is referenced, PSF calls the exit to give it an opportunity to request substitution.

  Note: Because the exit is called each time the resource is mapped, APSUX07 can request the substitution message (APS563I) for each occurrence of a substitution. As a result, if a resource is mapped many times in a data set, two or more substitution messages may possibly be issued.

- The resource exit cannot substitute inline resources. PSF ignores any request to substitute an inline resource.
- The exit cannot request that an inline resource be reloaded.
- The resource exit cannot substitute a user library resource name for another resource. PSF ignores an exit request to substitute a user library resource.
- Only system resources can be substituted. PSF forces the substituted resource to be loaded from the originating library. For example, when a system resource is substituted, PSF searches only in the system library for the new substituted resource, even if the substituted resource resides in the user library.
- The exit can request that a user library resource be reloaded.
- Other programming considerations apply for resources of the following types:

  - Overlays
  - Page segments
  - Fonts
  - Page definitions
  - Form definitions

RAC and Overlays

Overlays are mapped resources contained in a map medium overlay (MMO) structured field or a map page overlay (MPO) structured field. For PSF to print an overlay mapped in an MPO, the page must contain an include page overlay (IPO) structured field. Each time PSF encounters an MMO, an MPO, or a IPO structured field, it checks to see if APSUX07 is active for overlays at access time. If it is active, PSF builds a chain of resource-list control blocks (RLSTs) containing one entry for each overlay in the MMO field, the MPO field, or the IPO field. (For more information about structured fields, refer to *Mixed Object Document Content*

Architecture Reference.) The exit is called once, and PSF processes the RLST changes when it regains control from the exit.

RAC and Page Segments

Hard page segments (segments loaded in the printer) are found in a map page segment (MPS) structured field and are processed like overlays. Soft page segments (page segments sent inline with data) are not found in an MPS structured field, and are processed differently. You should be aware that:

- APSUX07 allows retention of hard page segments over data set boundaries.

  If a hard page segment is loaded from a previous data set, it need not be loaded again.

  If a job calls for a soft page segment, and that page segment is loaded from an earlier data set in that job, PSF treats the soft page segment as if it were a hard page segment.
- When soft page segments are declared, the reload option is ignored. For more information on reloading resources, see page 206.
- PSF calls the exit each time PSF encounters an MPS or Include Page Segment (IPS) structured field.

For more information about page segment structure, refer to *Mixed Object Document Content Architecture Reference.*

RAC and Fonts

Fonts are resources specified in a Map Coded Font (MCF) structured field. A font can be called out by a coded font name, a code-page–character-set pair, or a GRID. An MCF structured field can contain single-entry fonts or multiple-entry fonts. Each time PSF encounters an MCF structured field, it checks to see whether APSUX07 is active for fonts at access time. If APSUX07 is active, PSF builds a chain of resource-list control blocks (RLSTs) containing one entry for each single-entry font in the MCF field.

For dynamically defined fonts (those with code-page–character-set pairing), PSF builds an RLST font-extension area. PSF stores the names of the code-page–character-set pairs in this area.

A font cannot always be represented by a single entry in the MCF structured field. Raster double-byte fonts that are dynamically defined can contain up to 190 entries. To represent these double-byte fonts, PSF builds an RLST entry for each font section. This causes the number of RLST entries in XTP7LSTC to be greater than the number of fonts mapped in XTP7RESC.

A coded font from the system library can be substituted, even though the code page or character set may reside in the user library.

During a coded font access call, PSF does not yet know the names of the code page and the character set. Therefore, PSF does not know from which library the code page and character set will be loaded. RLSTUSER and RLSTINL are set to the status of the coded font only.

Restrictions

- An inline font cannot be substituted or reloaded into the printer.
- When a font is being substituted, the new font type must be the same as the old font type. For example, a coded font can be substituted only for another coded font. Similarly, a code-page–character-set pair can be substituted only for another code-page–character-set pair, and a GRID must be substituted for a GRID.
- The font is considered an inline font if, during an access call for a code-page–character-set pair, any part of the pair is inline. If no part of the pair is inline, but any part of the pair is from the user library, the font is considered a user library font. In either case, even if one part is from the system library, neither the code page nor the character set name can be substituted.
- Single-byte and double-byte fonts are not interchangeable.

  Note: PSF does not differentiate between single-byte and double-byte raster fonts if both are specified as coded fonts.
- For double-byte font character sets or GRID names, you cannot change font technologies. A raster font must be substituted for a raster font, or an outline font for an outline font. For coded fonts, you can change technologies.

For more information on coded fonts, code-page–character-set pairs, single-byte fonts, double-byte fonts, and the MCF structured field, refer to *Advanced Function Printing: Host Font Data Stream Reference.*

RAC, Page Definitions, and Form Definitions

Page definitions and form definitions are not loaded in the printer and are not mapped in a structured field. However, they are treated like overlays, page segments, and fonts, with one exception: the RLST chain contains only one entry for each type of resource.

Using RAC in Processing Page Definitions and Form Definitions

For an explanation of page definition and form definition types, see “Considerations Related to Page Definitions and Form Definitions” on page 199.

Default and Auxiliary Processing

Default and auxiliary page definitions and form definitions are processed in two ways:

- During the initialization of PSF,
  - XTP7AUX is set to indicate that an auxiliary resource is being processed.
  - XTP7DSAT flags are set to indicate whether the page definition or form definition is being used as a system default or as an auxiliary resource.
  - XTP7TDS (terminate data set) is ignored.
  - Reload functions are ignored.
  - Substitutions of page definitions or form definitions can occur. The page definition or form definition to be used for the FSA session can be changed in the RAC.

- While PSF is in use,
  - If no page definition or form definition is specified in the user data set, a default is used.
  - A substitution for the default page definitions or form definitions is valid only for the current data set.
  - If a reload function is specified, the new default is used from that point on. For more information on reloading resources, see page 206.

User-specified processing
A user can specify page definitions and form definitions in the OUTPUT JCL statement. All the functions listed under “Resource-Access Call (RAC)” on page 205 are then supported.

Inline processing
Inline page definitions and form definitions are contained in the print data set. For more information about using inline resources as printing resources, refer to *Print Services Facility/MVS: Application Programming Guide*. When inline resources are used, all the functions listed under “Resource-Access Call (RAC)” on page 205 are supported; however, no substitution for inline resources can be made.

Resource-Load Call (RLC)

The resource-load call (RLC) occurs just before and just after a resource is loaded from either a system library or a user library. This call consists of two subcalls: the load-begin subcall and the load-end subcall. These calls are optional and are controlled by the XTP7NLD field. This exit call is selected on the basis of resource type.

The timing of the load-begin and load-end subcalls enables the exit to request or release control of serially reusable resources in the resource library by use of ENQ or DEQ.[18]

For all resource-load calls:

- PSF supplies a partitioned data set concatenation number in the RLSTPDS, in the RLSTCPDS (code page), and in the RLSTFNDS (font character set).

  Note: For restrictions, see “Programming Considerations for RLC” on page 211.

- APSUX07 can modify the RLSTPLR flag for the resource being loaded, and it can modify XTP7NOTY and XTP7RRLV. For more information on XTP7NOTY, see “Initialization Call (INIT)” on page 203; for more information on XTP7RRLV, see “Partial Control of Resource Deletion” on page 211.
- Request termination of the data set (XTP7TDS). For information on issuing messages when terminating data sets, see page “Data-Set Termination Considerations” on page 200.

[18] ENQ and DEQ are MVS system macros. For more information on system macros, refer to the macro publication for your version of MVS.

Input and Output Indicators for RLC

For a complete cross-reference of the XTP7 and RLST fields used as input and output, see Table 19 on page 302.

Programming Considerations for RLC

- PSF cannot ensure that the ENQ and DEQ macros will be used by multiple users of a resource library. Each installation is responsible for the use of these macros.

  Note: You should not run the system ENQ macro for a member and then request the termination of the data set.

- Because RLC processes only one resource at a time, only one RLST entry exists at load time.
- PSF supplies the partitioned data set concatenation number. This number is *n*–1, where *n* is the search-order position of the partitioned data set in the system library or the user library.
- When loading single-byte fonts, PSF can access the library several times. When loading double-byte fonts, PSF can access the library 2*n*+1 times (where *n* is the number of sections in the double-byte font).
- At load time, special flags indicate whether the font name specified is a code page, a character set, or a coded font name.
- For double-byte fonts, the section ID is also provided.

Resource-Deletion Call (RDC)

The resource-deletion call (RDC) controls the deletion of resources at data set end (DSE). This call, which is optional, is controlled by the XTP7NDEL field. This exit call is selected on the basis of resource type.

PSF deletes resources by applying a least-recently-used algorithm. PSF uses this method to maintain the number of resources loaded, but you can use APSUX07 to take full or partial control of resource deletions. The following sections, “Partial Control of Resource Deletion” and “Full Control of Resource Deletion” on page 212, describe the PSF deletion algorithm.

Input and Output Indicators for RDC

For a complete cross-reference of the XTP7 and RLST fields used as input and output, see Table 19 on page 302.

Partial Control of Resource Deletion

You can modify the least-recently-used PSF algorithm in two ways:

- To modify the reasonable resource loading values (RRLV). See Table 20 on page 307 for a listing of the RRLV default values, which can be modified.

  PSF uses the RRLV to keep the number of resources loaded (in a printer or in virtual storage) at a manageable level. One RRLV exists for each resource type. PSF supplies the current RRLVs at every call to APSUX07.

  RRLVs have no upper or lower limit, except that for overlays and object containers RRLV is always 0. (For retention of overlays at DSE, see “Full Control of Resource Deletion” on page 212.) However, setting the RRLVs too high can cause a resource-constrained condition, in which the printer does not have enough storage for the resources required to print the current page. Setting the RRLVs too low can lead to additional downloading of resources.

Note: For retention of resources other than auxiliary resources at DSE, the RRLV must be larger than the number of auxiliary resources. For more information on auxiliary resources, see "Considerations Related to Page Definitions and Form Definitions" on page 199.

The one exception is that with overlays, you must have full control of resource deletion. For more information, see "Full Control of Resource Deletion."

- To use priority-level resources (PLRs)

  You can modify the least-recently-used algorithm by marking resources as PLRs. When PSF is using RRLV in deleting resources, it deletes non-PLRs before any PLRs. A frequently used resource can be marked as a PLR; this improves its chances of being retained at DSE.

  Note: Some PLRs might not be retained. The RRLV must be large enough to include all auxiliary resources before any others can be kept.

Full Control of Resource Deletion

You can take full control over resources kept over data set boundaries by setting the XTP7NDEL flag for the resource.

In building an RLST, PSF uses one entry for each resource currently loaded. The resource keep or delete flag (RLSTKEEP) is set according to the PSF deletion algorithm described in "Partial Control of Resource Deletion" on page 211.

You can take full control of resources kept or deleted by modifying RLSTKEEP. On return from APSUX07, PSF checks the RLST chain and deletes resources whose RLSTKEEP flags are turned off. The number of resources kept is not limited to the RRLV.

If APSUX07 does not modify the RLSTKEEP flags, PSF deletes resources on the basis of the least-recently-used algorithm.

Note: PSF tries to save all fonts and page segments that are identified as nested resources. PSF identifies overlays with nested resources by setting the RLSTDEP flag at delete time. PSF also sets the RLSTNEST flag whenever a nested resource is referenced.

Some restrictions apply, however. For more information, see "Using RDC to Control Resources."

Using RDC to Control Resources

Keep the following considerations in mind when using the resource-deletion call (RDC) to control resources:

- PSF never deletes auxiliary page definitions, auxiliary form definitions, or auxiliary fonts. When the RRLV for page segments is zero (the default), however, PSF deletes auxiliary page segments.
- Unless the printer does not support selective deletion, PSF does not delete a resource that is nested in another loaded resource.

  Note: Your printer might not support the selective deletion of overlays and page segments. For printers that do not support selective deletion, if PSF is requested to delete one resource of a particular type, it deletes *all* the resources of that type, and issues the PSF message APS551I. If a loaded overlay containing fonts or page segments maps a deleted resource, the overlay is deleted. To determine what support your printer offers for selective deletion, see the publications for that printer.

- APSUX07 can modify the XTP7NOTY fields.
- APSUX07 can modify the RLSTPLR flag and XTP7RRLV fields; however, these changes do not take effect until the next DSE.
- Unless full control is active, PSF deletes all overlays from the printer during resource deletion. If overlays are retained and are later deleted, PSF might not delete the resources nested in the deleted overlays.
- RDCs are based on the resources loaded at DSE, not on the resources used by the data set. For example, if a data set did not use any fonts and the respective XTP7NDEL flag were active, the exit would still be called for fonts and would contain an RLST chain with an entry for each font loaded.
- PSF deletes all inline resources at the end of the data set, before the exit is called. (Before APAR OY67181 was applied on PSF/MVS 2.1.0 or PSF/MVS 2.1.1, inline form definitions and page definitions could be retained at the end of the data set.)
- PSF always deletes user library resources at the end of the data set, before calling the exit.
- PSF does not delete object container resources. The printer manages these resources. PSF ignores requests to delete or retain object container resources.

Termination Call (TERM)

The termination call (TERM) can be used to undo any change in processing state made during exit processing. If XTP7ETRM is active during the termination of an FSA, TERM occurs. For example, if a resource exit started the GETMAIN system macro during the INIT call, the TERM call could be used to start FREEMAIN.

In this call, you can also control whether or not PSF automatically restarts after a restartable abend occurs. See “Considerations on Restartable Abends” on page 198 for more information.

Input and Output Indicators for TERM

For a complete cross-reference of the XTP7 and RLST fields used as input and output, see Table 19 on page 302.

MVS Download Print Parameter Exit: APSUX15

Using the APSUX15 exit with MVS Download, you can provide additional job information (such as passthru print parameters) to be transmitted to the target MVS Download server on AIX. Refer to *Print Services Facility/MVS: MVS Download Guide* for a description of this exit.

Direct-Print Exit: APSUX16

APSUX16 is the exit given control by PSF in direct-printing mode. The exit is invoked for the following calls:

- Begin-data-set call
- Message call
- End-data-set call

Begin-Data-Set Call (BDSC)

PSF invokes APSUX16 at the beginning of the data set. When APSUX16 receives control, no processing has occurred.

To request forms marking for data set separation, use XTP16MRK. This is the only call that can be used to request forms marking.

If the bit labeled XTP16MRK is set in the APSUXP16 control block, PSF sends the Mark Form order of the Execute Order Anystate command to the printer on data set boundaries.

Use XTP16RPC to request that PSF verify that the received page counter has been incremented by 1 for each page that has been sent to the printer. This is the only call that can be used to verify the received page counter. If the bit labeled XTP16RPC is set in the APSUXP16 control block, PSF abends unless the received page counter has been incremented by 1 for each page that has been sent to the printer. If PSF is attached to a 3800 printer, this bit is ignored.

Warning: For reasons of performance, some channel extenders do not increment the received page counter. Check with the vendor of your OEM channel extender before setting XTP16RPC.

Input for APSUX16 Begin-Data-Set Call

When this exit is entered, the contents of the registers are as follows:

Register 1	Pointer to the address of parameter area APSGEXTP
Register 13	Pointer to the caller's save area
Register 14	Return address
Register 15	Entry-point address

XTP16SPI	Stacked page ID
XTP16DCB	DCB address of the application program
XTP16DEV	Unit address of printer
XTP16ETY	Call type

Output for APSUX16 Begin-Data-Set Call

XTP16MRK	Forms mark flag
XTP16RPC	Received page counter flag

Message Call for APSUX16

PSF invokes APSUX16 before issuing most FSA-generated messages (APS*nnn*I and APS*nnn*A).

Use this exit call, with the XTP16ACT flag, to suppress or redirect messages. Messages can be redirected to the operator console, the security console, or both. If a message is intended for the operator or the security console, however, the exit cannot redirect it to be printed with the data set.

Input for APSUX16 Message Call

When the PSF direct-print exit is entered, the contents of the registers are as follows:

Register 1 Pointer to the address of parameter area APSGEXTP
Register 13 Pointer to the caller's save area
Register 14 Return address
Register 15 Entry-point address

XTP16MSG Message number (APS*nnn*I or APS*nnn*A)
XTP16ACT Message action to be taken by PSF:
- **XTP16WTO** Put the message to the operator console.
- **XTP16SEC** Put the message to the security console.
- **XTP16PAP** Print the message with the data set.

XTP16SPI Stacked page ID
XTP16DCB DCB address of the application program
XTP16DEV Unit address of the printer
XTP16ETY Call type

Output for APSUX16 Message Call

XTP16ACT Message action requested by the exit. The message can be redirected to any or all of the destinations by setting the correct combination of flags.
- **XTP16WTO** Put the message to the operator console.
- **XTP16SEC** Put the message to the security console.
- **XTP16PAP** Print the message with the data set.

Note: The exit cannot request that a message be printed with the data set unless that was the original destination set by PSF.

If the exit sets XTP16ACT to 0, PSF suppresses the message.

End-Data-Set Call (EDSC)

APSUX16 is invoked at the end of each data set—that is, after the data set is printed.

Input for APSUX16 End-Data-Set Call

Register 1 Pointer to the address of parameter area APSGEXTP
Register 13 Pointer to the caller's save area
Register 14 Return address
Register 15 Entry-point address

XTP16SPI Stacked page ID
XTP16DCB DCB address of the application program

XTP16DEV	Unit address of printer
XTP16ETY	Call type

Output for APSUX16 End-Data-Set Call

None.

Programming Considerations for APSUX16

The following general considerations apply to APSUX16:

- When this routine is entered, register 1 points to a one-word parameter list that contains the address of APSGEXTP. APSGEXTP contains a pointer to the exit-communications area, APSUECA, and a pointer to the exit 16 parameter list, APSUXP16.
- The following fields in APSUECA and APSUECE are valid when exit 16 is called:

ECAECEP	APSUECE address
ECAUCOMP	APSUCOM address
ECAUSAVE	Save area
ECAWKBUF	Exit work buffer

ECEDFDEF	FORMDEF from PRINTDEV
ECEDPDEF	PAGEDEF from PRINTDEV
ECEDPF	DPF is attached
ECEIMPS	Impression count
ECEJOBIM	Job impression count
ECEJOBLP	Job logical page count
ECEJOBPP	Job physical page count
ECELPCNT	Logical page count
ECEPPCNT	Physical page count
ECEPRTDT	Device type from OPC
ECEPRTM	Device model
ECEPRTMD	Device model from OPC
ECEPRTT	Device type
ECEPSFD	PSF Direct is attached
ECEREC	Record count
ECERPM	RPM is attached

- The work buffer (ECAWKBUF) in the exit-communications area (APSUECA) can be used as a work area by the exit. PSF does not clear this area between calls; so the exit can store data here between exit calls for one data set.
- The Begin-Data-Set and Message calls are invoked from the FSA subtask. The End-Data-Set call is invoked from the NST subtask.

  Because the APSUX16 calls are invoked from different subtasks, two ECAs are created for each printer:

  – ECEFECAP is the ECA address that is passed to the Begin-Data-Set and Message calls.

  – ECENECAP is the ECA address that is passed to the End-Data-Set call.
- When register 13 is entered, it points to a standard 18-word save area that should be used to save the caller's register.
- Use register 12 as a base register for all exit routines. If register 12 points to the beginning of an exit routine, and the module name begins in the sixth byte

of the routine, the module name will appear on the SDUMP title page if an abend occurs.

- The page and record counts—ECEIMPS, ECEJOBIM, ECEJOBLP, ECEJOBPP, ECELPCNT, ECEPPCNT, and ECEREC—are counts not of the number of pages printed, but of the pages and records that PSF has processed. PSF may have processed many pages and sent them to the printer, although the printer has not yet printed them.

  The stacked page ID, XTP16SPI, is the ID of the page at the stacker. As each page reaches the stacker, the stacked page ID is incremented. At the beginning of a data set, the stacked page ID contains either the last stacked page ID, or zero. To determine the number of pages stacked, calculate the difference between the end-data-set call stacked page ID and the begin-data-set call stacked page ID.

- The page counts—ECEIMPS, ECEJOBIM, ECEJOBLP, ECEJOBPP, ECELPCNT, and ECEPPCNT—include the pages on which messages are printed at the end of the data set.

  The record count (ECEREC) is the number of records received by PSF from the application.

- Execution is *not* transferred to the exit in any of the following circumstances:

  – If one of the following messages is issued:

    ```
    APS000I  APS006I  APS043I  APS049I  APS055I  APS066I  APS551I  APS955I
    APS001I  APS022I  APS044I  APS050I  APS060I  APS067I  APS605I  APS957I
    APS002A  APS023I  APS045I  APS051I  APS062I  APS068I  APS607I
    APS003A  APS025I  APS046I  APS052I  APS063I  APS226I  APS608I
    APS004I  APS038I  APS047I  APS053I  APS064I  APS516I  APS610I
    APS005I  APS042I  APS048I  APS054I  APS065I  APS517I  APS922I
    ```

  – In response to a message generated by any of the following:

    - The APSRMARK utility—messages APS580I to APS599I
    - The APSRFLU utility—messages APS1700I to APS1711I
    - The operator interface—messages APS620I to APS699I, with the exception of message APS691I
    - The trace formatter utility, APSTRFMT—messages APS960I to APS979I

- The exit can change the message action (XTP16ACT), as follows:

  – If the exit does not alter XTP16ACT, PSF takes the action originally intended for the message.

  – If the exit sets XTP16ACT to 0, PSF suppresses the message.

  – If the exit sets XTP16ACT to one or more actions, PSF takes those actions.

- If XTP16ACT suppresses a message, PSF still traces the message.

- The following restrictions apply to XPT16ACT:

  – Action messages (noted by the format APS*nnn*A) cannot be suppressed. If the exit requests that an action message be suppressed, PSF ignores the request and sends the action message to the operator.

- The exit cannot request that a message be printed with the data set (XTP16PAP), unless that was the original destination set by PSF.
- PSF issues some messages twice, once to the operator and once to be printed with the data set. When a message is to be redirected, you might want to code a check for the same message, to avoid duplicate messages.

Chapter 18. Understanding AFP Performance

This chapter presents the concepts of Advanced Function Presentation (AFP) performance. You can use this information in tuning the software and hardware to maximize the printer throughput with minimum effect on other applications running on the host system.

For information about printer throughput rates (given in pages per minute, lines per minute, impressions per minute, and so on), refer to *AFP: Printer Information*.

Advanced Function Presentation (AFP) is an integration of high-function printers with printer software. PSF is the software that runs on the host, preparing data and sending it to a printer, managing AFP resources both within the PSF address space and in the printer, and performing error handling and error recovery. To understand AFP performance, consider the following:

- Characteristics of the printer, and total number of printers installed
- Density and complexity of the print job or data stream sent to the printer
- Data rate at which the attachment hardware can send data to the printer
- Ability of the host processor to process and send data to the printer

The Printer

Printer performance, or throughput of the printer, is usually stated in impressions per minute (ipm). It depends on several factors, including:

- Mechanical speed
- Internal processing speed
- Paper size
- Printer options

Duplexed pages contain two impressions, one on each side of a sheet of paper.

Mechanical Speed

The mechanical speed of the printer is defined as the speed at which paper moves through the paper path. Mechanical speed, usually measured in impressions per minute (ipm), determines the rated (maximum) speed of the printer. The mechanical speed of a continuous-forms printer depends in part on the size of the paper being used.

Internal Processing Speed

A printer receives the data stream from the host processor. The internal processing speed of the control unit in the printer limits the amount of data the printer can process per unit of time. If the processing requirements of the print-job data stream are greater than the capabilities of the control unit in the printer, the printer throughput will be slower than its mechanical speed. If this happens on a cut-sheet printer, the printer stops picking input sheets for one or more cycles, until it is ready to send another page to its image generator. If this happens on a continuous-forms printer, the printer stops moving the form for a number of cycles, until it is ready to send another page to its image generator. In either case, the printer is running below its mechanical speed. See “Data Density and Complexity” on page 220.

Paper Size

Throughput in impressions per minute (ipm) can be faster for smaller paper sizes than for larger paper sizes. For example, the throughput of a 3835 printer is 88 ipm for 12-by-8.5 inch pages, but is 68 ipm for 14.875-by-11 inch pages.

Printer Options

In case print jobs have unusual performance requirements, familiarize yourself with printer features that could affect throughput. Some examples follow.

Raster Storage: If a print job contains large amounts of image data, or uses large or multiple fonts, it can cause a shortage of raster-pattern storage that would affect printer throughput.

Printer Features and RPQs: Some printers can be equipped with performance features or custom features, requests for price quotation (RPQs) that improve the throughput for dense or complex print jobs.

Duplex Printing: Some cut-sheet printers can print on both sides of the paper. Duplex printing does not necessarily affect the throughput of the printer. However, frequent changes from simplex to duplex or from duplex to simplex can noticeably reduce throughput.

For additional information about AFP printers, RPQs, and performance, refer to *AFP: Printer Information*.

The Print Job

Printer performance can be limited by such characteristics of the print job as:

- Data density and complexity
- Characteristics of the print data sets
- Number of resources
- Size of the print job

These characteristics affect the use of host resources, storage, DASD, device controller use, channel use, and the host processor.

Data Density and Complexity

The data stream sent to the printer consists of the following:

- Print data—text and images that print on the page
- Control information—the data defining the placement and appearance of the printed output

Printer throughput is affected by the amount of print data and control information. In general, bytes of print data require less processing than bytes of control information, and bytes of image data can be processed faster than bytes of text data; however, image data contains far more bytes than text data.

PSF generally uses more processor cycles and more DASD activity as the density and complexity of the print job increase. This increase continues until the limit of the control unit in the printer is reached, at which time the processor use becomes constant.

Characteristics of Print Data Sets

Print data sets can contain line data, data already composed into pages (MO:DCA-P data), image data, graphics data, and bar code data. The data can be simple enough to require little processing by PSF or the printer, or complex enough to require more processing.

The characteristics of the print job can affect printer throughput. If the data streams for jobs have different proportions of control bytes to data bytes, a given printer will handle them at different maximum rates. The following descriptions of the characteristics of print data sets indicate how the characteristics of print jobs can affect host-processor utilization, attachment limitations, and printer throughput.

Formatted line data uses few fonts, and formatting is done with a page definition. An example is a one-up or multiple-up computer listing. With the same number of characters per page, formatted line data requires more processing by PSF than does simple MO:DCA-P data, because each line is an independent element that must be positioned.

Unformatted line data uses several fonts, and formatting is done with a more complex page definition. An example is a billing application. With the same number of characters on a page, unformatted line data requires more processing by PSF than formatted line data, because parts of each line are independent elements that must be positioned.

Simple MO:DCA-P data uses few fonts, no justification, no tabulation, and no formatting. An example is a simple business letter. Simple MO:DCA-P data requires less processor utilization than line data, if both have the same number of characters per page. Little processing by the printer is required unless the pages contain large numbers of characters.

Complex MO:DCA-P data uses formatting, line and column justification, and multiple fonts. An example is in-house publishing with illustrations and tables that consist of formatted text and require several control bytes per text character. Image data is likely to be included. Because of the many controls in the data stream, complex MO:DCA-P data requires more processing by the printer, and is more likely to affect printer throughput, than simple MO:DCA-P data.

Simple image data uses a single raster pattern to define a simple image. An example is an uncompressed scanned photograph or a bank check. Simple images require less processing by the printer and PSF for each raster byte than complex images. Because simple image data consists of 7,200 bytes for each square inch of image at 240-pel density, higher data rates and processor utilizations may be required.

Complex image data consists of several independent raster patterns, or single raster patterns that have been compressed. Examples are graphic objects that have been converted to images, and compressed scanned images. Defining each raster pattern requires control bytes that must be processed by PSF or by the printer.

Number and Type of Resources

PSF manages the resources for the print data set. For example, PSF loads fonts, page segments, and overlays during the processing of a print data set, and deletes page segments and overlays once the processing is completed. PSF retains a number of fonts in the printer between print data sets, depending upon the printer model.

Some print data sets require many resources that need resource-management processing. Management of the resources can increase processor utilization and decrease printer throughput. For example, if PSF tries to load resources, and the printer runs out of raster pattern storage, PSF deletes the least recently used resources and tries again to load the new resources. If this action does not free enough raster pattern storage, PSF deletes all the resources in the printer and loads only those required by the current pages it is processing.

Resource management can reduce printer throughput in two ways: by using time during which no printing can be done, and by increasing the amount of processing required to print a data set.

Inline Resources

The use of inline resources may adversely affect performance when an error makes it necessary for PSF to reposition, because the inline resources have to be recached. This adverse effect may arise more frequently when the print job contains many inline resources, when the resources are large, or when PSF repositions frequently.

Resident Fonts

Some printers support resident fonts. Selecting resident fonts can improve some aspects of performance. For example, printer throughput can be increased if fonts do not have to be sent to the printer. To learn which AFP printers support resident fonts, refer to *AFP: Printer Information*.

Font Pruning of Raster Fonts

PSF optionally prunes raster fonts to reduce the amount of font data sent to the printer. This means that PSF does not send all the characters in a character set when downloading a font. Instead, PSF matches the characters in the character set to the code points in the code page, and sends to the printer only the characters that are referenced in the code page. Font pruning can save time needed to send fonts to a printer, and can reduce the amount of raster pattern storage used in the printer.

PSF does not prune outline fonts.

Font pruning is especially useful when the number of characters in the character set exceeds the number of code points in the code page. However, font pruning can increase processor use.

Sometimes font pruning does not improve performance—for example, when the font character sets match the code pages being used.

For information about how to deactivate font pruning, see the XTP7PRNS and XTP7PRND fields described on page 287.

Note: You can deactivate the pruning of double-byte fonts without deactivating the pruning of single-byte fonts.

Captured Fonts

Some printers, such as the 3130 printer, can capture downloaded fonts. Captured fonts automatically become new temporary printer-resident fonts. This improves performance on future jobs that use the same fonts. for more information, see Chapter 14, “Marking Fonts for Font Capture” on page 141.

Size of Print Jobs

Printing several short print jobs requires that job-initialization code and job-termination code be invoked frequently. Therefore, short jobs require more processor power than longer jobs, and can affect printer throughput.

Using Outline Fonts

The character set for an outline font is generally much smaller than its equivalent for a raster font, and therefore requires less downloading of data. Further, coded fonts for outline fonts of different point sizes can share the same character set in the printer, reducing the time for downloading. Also, code pages can be shared. Therefore, when raster font downloading time would affect printer throughput, the use of outline fonts may improve printer performance. For information on how to use APSUX07 to automatically map raster fonts to outline fonts (for printers that support outline fonts), see “Considerations Related to Outline Fonts” on page 202 and “Considerations Related to Mapping to Outline Fonts” on page 202. This mapping will slightly increase PSF's use of the central processor.

When you request that an outline font be reloaded, you may experience some performance degradation if several coded fonts are using the same code page or character set. All the fonts sharing the code page or character set will be deactivated and then reactivated, all from the same reload request. (Deactivating and reactivating an outline coded font does not make it necessary to send all the font data to the printer again; only the commands necessary to associate the code page and the character set are sent.)

When Exit 7 deletes fonts, it deletes them at the coded font level; that is, Exit 7 deletes both the code page and the character set. Because different outline coded fonts may use the same code pages and character sets, other instances of these font members will be deleted (and made unusable) at the same time.

Transmission and Compression of Data

When PSF is driving TCP/IP-attached printers or SNA-attached printers that are communication-attached, PSF can compress line data to use communication lines more efficiently. You can activate line-data compression by specifying the optional COMPRESS keyword in the PRINTDEV statement.

PSF itself can do two types of compression. One is line-data-only compression, activated by specifying the optional COMPRESS keyword in the PRINTDEV statement. The other is Huffman compression, which can be used to compress any type of data and is activated by using Exit 7. The kind of compression you will use will depend on what kind of data you will mostly be sending to the printer and on the type of printer (SDLC-attached) you will be using. If possible, you may want to try both to see which gives better results. For more on Exit 7, see “Considerations Related to Huffman Compression” on page 203.

Line-data compression may improve performance, because PSF can compress continuous blanks and trailing blanks and can replace strings containing more than 5 blanks with Repeat String (RPS) presentation text controls. Line-data compression is most likely to improve printer throughput when the following conditions are met:

- The printer is attached with a low-speed attachment.
- Printer throughput is limited by the attachment.
- The print job contains line data. PSF does not compress print jobs containing other kinds of data.
- The line data contains a significant number of blank strings or trailing blanks containing more than 5 blanks.

Note: If the line data does not have these characteristics, the use of data compression may degrade performance. For more on blank compression, see “Blank Compression for Line Data (COMPRESS Parameter)” on page 229.

Huffman compression may improve performance, because the algorithm compresses on a byte-by-byte basis and assigns a different compression bit code to represent each indivisible character in the uncompressed data. The number of bits used to represent each uncompressed character depends on how frequently the character occurs in the data to be compressed. Characters that occur frequently are assigned short compression bit codes; characters that occur less frequently are assigned longer compression bit codes.

The Huffman compression algorithm can be used on any type of data—line data, MO:DCA-P, image, fonts, etc. Data compressed by use of the Huffman algorithm can be sent only to SDLC-attached printers that have the decompression microcode—that is, AFCCU printers; the algorithm cannot be used if the AFCCU printers are attached to DPF or PSF Direct.

Huffman compression may improve printer throughput when the following conditions are met:

- The printer is attached with a low-speed communication line—for example, a 9.2-kBs line.
- Printer throughput is limited by the attachment.
- The arrangement of the data (occurrence and position) is irregular; for example, text that has many embedded blanks will do well.

The compression ratio varies considerably. It is best when the arrangement of the data (occurrence and position) is irregular; for example, text that has many embedded blanks or trailing blanks will do well. Double-byte fonts also do well. In general, you may get a compression ratio of about 1.75 to 1, *but* the results will depend on the characteristics of the print data.

Using Huffman compression may significantly increase the number of CPU cycles that PSF requires. It may also cause PSF to become a CPU-bound process rather than an I/O-bound process. The cost effectiveness of compression depends on a balance between the availability and cost of additional host CPU cycles to perform compression and the cost of additional communications bandwidth.

For maximum data transfer, remember to set the RU size. On a clean line (a line on which retransmissions are infrequent), choose the largest possible RU size.

For more on Huffman compression, see “Compressing Data by Use of the Huffman Algorithm” on page 229.

The Attachment Hardware

If you install a printer with an attachment that sends data at a rate below the maximum rate of the printer, the printer can print more slowly than its mechanical speed. The attachment methods for printers include:

- A non-SNA printer can be host-connected and channel-attached.
- A printer can be local-attached.
- A printer can be communication-attached.
- A printer can be TCP/IP-attached.

Host-Connected, Channel-Attached Printer

Channels transfer data fast enough that printer throughput is not usually limited. When configuring a system, consider the total data rate requirements of all the devices attached to a channel.

Specify that the channel adapter is to be set in high-speed data-streaming mode if possible. To set this mode, refer to the appropriate processor and printer publications. The use of other modes can affect printer throughput, in ways that depend on the printer, the application, and the length of the printer channel cable.

With the ESCON card installed in a printer, the printer can be attached to the host with an ESCON channel, which can send and receive data faster than a parallel System/390 channel. For information about printers that support the ESCON card, refer to *AFP: Printer Information*.

Local-Attached Printer

For a description of attachment methods, see “Local-Attached Devices” on page 17.

The capacity of a control unit can limit the throughput of the attached printers. When PSF uses VTAM, more processor utilization is required than with channel attachment.

Communication-Attached Printer

For a description of attachment methods, see “Communication-Attached Devices” on page 18.

The speed of the communication line can be the most important performance factor for a communication-attached printer. If multiple printers are attached to the same communication line or network path, the total capability of the network is shared and can limit printer throughput.

When PSF uses VTAM, more processor utilization is required than with channel attachment.

Table 16 on page 226 shows the approximate maximum data rates that can be reached with the attachments listed. The rates shown are not always achievable. The results depend on network and system loading, tuning parameters, and the characters of the printer data stream. The attachment can limit printer throughput.

Table 16. Approximate Maximum Attachment Data Rates

Communication line speed	**9600 bits per second**	**19200 bits per second**	**56000 bits per second**
Maximum printer data rate	1000 bytes per second	2200 bytes per second	6700 bytes per second

PSF Direct or Distributed Print Function (DPF)

Printers can be remotely attached with the Distributed Print Function (DPF) of PSF for OS/2 or PSF Direct. PSF can send the print data set to PSF Direct or DPF over an SDLC line, over a local token ring, or over a remote token ring.

| The Huffman compression algorithm cannot be used if the AFCCU printer is
| attached to DPF or PSF Direct.

PSF Direct: In a PSF Direct configuration for PSF/2 or PSF for AIX, PSF sends the print data set to PSF Direct, which passes the data set to a printer attached to PSF/2 or PSF for AIX. PSF Direct does not support remote resource caching.

When installing and configuring PSF Direct, consider the following:

- Data rate capability of the printer
- Data rate capability of the host attachment
- Data rate requirements of the print data set

DPF: DPF works with Resource Saving either OFF or ON.

In a DPF configuration, PSF sends the print data set to DPF, to be spooled and later printed. DPF requires only one PS/2. The PS/2 spools the print data sets in one stage and sends them to be printed in another stage.

The maximum data rates in DPF can be lower than data rates with PSF Direct for PSF/2 or PSF for AIX, depending on attachment speeds.

Size of the VTAM Request Unit (RU)

A VTAM request unit (RU) contains system data information for a communication-attached printer. An increase in VTAM RU size may increase printing rate, because processing larger (and, therefore, fewer) RUs contributes to higher efficiency.

The maximum size of the RU, however, is limited and dependent upon the limitations of hardware and software. For example, the RU size is limited by the communications controller or by the maximum buffer size allowed by the printer, whichever is smaller. Generally, use the largest RU size suitable for the printer or attachment being used.

VTAM PSNDPAC and DELAY Parameters

VTAM includes a coat-tailing algorithm designed to minimize the number of buffers VTAM processes. In intermediate-speed and high-speed communication configurations, this algorithm can affect the performance both of communication-attached printers and of DPF or PSF Direct.

- In intermediate-speed communication configurations, you can reduce or eliminate coat-tailing delays by specifying a large PSNDPAC pacing value in your VTAM logon-mode table entry.
- In high-speed communication configurations, you must eliminate coat-tailing delays in order to achieve full utilization of high-speed communication links. VTAM provides a DELAY parameter for this purpose; specify **DELAY=0**.

For more information about coat-tailing, refer to *VTAM Customization*.

NCP MAXOUT Value

MAXOUT (maximum output) is an operand in the physical unit (PU) macro instruction for the Network Control Program (NCP). The value specified for MAXOUT defines the number of units of data that the host system can send to the PU before the PU must respond. You should specify a value from 3 to 7. A value less than the maximum supported by the PU can cause delays in the communication line, and thus degrade performance. For more information, refer to *Network Control Program System Support Program Emulation Program Resource Definition Reference* for your system.

TCP/IP-Attached Printer

For a description of TCP-IP-attached printers, see Chapter 19, “Connecting a TCP/IP-Attached IPDS Printer” on page 235.

The throughput of TCP/IP-attached can be limited by the following factors:

- The configuration of your TCP/IP networks.

  The default maximum transmission unit (MTUP) size, specified on the GATEWAY statement in the TCP/IP profile on the MVS system, may slow the transmission of data to the printer. IBM recommends an MTU size of 2000. For more information, see “GATEWAY Statement” on page 238.
- A large amount of network traffic.
- Communication-line speed, if data is transmitted over telephone lines.

  Table 16 on page 226 shows the approximate maximum data rates that can be reached on telephone lines with different line speeds. The rates shown are not always achievable; they depend on network and system loading, tuning parameters, and the characteristics of the printer data stream.

Also, when PSF sends data to TCP/IP-attached printers, more processor utilization is required than for channel-attached printers.

The Host System

The following factors related to the host system can affect printer throughput:

- Processing capability
- Amount of real storage available
- Capacity of the I/O subsystem
- System tuning

Processing Capability

The capability of the host processor can affect printer throughput. The greater the density or complexity of a print job on higher-speed printers, the greater the processing capability required. A configuration with inadequate processor power, or with a large load other than printing, can limit throughput for dense or complex applications being printed on a high-speed printer.

For print jobs, the processor requires time to do each of the following tasks:

- Read the data from auxiliary storage
- Convert the data to a printer data stream
- Send the data to a printer
- Manage resources
- Manage checkpointing
- Prepare for error recovery
- Manage VTAM attachment protocol

The speed of the printer and the characteristics of the print data set affect processing. High-speed, channel-attached printers place a greater demand on the processor than low-speed printers.

User applications other than PSF (for example, NJE) can affect the processor capacity available for printing and can limit printer throughput.

Amount of Real Storage Available

Real storage holds PSF code, working storage used by PSF, and I/O buffers. A host-system processor that has limited real storage can cause contention for real storage and DASD that can decrease printer throughput.

Real-storage requirements may be lower for low-speed printers than for high-speed printers.

Capacity of the I/O Subsystem

The system data rate capacity and speed of the DASD devices can affect printer performance. Some print data sets require more DASD I/O from the spool or the PSF resource libraries than other print data sets.

The 3380 and 3390 DASD devices can handle large quantities of image data. Cached data on a DASD device can hold frequently accessed page segments.

System Tuning

A host system running many applications and driving printers with PSF must be tuned correctly, or balanced, to satisfy the data-rate requirements of the printers.

Blank Compression for Line Data (COMPRESS Parameter): The COMPRESS parameter provides a data compression function for line data. In line data that contains more than five contiguous blanks, PSF compresses the blanks to a single blank, plus control information for the printer to use in decompressing the blanks.

Blank compression is most likely to improve printer throughput when a printer and the jobs being printed have the following characteristics:

- The printer has a low-speed attachment.
- Printer throughput is limited by the attachment.
- The jobs are line data. Blank compression is not attempted for jobs of other types.
- The line data contains a significant number of interior strings of more than five blanks.

Blank compression can be used for host-connected, channel-attached printers, for SNA-attached printers, and for TCP/IP-attached printers. However, IBM does not recommend that you use it for host-connected, channel-attached printers, unless the attachment is the main performance bottleneck.

Note: Unless the above characteristics are present, you may see no improvement in printer throughput. Increased CPU cycles may be used to perform the blank compression, but if the line data does not contain significant interior blanks, the volume of data generated may not be decreased enough to affect the overall performance of the attachment and the printer.

For unformatted line data that contains fields of data whose placement is specified in a page definition, for example, blank compression might give little or no improvement in throughput. The data sent to the printer may contain no interior blanks, although the text on the spool contains many blanks.

The same may be true of formatted line data that has no interior strings of consecutive blanks, when JES has eliminated all trailing blanks because blank truncation was specified.

Because the COMPRESS parameter is a PRINTDEV keyword and affects all line data jobs sent to the printer, the system programmer should carefully check the characteristics of the jobs being printed, and the above listed characteristics for the attachment and printer, before implementing the new parameter.

Compressing Data by Use of the Huffman Algorithm: APSUX07 can be used to enable the compression of data by PSF before a job is sent to the printer. The data is compressed byte by byte, and each indivisible character in the uncompressed data is assigned its own compression bit code. The number of bits used to represent each uncompressed character depends on how frequently the character occurs in the data to be compressed. Characters that occur frequently are assigned short compression bit codes; characters that occur less frequently are assigned longer compression bit codes.

The Huffman compression algorithm can be used on any type of data—line data, MO:DCA-P, image, fonts, etc. Data compressed by use of the Huffman algorithm can be sent only to SDLC-attached printers that have the decompression microcode—that is, AFCCU printers; the algorithm cannot be used if the AFCCU printers are attached to DPF or PSF Direct.

Huffman compression may improve printer throughput when the following conditions are met:

- The printer is attached with a low-speed communication line, such as a 9.2-kBs line.
- Printer throughput is limited by the attachment.
- The arrangement of the data (occurrence and position) is irregular. For example, text that has many embedded blanks will do well.

The compression ratio varies considerably. It is best when the arrangement of the data (occurrence and position) is irregular, as for example in text that has many embedded blanks or trailing blanks, and in double-byte fonts. In general, you may get a compression ratio of approximately 1.75 to 1, *but* the results will depend on the characteristics of the print data.

The use of Huffman compression may significantly increase the number of CPU cycles that PSF requires. It may also make PSF an I/O-bound process rather than a CPU-bound process. The cost-effectiveness of compression depends on a balance between the availability and cost of the additional host CPU cycles required to perform compression, and the cost of additional communications bandwidth.

For maximum data transfer, remember to set the RU size. On a clean line (retransmissions are infrequent), choose the largest RU size possible.

Printer System Data Rate Requirements: The data rate to all-points-addressable (APA) printers increases with page density until the control unit in the printer is processing at its maximum rate. This data rate can be high, requiring substantial processor utilization.

Table 17 on page 231 shows the approximate system data rates that some printers require while printing different data applications that have pages containing a large amount of data. The data rate capability of high-speed printers is usually greatest for simple images. The data rates are high enough to warrant special attention when you configure and balance the host to the capabilities of one or more of these printers. Host processor utilization is related to the rate at which data is sent to the printers. For information about the characteristics of data, see “The Print Job” on page 220.

	Printer data rate (bytes per second)							
	Local-attached		Channel-attached					

Description of data	3816
	(VTAM)
	3174
		3820
		(VTAM)
		channel
			3825
			3827
			3835
				3827(A)
				3835(A)
				3828
					3800-3(S)	3835-2	3900	3900(A)

Formatted line data	3700	2200	31 000	52 000	50 000	52 000	120 000	124 000
Simple composed text	3700	2200	26 000	42 000	47 000	54 000	108 000	112 000
Complex composed text	3700	2200	13 500	20 000	24 000	65 000	64 000	65 000
Simple image	5400	5500	300 000	600 000	550 000	900 000	690 000	900 000
Complex image	5000	1500	210 000	380 000	255 000	660 000	550 000	660 000
Note: (S): With storage-expansion feature. (A): With AFIG feature.								

Table 17. Approximate Maximum Printer Data Rates

System data rates are highest for image data printed on high-speed printers. Image data can be dense, with hundreds of thousands of bytes per page, and requires high data rates. Data rates are lowest for text data printed on low-speed printers. Text data is not as dense as image data, and requires lower data rates.

Note: The use of the ESCON attachment may improve data rate, physical path length, space requirements, and overall system connectivity.

Printer Data Checks: For normal operation, use UNBLOCK in the DATACK parameter on PRINTDEV statements in the PSF startup procedure. This causes PSF to report and correct errors that can affect printer throughput in some circumstances, unless the default (BLOCK) is used for a particular job.

Buffers: The use of input (DASD) and output (printer) buffers can improve printer throughput.

The number and size of input buffers in the spooling subsystem[19] can affect the amount of resources available for a print job, and thus affect the printer throughput. As the number and size of buffers are increased to get more efficient processing, more real storage is required but less processor utilization per page is used. As print rate increases, processor utilization *per unit of time* increases.

For each printer started, PSF acquires a resource library input buffer pool. The size of the buffer pool is based on the printer model started and on the sizes of the resource libraries specified in the PSF startup procedure.

If a BUFNO parameter is specified on the DD statement of a resource library, an additional buffer pool is acquired exclusively for that resource library. This is generally not required for most printers and print jobs, but it can be considered for the page segment library when large, simple images are being printed on high-speed printers.

PSF ordinarily maintains a minimum of 6 page definitions and form definitions in virtual storage for each started printer between data sets. At maximum, PSF can maintain as many page definitions and form definitions as virtual storage can hold. You can use the resource-management exit, APSUX07, to change the values that represent the page definitions and form definitions specified in the PSF startup

[19] As specified in JES2 trackcell parameter, for example. A TRKCELL value of 5 is recommended.

procedure for the data set header and trailer pages, the messages data set, and the printer defaults. You can also change the specifications in the print data set, regardless of where they were first specified.

For channel-attached printers, the number of output buffers is controlled by the BUFNO parameter on the PRINTDEV statement. For more information, see Chapter 6, “PRINTDEV Parameters.”

Set the BUFNO value at 5 initially; for ESCON channel-attached printers and printers that support 16K I/O buffers, however, set the BUFNO value at 9. This is adequate for most applications. For large, simple images on high-speed printers, values up to 13 are recommended.

For each printer not attached to VTAM, PSF acquires a buffer pool equal to 4K times the BUFNO value in the PRINTDEV statement, plus an additional 4K. The output buffer pool is fixed in nonextended storage.

If the Huffman compression algorithm is being used, 38 000 bytes per FSA will be required in order to perform the compression.

Block Size: The physical block size of the spool files and resource libraries can affect the speed and efficiency with which the resources needed for a print data set can be accessed. Larger block sizes result in more efficient movement of data, which can reduce the load on the processor, the DASD, the device controllers, and the channels. Larger blocks, however, require more real storage.

Increasing the block size can increase processor utilization by increasing the print rate. Increasing the block size for jobs that are printing at rated speed can decrease processor utilization. The efficient use of blocks is controlled by the TRKCELL parameter in the spooling subsystem. For more information, refer to the publications for your system.

> For channel-attached printers, set the TRKCELL value to 4 or 5. For printers that are not channel-attached, do not enable the TRKCELL value.

Data Separation: To avoid contention for I/O, printer resources must be spread evenly across channels, volumes, and actuators. For example, the spool volume and the PSF resource library that holds page segments should be on different channels, and both should be separated from the printer channel.

Functional Subsystem Interface (FSI): In deferred-printing mode, JES communicates with PSF through the Functional Subsystem Interface. As an FSI system, PSF is marked pageable but not swappable. For efficient use of operating system paging and swapping facilities, PSF and TSO users should be assigned to different performance groups.

Checkpointing: A checkpoint provides the information needed to restart a print data set after an interruption or a recoverable error condition. When restarting, PSF instructs the printer to discard all unprinted pages, reposition to the last checkpoint, and resume printing from that point. If no checkpoints have been taken, PSF repositions to the beginning of the print data set.

The checkpoint interval can be specified in pages or in time. When you choose a value for it, you should bear in mind the speed of the printer, the characteristics of typical print data sets sent to the printer, and the following considerations:

- A checkpoint interval that is too small—that is, too frequent—can increase processor utilization and decrease printer throughput.
- A checkpoint interval that is too large makes the recovery of a print job too lengthy. The reason is that, for SNA Logical Unit 1 (LU 1)-attached printers and TCP-IP-attached printers that can produce asynchronous data stream errors,[20] PSF issues a print buffered data command each time a checkpoint is taken. This command is required to make sure that the printer has completely checked the data for errors. After issuing this data command, PSF must wait until all the pages that have been sent to the printer have been printed. If the checkpoint intervals are too frequent, the performance of these printers can be slowed.

Tracing: Tracing is useful in diagnosing problems; however, tracing increases processor utilization and DASD requirements, and can decrease printer throughput. Unless you need to do problem determination, turn tracing off.

[20] Some examples of TCP/IP-attached and LU 1-attached printers that produce asynchronous data stream errors are the 3812, 3816, 3912, 3916, 3930, and 4028 printers.

Chapter 19. Connecting a TCP/IP-Attached IPDS Printer

This chapter describes how to configure MVS, TCP/IP, and PSF to print on TCP/IP-attached IPDS printers, including IPDS printers attached by use of the i-data 7913 IPDS Printer LAN Attachment.

Requirements for Using a TCP/IP-Attached IPDS Printer

To print on TCP/IP-attached IPDS Printers through PSF, you need the following software:

- PSF/MVS Version 2 Release 2.0 with APAR OW15599.
- MVS Scheduler APAR OW12236 to support two new PRINTDEV keywords: IPADDR and PORTNO.
- TCP/IP Version 3 Release 1, or higher, installed and configured on MVS.

  Note: To specify a TCP/IP address space name other than the default, TCPIP, code the following EXEC statement in the appropriate PSF writer procedure:

  ```
  //STEP01   EXEC PGM=APSPPIEP,REGION=4096K,PARM=(,,,,tcpip_name)
  ```

  where *tcpip_name* is the name of the TCP/IP address space. If this parameter is not coded, PSF uses the default name, `TCPIP`. For a full description of the PARM parameter, refer to *Print Services Facility/MVS: Diagnosis Guide and Reference*.

For a list of the printers that can be TCP/IP-attached, refer to *Advanced Function Presentation: Printer Information*, G544-3290.

Overview

PSF views TCP/IP-attached printers in the same way as it views SNA-attached printers. PSF performs error recovery, handles intervention-required situations, and displays messages exactly as it does for SNA-attached printers.

TCP/IP software provides universal communication services (interfaces) between physical networks and applications. The communication services reside at the network layer and are independent of the topology of the underlying physical network, as Figure 62 shows.

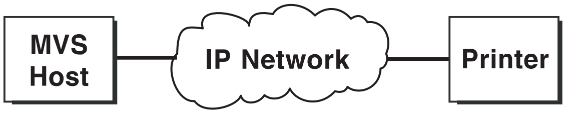


Figure 62. Appearance of the TCP/IP Connection between an MVS Host and a Printer

For example, in communications between an MVS host and a printer, as in Figure 62, the IP network performs like a single network. In fact, however, the network is physically like that shown in Figure 63 on page 236.

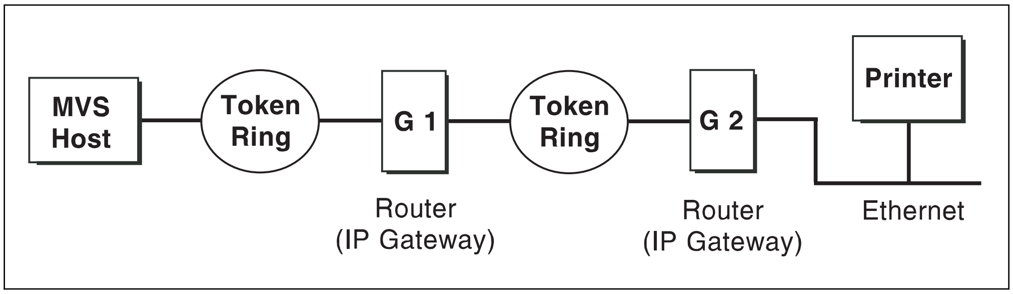


Figure 63. Actual Physical TCP/IP Connection between an MVS Host and a Printer

The routing of information is determined on the basis of the IP address, and is performed by IP gateways. After the network is configured correctly, the MVS host appears to be communicating directly with the TCP/IP-attached printer.

For a more detailed description of TCP/IP networks, refer to the following:

- *Internetworking with TCP/IP, Principles, Protocols, and Architecture*
- TCP/IP *Tutorial and Technical Overview*

Connecting TCP/IP-Attached Printers

To connect a TCP/IP printer so that you can print from PSF, do the following:

1. Define the MVS communications control unit to MVS.
2. Modify the TCP/IP profile on your MVS system, if necessary.
3. Install the i-data 7913 IPDS Printer LAN Attachment, if necessary (see Appendix J, “Installing the i-data 7913 IPDS Printer LAN Attachment” on page 329).
4. Ping the printer; see “Pinging the Printer” on page 239.
5. Define the printer as a writer-controlled printer to JES.
6. Define the printer to PSF with a PRINTDEV statement, including the IP address.

This section does not provide all the information you need to install and configure TCP/IP on your MVS system. For more information on installing TCP/IP, refer to *TCP/IP for MVS: Customization and Administration Guide*.

Defining the Communications Control Unit to MVS

If you have not already done so, define the communications control unit for the MVS host, such as the 3172 or the 3745 control unit, to MVS. Use either an MVS configuration program (MVSCP), or a hardware configuration definition (HCD), depending on the version of your MVS system:

- When using a version earlier than MVS 4.1.0, use an MVSCP.
- When using a version of MVS 4.1.0 or later, use an HCD or an MVSCP.

For more information about using these methods, refer to one of the following publications:

- *MVS/ESA Planning: Dynamic I/O Configuration*
- *OS/390 HCD Planning*
- *HCD User's Guide*

Modifying the TCP/IP profile in MVS

The TCP/IP profile contains system configuration statements used to initialize the TCP/IP address space. Of those statements, the following are subject to special considerations when you are printing from PSF on TCP/IP-attached printers:

- DATABUFFERPOOLSIZE, which specifies the number and size of data buffers
- SMALLDATABUFFERPOOLSIZE, which specifies the number of small data buffers
- TINYDATABUFFERPOOLSIZE, which specifies the number of tiny data buffers
- KEEPALIVEOPTIONS, which specifies the interval between keep-alive transmissions
- GATEWAY, which defines static routes, including the maximum packet size for each route

Samples of these statements are printed in bold in Figure 64.

```
ACBPOOLSIZE                 1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                 150
DATABUFFERPOOLSIZE          160  32768
ENVELOPEPOOLSIZE            750
IPROUTEPOOLSIZE             300
LARGEENVELOPEPOOLSIZE       50
RCBPOOLSIZE                 50
SCBPOOLSIZE                 256
SKCBPOOLSIZE                256
SMALLDATABUFFERPOOLSIZE     256
TCBPOOLSIZE                 512
TINYDATABUFFERPOOLSIZE      256
UCBPOOLSIZE                 100

KEEPALIVEOPTIONS INTERVAL 10 SENDGARBAGE FALSE ENDKEEPALIVEOPTIONS
GATEWAY
; * Network  First hop   Linkname Packet size Subnet mask Subnet value
  9          =           BPCLAN    2000       0.255.255.0  0.99.12.0
  DEFAULTNET 9.99.12.254 BPCLAN    2000       0.255.255.0  0
```

Figure 64. Sample Parameters in the TCP/IP Profile. This example is **not** *a complete TCP/IP profile.*

This section contains information about the statements, the special considerations that apply to them, and the changes they may make necessary. If you change any of the values in the TCP/IP profile, restart TCP/IP to pick up the changes.

For information about the TCP/IP profile and the statements described in this section, refer to:

- *IBM TCP/IP MVS Customization and Administration Guide*
- *IBM TCP/IP Performance and Tuning Guide*

DATABUFFERPOOLSIZE Statement: The DATABUFFERPOOLSIZE statement defines the number and size of the data buffers. For printing on TCP/IP-attached printers, IBM recommends that you specify at least the following:

- 160 data buffers
- 32 768 buffer size

SMALLDATABUFFERPOOLSIZE Statement: The SMALLDATABUFFERPOOLSIZE statement defines the number of small data buffers. For printing on TCP/IP-attached printers, IBM recommends that you specify at least 256 small data buffers.

TINYDATABUFFERPOOLSIZE Statement: THE TINYDATABUFFERPOOLSIZE statement defines the number of tiny data buffers. For printing on TCP/IP-attached printers, IBM recommends that you specify at least 256 tiny data buffers.

KEEPALIVEOPTIONS Statement: PSF relies on TCP to detect when a connection with a TCP/IP-attached printer or an i-data 7913 IPDS Printer LAN Attachment is no longer usable. When no data has been exchanged between PSF and its connection partner, TCP sends keep-alive probes to the connection partner periodically. These periodic probes, called keep-alive transmissions, enable TCP to discover when a connection is no longer usable even if the connection partner is abruptly powered off or is no longer accessible through the network.

The frequency of keep-alive transmissions is controlled by the INTERVAL parameter on the KEEPALIVEOPTIONS statement. The frequency applies to all TCP applications that direct TCP to send keep-alive transmissions. The default frequency is after about two hours of inactivity.

For printing on TCP/IP-attached printers, IBM recommends that you specify a shorter interval than the default, such as 10 minutes, for the interval between keep-alive transmissions.

Also, if any target host requires that the keep-alive packet contain data, specify SENDGARBAGE TRUE. For example:

```
KEEPALIVEOPTIONS INTERVAL 5 SENDGARBAGE TRUE ENDKEEPALIVEOPTIONS
```

GATEWAY Statement: The *Packet_size* parameter of the GATEWAY statement defines the maximum transmission unit (MTU) for the MVS host. The MTU size must not exceed the maximum size that can be sent through the control unit; if it does, transmission problems will occur.

For printing on TCP/IP-attached printers, IBM recommends the following:

- The MTU size should be 2000 bytes.
- The MTU size for the MVS host should be the same as the MTU size for the printer.

Note: For values in the GATEWAY statement other than the packet size, specify the values that are correct for your installation.

Installing the i-data 7913 IPDS Printer LAN Attachment

Some IPDS printers must be attached to a LAN by use of the i-data 7913 IPDS Printer LAN Attachment. The i-data 7913 IPDS Printer LAN Attachment must be configured to TCP/IP on an AIX or OS/2 system that is running the **bootp** daemon program, which is required for initialization of the i-data 7913 IPDS Printer LAN Attachment.

If you have not already installed and configured the i-data 7913 IPDS Printer LAN Attachment, see Appendix J, “Installing the i-data 7913 IPDS Printer LAN Attachment” on page 329, for information about installing and configuring it.

Pinging the Printer

To verify that the MVS system can establish a connection with the TCP/IP-attached printer, ping the printer from the MVS system. To do this, issue the following command from a TSO session:

```
TSO PING ip-address
```

In JES2, enter the following command from the System Display and Search Facility (SDSF) menu 6:

```
ping ip-address
```

ip-address
Specifies the Internet Protocol (IP) address of the printer or of the i-data 7913 IPDS Printer LAN Attachment.

Here is an example of a successful ping:

```
EZA0458I Ping V3R1: Pinging host 9.99.12.33
(Use ATTN to interrupt.)
EZA0463I PING: Ping #1 response took 0.084 seconds.
Successes so far = 1.
```

And here is an example of an unsuccessful ping:

```
EZA0458I Ping V3R1: Pinging host 9.99.12.33
(Use ATTN to interrupt.)
EZA0464I PING: Ping #1 timed out
```

Ping Is Not Successful: If the ping is not successful, and the printer is attached by use of the i-data 7913 IPDS Printer LAN Attachment, see “Troubleshooting the i-data 7913 IPDS Printer LAN Attachment” on page 342; otherwise, verify the following:

- The printer is powered on.
- The MVS address is unique in the TCP/IP network.

  If the IP address of the MVS system is not unique, see your network administrator to resolve the IP-address problem.
- The Maximum Transmission Unit (MTU) size of the IP packet for the MVS system is equal to the MTU size of the printer. For information about the values recommended for the MVS system, see “GATEWAY Statement” on page 238.
  - To change the MTU size for the MVS system, change the GATEWAY statement in the MVS TCP/IP profile, and restart TCP/IP to pick up the changes.
  - To change the MTU size for the printer, refer to the documentation for your printer.

If these items are all in order, consult your TCP/IP network administrator about a possible network problem.

Ping Is Successful: A successful ping usually indicates that the MVS system can communicate with the printer; however, you might receive a successful ping even though the IP address of the TCP/IP-attached printer is a duplicate of another IP address.

If PSF is unable to establish a network connection with the TCP/IP-attached printer, or if PSF output for this printer prints elsewhere, follow these steps to determine whether the IP address of the printer is unique:

1. Turn off the printer and the i-data 7913 IPDS Printer LAN Attachment, if any.
2. Wait at least 5 minutes for TCP/IP to clear the Address Resolution Protocol (ARP) tables.[21]
3. Enter the **ping** command again from the MVS system.

If you receive a successful response to the ping command, there is a duplicate IP address, Consult your TCP/IP network administrator.

Defining the Printer to JES

When a TCP-attached printer is to be used with JES, it must be defined for deferred-printing mode. For the procedure, see Chapter 7, "Using Deferred-Printing Mode" on page 63; in particular, see "Sample Initialization Statements for JES2" on page 64 and "Sample Initialization Statements for JES3" on page 69.

Defining the Printer to PSF

Each TCP/IP-attached printer must be defined to PSF with a PRINTDEV statement in the PSF startup procedure. The following keyword is required on the PRINTDEV statement:

IPADDR='*ip-address***'**

ip-address
Specifies the Internet Protocol (IP) address for the printer or attachment, in dotted-decimal notation. Do not specify a host name in place of the dotted-decimal address.

Figure 65 on page 242 shows a sample procedure, APSWPROT, shipped with this APAR, which you can modify to suit your installation. The PRINTDEV statement shows the required IPADDR keyword.

The APSWPROT procedure contains JCL parameters to produce a startup procedure for either 240-pel and 300-pel printers, but the references to 300-pel resolution are commented out. To create separate startup procedures, make two copies of APSWPROT, and then follow the commented instructions in APSWPROT. Keep one unchanged copy as your startup procedure for a resolution of 120 or 240 pels. In the second copy, delete or comment out all references to 240-pel resolution, and make all references to 300-pel resolution active. This changed copy is your startup procedure for 300-pel resolution.

If a printer can print both 240-pel and 300-pel jobs, create two separate FSAs, one for jobs with resolution of 120 or 240 jobs, and one for jobs with resolution of 300

[21] If your installation specified a longer interval on the ARPAGE configuration statement in the TCP/IP profile, you may need to wait longer. For information about the ARPAGE statement, refer to *IBM TCP/IP MVS Customization and Administration Guide*.

pels.[22] The 240-pel resolution FSA uses 240-pel resources,[23] and the 300-pel resolution FSA uses 300-pel resources. Both FSAs are defined to drive the same printer; however, only one FSA can be active at one time for a given printer. You should define separate classes or destinations for each FSA. Your installation must route the appropriate jobs to the appropriate class or destination. The two FSAs can be defined in the same or in separate startup procedures.

[22] Although doing so is not recommended, you can drive the printer with just one FSA, if one of the following conditions is met:

- All the resources in the resource libraries are resolution-independent, and all the jobs that use inline resources or user libraries contain resources that are resolution-independent.
- All jobs that request a resolution that is different from the resolution-dependent resources in the PSF system and security libraries contain the resources at the requested resolution, either as inline resources or in the user's library.

[23] Even if a PSF user requests 120-pel resolution for an AFP print job being sent to the printer, PSF requires 240-pel resources from the library. Do not store 120-pel resources in the library.

```
//APSWPROT PROC
//**************** THE PSF TCP/IP WRITER PROCEDURE ***************
//*
//*01* MODULE-NAME = APSWPROT
//* $MOD(APSWPROT) COMP(APS) PROD(PSF) : RELEASE 2.2.0
//*
//*01* DESCRIPTIVE-NAME = START PROCEDURE FOR PSF:
//*
//*                             TCP/IP ATTACHED DEVICES
//*
//*01* STATUS = VERSION 2, RELEASE 2, LEVEL 0
//*
//*01* FUNCTION = THIS PROCEDURE IS COPIED FROM THE
//*               DISTRIBUTION LIBRARY TO SYS1.PROCLIB.
//*               THIS COPY IS PERFORMED BY MACRO 'SGAPS5PR'.
//*
//*01* NOTES = THE FULL NAME OF THE DEFAULT PAGEDEF IS
//*               P1A06462.
//*            THE FULL NAME OF THE DEFAULT FORMDEF IS
//*               F1A10110.
//*            THE FULL NAMES OF THE DEFAULT FONTS ARE
//*               X0GF10, X0GS10, X0TU10, AND X0GU10.
//*            THE FULL NAME OF THE SEPARATOR PAGE PAGEDEF IS
//*               P1V06483.
//*            THE FULL NAME OF THE SEPARATOR PAGE FONT IS
//*               X0GT15.
//*
//* REQUIRED ACTIONS =
//*    RESOLUTION - THIS START PROCEDURE IS SET UP FOR DRIVING
//*      A TCP/IP DEVICE AT 240 PEL RESOLUTION. TO CHANGE IT
//*      TO DRIVE 300P PEL RESOLUTION TCP/IP ATTACHED DEVICES:
//*
//*      A) COMMENT OUT THE FIRST PRINTDEV STATEMENT AND UNCOMMENT
//*         THE SECOND PRINTDEV STATEMENT
//*
//*      B) THE FONT02 DATA SET MUST CONTAIN THE CONVERTED
//*         DEFAULT FONTS LISTED IN THE PRINTDEV.
//*         THE FONTS ARE CONVERTED USING THE FONT
//*         CONVERSION UTILITY APSRCF30. SEE THE PSF
//*         SYSTEM PROGRAMMING GUIDE.
//*
//*01* CHANGE-ACTIVITY :
//* $H1=LAPS0001, HAF1103, 080195 DKU1JAU : TCP/IP SUPPORT
//*
//**** END OF SPECIFICATIONS ***/
```

Figure 65 (Part 1 of 2). Sample JCL Startup Procedure for a TCP/IP-Attached Printer (APSWPROT)

```
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K
//JOBHDR OUTPUT PAGEDEF=V06483,     /* JOB SEPARATOR PAGEDEF        */
//       FORMDEF=A10110,CHARS=GT15  /* JOB SEPARATOR FORMDEF        */
//JOBTLR OUTPUT PAGEDEF=V06483,     /* JOB SEPARATOR PAGEDEF        */
//       FORMDEF=A10110,CHARS=GT15  /* JOB SEPARATOR FORMDEF        */
//DSHDR  OUTPUT PAGEDEF=V06483,     /* DS  SEPARATOR PAGEDEF        */
//       FORMDEF=A10110,CHARS=GT15  /* DS  SEPARATOR FORMDEF        */
//MSGDS  OUTPUT PAGEDEF=A06462,     /* MESSAGE DATASET PAGEDEF      */
//       FORMDEF=A10110             /* MESSAGE DATASET FORMDEF      */
//FONT01   DD   DSN=SYS1.FONTLIBB,  /* SYSTEM FONTS - 240 PEL       */
//         DISP=SHR
//FONT02   DD   DSN=SYS1.FONT300,   /* SYSTEM FONTS - 300 PEL       */
//         DISP=SHR
//PSEG01   DD   DSN=SYS1.PSEGLIB,   /* SYSTEM PAGE SEGMENTS         */
//         DISP=SHR
//OLAY01   DD   DSN=SYS1.OVERLIB,   /* SYSTEM MEDIUM OVERLAYS       */
//         DISP=SHR                 /*                              */
//PDEF01   DD   DSN=SYS1.PDEFLIB,   /* SYSTEM PAGEDEFS              */
//         DISP=SHR
//FDEF01   DD   DSN=SYS1.FDEFLIB,   /* SYSTEM FORMDEFS              */
//         DISP=SHR
//* ***********************************************************     */
//*             PRINTDEV                                            */
//* ***********************************************************     */
//PRT1     CNTL
//PRT1     PRINTDEV FONTDD=*.FONT01, /* 240 PEL FONT LIBRARY DD     */
//*                                  /* <-- SEE REQUIRED ACTIONS    */
//*                                  /*     ABOVE                   */
//*PRT1     PRINTDEV FONTDD=*.FONT02, /* 300 PEL FONT LIBRARY DD    */
//*                                   /* <-- SEE REQUIRED ACTIONS   */
//*                                   /*     ABOVE                  */
//          OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD         */
//          PSEGDD=*.PSEG01,         /* SEGMENT  LIBRARY DD         */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD         */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD         */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR        */
//*                                  /*   OUTPUT                    */
//          JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR       */
//*                                  /*   OUTPUT                    */
//          DSHDR=*.DSHDR,           /* DATA SET HEADER             */
//*                                  /*   SEPARATOR                 */
//          MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT     */
//          BUFNO=5,                 /* NUMBER OF WRITE DATA BUFFERS */
//          PAGEDEF=A06462,          /* DEVICE PAGEDEF DEFAULT      */
//          FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT      */
//          CHARS=(GF10,             /* DEVICE                      */
//          GS10,TU10,GU10),         /* DEFAULT FONT SET            */
//          PIMSG=YES,               /* ACCUMULATE DATA SET         */
//*                                  /*   MESSAGES                  */
//          DATACK=BLOCK,            /* REPORT ALL DATA-CHECK       */
//*                                  /*   ERRORS                    */
//          TRACE=NO,                /* CREATE INTERNAL TRACE       */
//          FAILURE=WCONNECT,        /* PSF ACTION ON PRINTER       */
//*                                  /*   FAILURE                   */
//          TIMEOUT=REDRIVE,         /* PSF ACTION ON TIMEOUT       */
//          MGMTMODE=IMMED,          /* PRINTER MANAGEMENT MODE     */
//          DISCINTV=0,              /* DISCONNECT INTERVAL IN      */
//*                                  /*   SECONDS                   */
//          IPADDR='xxx.xxx.xxx.xxx' /* IP ADDRESS FOR TCP/IP       */
//PRT1      ENDCNTL
```

Figure 65 (Part 2 of 2). Sample JCL Startup Procedure for a TCP/IP-Attached Printer (APSWPROT)

Operating a TCP/IP-Attached Printer

To operate a TCP/IP-attached printer, including a printer attached by use of the i-data 7913 IPDS Printer LAN Attachment, you use JES operator commands, just as if the printer were channel-attached or SNA-attached.

Starting a TCP/IP-Attached Printer

To start a TCP/IP-attached printer, do the following:

1. Start TCP/IP.
2. Power on the printer.
3. Power on the i-data 7913 IPDS Printer LAN Attachment, if any.
4. Start the printer FSA by the method described in Chapter 7, “Using Deferred-Printing Mode” on page 63.

Stopping a TCP/IP-Attached Printer

You can stop a TCP/IP-attached printer in several ways:

- The preferred method is to first stop the PSF FSA for the printer by entering the following command from the MVS console:

  JES2:

  ```
  $Pprinter-name
  ```

  JES3:

  ```
  *VARY printer-name,OFF
  *CANCEL printer-name
  ```

  where *printer-name* specifies the name of the printer FSA.

  Then you can turn off power to the printer and the i-data 7913 IPDS Printer LAN Attachment, if any.

- You can turn off power to a printer that is not attached by use of the i-data 7913 IPDS Printer LAN Attachment.

  This procedure is not recommended, because it may take PSF some time to recognize that the printer is powered off. The length of that time depends on the value specified for the keep-alive transmissions, described on page 238.

  To end the PSF FSA for the printer, use the JES commands described in the first bullet. If you are unable to purge or cancel the printer by use of the JES commands, enter the following command:

  ```
  MODIFY FSSname,FORCE,printer-name
  ```

- You can turn off power to a printer that *is* attached by use of the i-data 7913 IPDS Printer LAN Attachment. The 7913 notifies PSF that the printer is powered off, and the printer FSA eventually ends normally.

- You can turn off power to the i-data 7913 IPDS Printer LAN Attachment without turning off power to the printer.

  This procedure is not recommended, because it may take PSF some time to recognize that the attachment is powered off. The length of that time depends on the value specified for the keep-alive transmissions, described on page 238.

To end the PSF FSA for the printer, use the JES commands described in the first bullet. If you are unable to purge or cancel the printer by use of the JES commands, enter the following command:

```
MODIFY FSSname,FORCE,printer-name
```

Note: Before powering the i-data 7913 IPDS Printer LAN Attachment back on, be sure to power the printer off and then on again.

Chapter 20. Responding to PSF Errors

PSF messages are formatted and printed in accordance with a page definition and a form definition specified in the PSF startup procedure. If no default page definition and form definition are specified for formatting messages, PSF will not print the message data set.

Notes:

1. PSF messages cannot be printed in double-byte fonts. The page definition you specify to format and print messages must use only single-byte fonts.
2. To ensure that PSF messages are formatted to fit on a page, use a page definition that specifies the ACROSS printing direction. For a list of PSF-supplied page definitions that you can use to format messages in the ACROSS direction, refer to *Print Services Facility/MVS: Application Programming Guide*.

For some JES operator-initiated interrupts, such as the JES2 $I and $B commands, PSF prints a message within the data set, informing the user of the interrupt.

Messages not associated with a print data set—for example, messages related to separator-page processing—are sent to the operator's console. All messages contain information about the cause of the error. The messages are explained in *Print Services Facility/MVS: Messages and Codes*.

Accumulating PSF Error Messages

PSF usually accumulates error messages generated during the processing of a data set, and prints those messages at the end of the data set, preceding any separator pages.

To specify whether PSF is to accumulate messages, use the PIMSG parameter, either in the PRINTDEV statement or in the JCL. You can also use the PIMSG parameter to specify that PSF is to stop processing and purge a data set after a certain number of errors, or to redirect the messages; see “Redirecting the Message Data Set” on page 115. For more information on PIMSG, see “Printing PSF Error Messages” on page 116 and Table 2 on page 53. For information about using PIMSG in JCL, refer to *Print Services Facility/MVS: Application Programming Guide*.

Repeating Error Messages

If a paper jam occurs in your printer, or if the system operator enters a FORWARD or BACKWARD space command, some error messages may be repeated in the message data set. Usually, these messages indicate 1) errors detected in the data stream when the page was initially sent to the printer and 2) errors repeated when the page was retransmitted.

Recovering from PSF Errors

Under JES, PSF monitors pages as they pass through the printer, and records checkpoint data. If a recoverable I/O error interrupts the processing of a data set, PSF uses this checkpoint data to resume processing at the point of interruption, with minimal operator intervention.

For some kinds of error, sense-byte information may be provided to the operator. Some of the printers PSF supports report 24 bytes of sense information to identify the cause of every unit-check error; other printers report fewer sense bytes. Sense byte 2 contains error-recovery action codes for conditions sensed by the printer. For more detailed sense-byte information, refer to the publications for your printer.

If PSF is running in direct-printing mode, error conditions that cause data loss can also cause an abend.

Recovering from Printer-Hardware Problems

PSF provides System Display and Search Facility (SDSF) notification to an operator authorized to receive messages for a printer. However, when the JES SDSF operator receives a PSF error message indicating that a printer-hardware condition requires intervention, the type of intervention is not specified. For more specific information about SDSF, refer to the SDSF publications listed in Appendix K, "Related Publications."

For most other printer-hardware problems, PSF invokes device-dependent error-recovery procedures, issues a message to the operator, and then takes appropriate action. These actions are described in *Print Services Facility/MVS: Messages and Codes*. If operator intervention is required (to clear a paper jam, for example), the action taken depends on the type of printer being used.

If a printer is host-connected and channel-attached, PSF issues a message to the operator console, calling for operator intervention. PSF will not resume processing until the operator takes corrective action.

If the printer has a local-intervention timer, as does the 3820 printer for example, and if the timer expires, PSF issues a message. Action is then taken as defined in the FAILURE parameter of the PRINTDEV statement. You can also set the timer so that it never expires.

For a printer, such as the 3812 printer, that has no local-intervention timer, PSF waits for the printer operator to correct problems as they occur.

Terminating a Printer FSA

If you are unable to purge or cancel a printer by use of the JES command, you can use the PSF operator interface to terminate the printer FSA. The syntax of the MODIFY command for terminating a printer FSA is as follows:

```
{MODIFY|F} FSSname,FORCE ,printer-name
```

To terminate a printer FSA, an operator enters the MODIFY (or F) command with the FORCE parameter. The following list describes the parameters used with the MODIFY command. All the parameters are required.

FSSname
Specifies the name of the FSS that manages the printer FSA that is to be terminated. The *FSSname* parameter must match the FSS name on the JES2 FSS statement or the JES3 FSSDEF statement for the FSS.

FORCE
Specifies that the printer FSA specified by *printer-name* is to be terminated.

printer-name
Specifies the printer FSA that is to be terminated. For JES2, either of two formats, PRINTR*nn* or PRT*nnnn*, can be used for *printer-name*.

Note: When processing the FORCE command, PSF issues APS627I.

If the FORCE command is unsuccessful on JES3 systems, use the JES3 command to terminate the printer FSA. That command is as follows:

```
*FAIL,PRTXX
```

Understanding Restartable Abends

PSF refreshes the FSA and restarts it for some abend conditions. (However, PSF does not request a restart more than three times within two hours for an abend code.) This processing enhancement increases the availability of PSF and reduces the need for operator intervention, while retaining the ability to provide diagnostic information.

To determine whether an abend is restartable, refer to *Print Services Facility/MVS: Messages and Codes*. For an explanation of restartable abends, refer to *Print Services Facility/MVS: Diagnosis Guide and Reference*.

| Because restarting after an abend may cause duplicate pages to print without any
| notification to the operator or job submitter, you may want to prevent PSF from
| restarting after abend conditions. See “Considerations on Restartable Abends” on
| page 198 for information about how to use exit APSUX07 to prevent PSF from restarting after restartable abends.

Requesting Traces and Dumps

PSF provides two types of diagnostic aids: trace reports and formatted dumps. These aids provide a record of events within PSF that can be used in problem determination and diagnosis. For information on using PSF traces and dumps, refer to *Print Services Facility/MVS: Diagnosis Guide and Reference*.

Chapter 21. Supporting AFP Output on Microfilm

PSF/MVS 2.2.0 with APAR OW10067 adds support for sending Advanced Function Presentation (AFP) data to microfilm[24] devices in deferred-printing mode. PSF/MVS does *not* support microfilm for MVS users who are using:

- PSF Direct
- Direct Print mode
- Distributed Print Facility (DPF)

After you establish a PSF startup procedure for microfilm, the microfilm device appears to PSF to be a continuous-forms N_UP printer operating in simplex mode.

Note: This chapter assumes that you know how to configure PSF to enable a PSF user to print AFP data on paper output.[25] On the basis of this assumption, additional prerequisites and information are presented so that you can enable PSF users to also print AFP data to microfilm in the MVS environment. Just as you might need to refer to other documentation for printer-specific information, you may need to refer to documentation supplied by the manufacturer of the microfilm device. For a list of documents about microfilm, see Appendix K, “Related Publications.”

PSF Hardware Attachment Requirements for Microfilm Devices

PSF can physically attach to a microfilm device in the following ways, by specific protocols:

- Parallel channel
  - Block multiplexer
  - Data streaming
  - Two-channel switch—optional
- Enterprise Systems Connection (ESCON)

PSF Software Requirements for Microfilm Devices

The following levels of software, as well as higher ones, are supported for printing microfilm:

- MVS Versions 5.1, 4.3, and 4.2 with APARs
- PSF/MVS Version 2.2.0 with APARs
- JES2 Versions 5.1, 4.3, 4.2 with APARs
- JES3—No additional level is needed

[24] *Microfilm* can mean either microfiche or 16mm film.

[25] If you are not familiar with configuring PSF, IBM recommends that, in this publication, you read the introductory chapter and the information about defining channel-attached printers, ESCON-attached printers, and SNA-attached printers that use IBM token rings, as well as the chapters that describe working with libraries, changing the system environment, using PRINTDEV parameters, using deferred-printing mode, managing resources and selecting special processing parameters, understanding PSF data protection and security, using installation exits, and understanding AFP performance. If you are not familiar with PSF vocabulary terms, you may want to check the glossary.

- SDSF Version 1.5 (if you are modifying the user-specified parameter, **COMSETUP=**)[26]

PSF System Programmer Tasks for Microfilm Devices

This section presents a visual representation of how PSF and a microfilm device work together to print AFP information to microfilm. The rest of this chapter describes the process by which a user of PSF/MVS can be enabled to print AFP jobs to microfilm.

Figure 66 shows a possible configuration for a microfilm system. This figure represents the flow of an AFP document and related resources through PSF into the microfilm device.

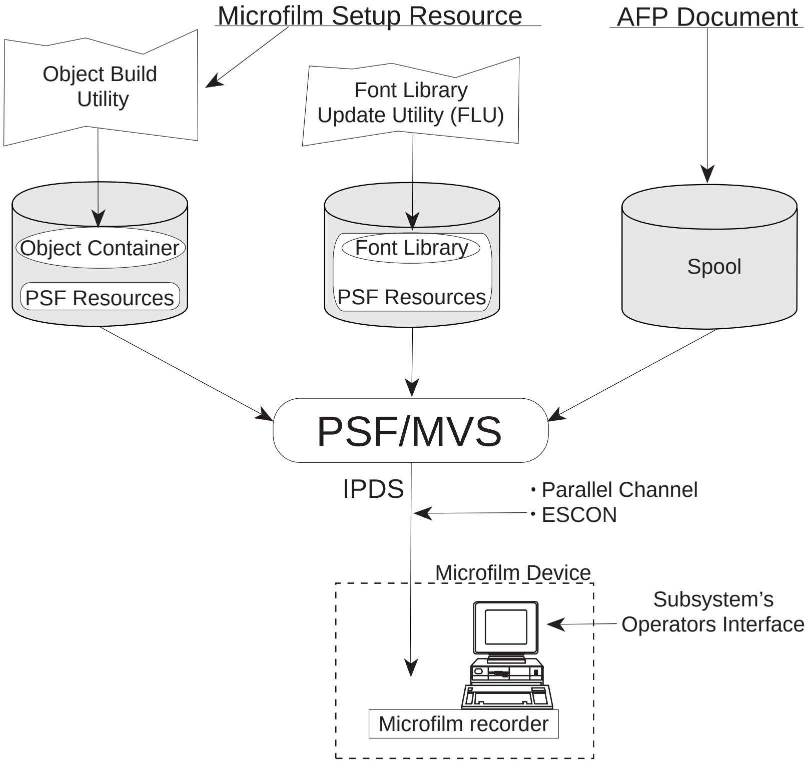


Figure 66. Example of AFP Data Being Transmitted to a Microfilm Device

Even though your system has been configured and installed as instructed by the manufacturer of the microfilm device, you still have some tasks to perform before the microfilm device and PSF/MVS can work together to enable PSF users to print AFP jobs on microfilm:

- Attaching and defining the microfilm device to MVS or VTAM
- Defining the microfilm device to JES2 or JES3
- Defining the microfilm device to PSF
- Establishing a PSF startup procedure

[26] **COMSETUP=** is a keyword required when AFP jobs are sent to microfilm; it is described later in this chapter.

- Working with the libraries required for microfilm jobs
- Managing PSF resources and selecting special processing parameters for microfilm jobs
- Managing PSF data protection and security for microfilm jobs
- Using PSF installation exits for microfilm jobs
- Understanding PSF performance issues as they pertain to microfilm
- Responding to PSF errors reported by a microfilm device
- Redirecting PSF messages for microfilm jobs
- Recovering or restarting AFP print jobs for microfilm output

Defining the Microfilm Device as Channel-Attached or ESCON-Attached

The microfilm device can be channel-attached or ESCON-attached. With this in mind, you must attach the microfilm device and define it to MVS, to JES2 or JES3, and to PSF.

Defining the Channel-Attached or ESCON-Attached Microfilm Device to MVS

Whether the microfilm device is channel-attached or ESCON-attached, follow the procedures for attaching and defining channel-attached printers to MVS as explained in Chapter 2, “Defining Channel-Attached Printers to MVS,” entering **AFP1** as the unit type.

Defining the Microfilm Device to JES2 and JES3

Just as you coded JES initialization statements to define PSF channel-attached and SNA-attached printers, you must also code JES initialization statements to define the microfilm device.

What should you define differently for microfilm?

The microfilm device requires that you run with checkpointing turned OFF when printing. Therefore,

- Avoid using the CKPTPAGE and CKPTSEC JCL parameters for microfilm jobs.
  - For JES2, specify CKPTMODE=SEC and CKPTSEC=0.
  - For JES3, specify CKPNTPG=32767.

To avoid printing unneeded microfilm frames and to improve performance, consider the following suggestions:

- Specify the Non-Process Run-Out parameter (NPRO).

  Performing the NPRO may take significantly longer on a microfilm device than on a printer; it depends on the microfilm device used.

  When specifying a non-zero NPRO value, consider the following:

  - While PSF is performing an NPRO, no jobs are selected for printing. Selection of jobs for printing resumes after the NPRO is complete.

- The Missing Interrupt Handler timeout value needs to be set to a value twice as long as the time to perform an NPRO.
- PSF issues an NPRO before the device is drained.
- If a printer setup, such as a forms change, is requested through JES, the setup message is not issued to the operator until the NPRO is complete.

When specifying an NPRO value of zero, consider the following:

- When PSF receives a request to drain the microfilm device, PSF does not make sure that all pages have been stacked. It marks as complete all jobs sent to the microfilm device. If an error occurs after the device is drained, and the microfilm is lost, the job must be run again.
- PSF does not query the microfilm device when it has no work to do. Therefore, a job that is stacked may not be released to JES as complete (and purged off the queue) until a subsequent job begins processing, or the microfilm device is drained.
- If the microfilm device is powered off or the PSF address space is cancelled, PSF releases as incomplete all jobs that were not stacked at the time PSF last queried the device. Unless they are manually purged from the queue, all these jobs are reprinted from the beginning (because there are no checkpoints).
- An NPRO is performed even if a printer setup is requested through JES (for example, a forms change). The setup message is not issued to the operator until the NPRO is complete.

- Recommend that printer operators *not* initiate system changes for different resolutions from the microfilm operator panel.
- Users sending AFP data to a microfilm device may not want to print header, trailer, or separator pages.
- To prevent data checks from being reported, specify DATACK=BLOCK in your PRINTDEV statement.

For more information about defining printers to JES, see Chapter 7, “Using Deferred-Printing Mode.”

PRINTDEV Parameters for Microfilm Devices

PSF requires the following keywords for all microfilm devices:

COMSETUP=
OBJCONDD=

COMSETUP=*membername* is a microfilm setup resource parameter that specifies a member name. The manufacturer of the microfilm device provides a utility for you to use when generating member names. IBM recommends that you use **H1** as a prefix to every member name for the microfilm setup resource parameter. In Figure 67, for example, *H1SETUPD* is specified as a default member name for the microfilm setup resource parameter.

Note: If you want the member name to be prefixed with **H1**, you must include **H1** as part of the name you specify; PSF does not add a prefix.

The microfilm setup resource parameter is used only for sending output to a microfilm device. For syntax rules, see Table 2 on page 53.

As Figure 67 on page 256 shows, the COMSETUP= keyword is also used in the sample OUTPUT statement for microfilm. Although you are advised to define the microfilm device to JES without job headers, job trailers, and data set separator pages, and to turn off messages to save microfilm frames, you may elect to print this information to microfilm. Without a COMSETUP, the microfilm device cannot print the headers, trailers, data set separators, or messages to microfilm.

OBJCONDD=*.label* identifies the DD statement specifying the object container library.[27] This keyword follows the rules of any other PSF resource library reference on the PRINTDEV statement; for the syntax, see Table 2 on page 53.

Defining the Microfilm Device to PSF

You should create two separate PRINTDEVs: one for thejobs with 120-pel and 240-pel resolution, and one for those with 300-pel resolution jobs.[28] The 240-pel resolution PRINTDEV uses 240-pel resources,[29] and the 300-pel resolution PRINTDEV uses 300-pel resources. Both PRINTDEVs are defined to drive the same microfilm device; for a given microfilm device, however, only one PRINTDEV can be active at a time. You should define separate classes or destinations for each PRINTDEV. Your installation must route the appropriate jobs to the appropriate class or destination. The two PRINTDEVs can be defined in the same (or separate) startup procedures.

Figure 67 on page 256 shows the sample startup procedure supplied with PSF for microfilm. It contains all the JCL parameters you need to produce a startup procedure for print jobs with 240-pel or 300-pel resolution, but all the references to 300-pel resolution have been commented out. To create separate startup procedures, make two copies of the sample startup procedure for microfilm, and keep one unchanged copy as your startup procedure for 120/240-pel resolution. In the second copy, delete or comment out all the references to 240-pel resolution, and make all the references to 300-pel resolution active. This changed copy is your 300-pel resolution startup procedure.

In Figure 67 on page 256, notice how the PSF PRINTDEV control statements are used:

COMSETUP=*H1SETUPD*

FONTDD=**.FONT01* or **FONTDD=****.FONT02*

OBJCONDD=**.OBJCON01*

[27] Microfilm setup is a type of object container resource. PSF uses a single object container system resource library, designed to contain various types of object container resource.

[28] Although doing so is not recommended, you can drive the microfilm device with just one startup procedure, if either of the following conditions is met:

- All the resources in the resource libraries are resolution-independent, and all the jobs submitted that use inline resources or user libraries contain resources that are resolution-independent; or
- All submitted jobs that request a resolution that is different from the resolution-dependent resources in the PSF system and security libraries must contain the needed resources at the requested resolution, either as inline resources or in the user's library.

[29] Even if a PSF user requests 120-pel resolution for an AFP print job being sent to a microfilm device, PSF requires 240-pel resources from the library. Do not store 120-pel resources in the library for microfilm devices.

Before libraries are referenced by the FONTDD, the Font Conversion Utility needs to be run against them. For additional information, see “Using the Font Library Update (FLU) Utility” on page 259.

```
//APSWPROM PROC
//**************** THE PSF STARTUP PROCEDURE ******************
//*
//*01* MODULE-NAME = APSWPROM
//* $MOD(APSWPROM) COMP(APS) PROD(PSF) : RELEASE 2.2.0
//*
//*01* DESCRIPTIVE-NAME = START PROCEDURE FOR MICROFILM
//*                          DEVICES
//*
//*01* STATUS = VERSION 2, RELEASE 2, LEVEL 0
//*
//*01* FUNCTION = THIS PROCEDURE IS COPIED FROM THE
//*               DISTRIBUTION TO SYS1.PROCLIB. THE 'COPY'
//*               IS PERFORMED BY MACRO 'SGAPS5PR'.
//*
//*01* NOTES = THE FULL NAME OF THE DEFAULT PAGEDEF IS
//*               P1A06462.
//*            THE FULL NAME OF THE DEFAULT FORMDEF IS
//*               F1A10110.
//*            THE FULL NAMES OF THE DEFAULT FONTS ARE
//*                X0GF10, X0GS10, X0TU10, AND X0GU10.
//*            THE FULL NAME OF THE SEPARATOR PAGE PAGEDEF IS
//*                P1V06483.
//*            THE FULL NAME OF THE SEPARATOR PAGE FONT IS
//*                X0GT15.
//*
//*01* REQUIRED ACTIONS =
//*       1) DEFAULT COMSETUP - DEFAULT COMSETUP FILES ARE SPECIFIED
//*            IN THIS STARTUP PROCEDURE BUT ARE NOT PROVIDED WITH
//*            PSF/MVS.  DEFAULT COMSETUP FILES MUST BE GENERATED
//*            WITH SOFTWARE PROVIDED WITH YOUR MICROFILM DEVICE AND
//*            PLACED IN THE DATA SET SPECIFIED ON THE OBJCON01 DD
//*            IN THIS START PROCEDURE.
//*
//*            THE FULL NAME OF THE DEFAULT COMSETUP IS H1SETUPD.
//*            THE FULL NAME OF THE SEPARATOR PAGE COMSETUP IS
//*                H1SETUPS.
//*
//*
//*       2) RESOLUTION - THIS START PROCEDURE IS SET UP FOR DRIVING
//*            THE MICROFILM DEVICE AT 240-PEL RESOLUTION.  TO CHANGE IT
//*            TO DRIVE 300-PEL RESOLUTION MICROFILM DEVICES:
//*            A) COMMENT OUT THE FIRST PRINTDEV STATEMENT AND
//*               UNCOMMENT THE SECOND PRINTDEV STATEMENT.
//*            B) THE FONT02 DATA SET MUST CONTAIN THE CONVERTED
//*               DEFAULT FONTS LISTED IN THE PRINTDEV.
//*               THE FONTS ARE CONVERTED USING THE FONT
//*               CONVERSION UTILITY APSRCF30. SEE THE PSF
//*               SYSTEM PROGRAMMING GUIDE.
//*
//*01* CHANGE-ACTIVITY :
//* $00=OW?????,  HPRF220, 102494 BJ13JPG : Microfilm Support
//*
//**** END OF SPECIFICATIONS ***/
```

Figure 67 (Part 1 of 2). Sample JCL Startup Procedure for a Microfilm Device (APSWPROM)

```
//STEP01   EXEC PGM=APSPPIEP,REGION=4096K
//JOBHDR OUTPUT PAGEDEF=V06483,     /* JOB SEPARATOR PAGEDEF         */
//       FORMDEF=A10110,CHARS=GT15, /* JOB SEPARATOR FORMDEF         */
//       COMSETUP=H1SETUPS          /* JOB SEPARATOR COMSETUP        */
//JOBTLR OUTPUT PAGEDEF=V06483,     /* JOB SEPARATOR PAGEDEF         */
//       FORMDEF=A10110,CHARS=GT15, /* JOB SEPARATOR FORMDEF         */
//       COMSETUP=H1SETUPS          /* JOB SEPARATOR COMSETUP        */
//DSHDR  OUTPUT PAGEDEF=V06483,     /* DATA SET SEPARATOR PAGEDEF    */
//       FORMDEF=A10110,CHARS=GT15, /* DATA SET SEPARATOR FORMDEF    */
//       COMSETUP=H1SETUPS          /* DATA SET SEPARATOR COMSETUP   */
//MSGDS  OUTPUT PAGEDEF=A06462,     /* MESSAGE DATA SET PAGEDEF      */
//       FORMDEF=A10110,            /* MESSAGE DATA SET FORMDEF      */
//       PIMSG=NO,CLASS=A           /* SEND MSG DATA SET TO CLASS A  */
//FONT01   DD   DSN=SYS1.FONTLIBB,  /* SYSTEM FONTS - 240 PEL        */
//         DISP=SHR
//FONT02   DD   DSN=SYS1.FONT300,   /* SYSTEM FONTS - 300 PEL        */
//         DISP=SHR
//PSEG01   DD   DSN=SYS1.PSEGLIB,   /* SYSTEM PAGE SEGMENTS          */
//         DISP=SHR
//OLAY01   DD   DSN=SYS1.OVERLIB,   /* SYSTEM OVERLAYS               */
//         DISP=SHR
//PDEF01   DD   DSN=SYS1.PDEFLIB,   /* SYSTEM PAGEDEFS               */
//         DISP=SHR
//FDEF01   DD   DSN=SYS1.FDEFLIB,   /* SYSTEM FORMDEFS               */
//         DISP=SHR
//OBJCON01 DD   DSN=SYS1.APSOCLIB,  /* SYSTEM OBJECT CONTAINERS      */
//         DISP=SHR                 /*                               */
//PRT1     CNTL
//PRT1     PRINTDEV FONTDD=*.FONT01, /* 240 PEL FONT LIBRARY DD      */
//*                                  /* <-- SEE REQUIRED ACTIONS     */
//*                                  /*     ABOVE                    */
//*PRT1     PRINTDEV FONTDD=*.FONT02, /* 300 PEL FONT LIBRARY DD     */
//*                                   /* <-- SEE REQUIRED ACTIONS    */
//*                                   /*     ABOVE                   */
//         OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD           */
//         PSEGDD=*.PSEG01,         /* SEGMENT  LIBRARY DD           */
//         PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD           */
//         FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD           */
//         OBJCONDD=*.OBJCON01,     /* OBJECT   LIBRARY DD           */
//         JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR OUTPUT   */
//         JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR OUTPUT  */
//         DSHDR=*.DSHDR,           /* DATA SET HEADER OUTPUT        */
//         MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT       */
//         BUFNO=5,                 /* NUMBER OF WRITE DATA BUFFERS  */
//         PAGEDEF=A06462,          /* DEVICE PAGEDEF DEFAULT        */
//         FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT        */
//         COMSETUP=H1SETUPD,       /* DEFAULT COMSETUP MEMBER       */
//         CHARS=(GF10,             /* DEVICE                        */
//         GS10,TU10,GU10),         /* DEFAULT FONT SET              */
//         PIMSG=YES,               /* ACCUMULATE DATA SET MESSAGES  */
//         DATACK=BLOCK,            /* UNBLOCK DATA CHECKS           */
```

Figure 67 (Part 2 of 2). Sample JCL Startup Procedure for a Microfilm Device (APSWPROM)

The startup procedures specify PSF initialization parameters and libraries that contain system and installation resources for microfilm. For more information, see “PSF Startup Procedures for AFP Printers” on page 74.

Redirecting Message Data Sets for Microfilm Jobs

IBM suggests that users *not* print PSF message data sets on microfilm. For more information, see “Redirecting the Message Data Set” on page 115. In the sample provided, APSWPROM, the message data sets are redirected to class A.

Working with Resource Libraries Required by Microfilm Devices

The resources that you use and store in PSF libraries for microfilm devices are the same resources as other printers use, plus microfilm setup resource object containers.

Working with the Microfilm Setup Resource Object Container Library

Microfilm setup is a category of resource called an *object container*,[30] which is stored in a PSF-specified host object container library, in a user library, or inline in the AFP data. You should specify a default COMSETUP parameter in the PRINTDEV statement, and the PSF users can also specify the COMSETUP parameter with JCL; this parameter is described in *Print Services Facility/MVS: Application Programming Guide*. If no default COMSETUP keyword is specified in the PRINTDEV statement, a COMSETUP keyword must be specified in every separator and user data set. If any are encountered without a COMSETUP keyword, PSF terminates.

You should have already installed or created and named resource libraries that contain resources needed for printing to paper media. You must also create and name a resource library to contain the object containers, which are required for printing to microfilm media. No default library is provided with PSF/MVS.

Note: Because a microfilm setup resource is one type of object container resource, and PSF does not enforce a prefix for the 8-character name of the microfilm setup resource, you should define a naming convention to avoid future conflicts with additional object container resources. IBM recommends a prefix of **H1** for all Microfilm setup resources.

Working with Font Libraries

Every time a font is added, deleted, or changed in a library, you must run an IBM command-line utility called the Font Library Update (FLU). For more information, see “Using the Font Library Update (FLU) Utility” on page 259.

Another requirement for printing to microfilm is that font libraries must be available for the resolutions specified. The fonts with 240-pel resolution are used whenever a user specifies 120-pel or 240-pel resolution for microfilm.

The sample startup procedure provided in SYS1.SAMPLIB—APSWPROM, represented by Figure 67 on page 256—identifies both FONT01 (to support 120-pel and 240-pel resolution) and FONT02 (to support 300-pel resolution) for printing to microfilm.

[30] Microfilm setup resources are generated by running a utility, provided with the microfilm device. For more information, see “Generating Microfilm Setup Resource Object Containers” on page 259.

Managing Resources and Selecting Special Processing Parameters

Microfilm support requires the following tasks, in addition to those for setting up printers:

- Using a utility provided with the microfilm device, create microfilm setup resource object containers.
- Using the Font Library Update (FLU) utility, update the font libraries.

Generating Microfilm Setup Resource Object Containers

Run the microfilm utility provided by the manufacturer of your microfilm device to generate microfilm setup resource object containers.[31] For more information about using a utility to generate object containers, refer to the instructions provided with your microfilm device.

The utility provided with the microfilm device checks the syntax of the setup parameters and places the setup information into a microfilm setup resource object container,[32] which is associated, through JCL keywords, with the AFP print job data.

A PSF system library is required for storing the object containers. For more information, see Chapter 4, "Working with Libraries."

Using the Font Library Update (FLU) Utility

Microfilm devices require that all fonts downloaded from the host contain Global Resource ID (GRID) information. If a requested font does not contain GRID information, the page or data set using that font is not printed.

GRID information uniquely describes code page and character set characteristics, such as style, size, set of characters, and valid code points. Font resolution does not affect the GRID value.

IBM and non-IBM fonts usually do not contain GRID information, although space for that information is reserved. The FLU uses system tables and user-modifiable tables to place needed GRID information in fonts that do not contain complete GRID information. The FLU uses the same Name-to-GRID user mapping table (APSRNTGM) that PSF uses (for more information on how to make a Name-to-GRID user mapping table, see "Using the Font Mapping Tables" on page 146). The user mapping table is searched first when insufficient GRID information is found in a code page or character set. The FLU also generates GRID values from the user range of GRID values and from information in the font, if no entry is found for the font in the system or user mapping table and no font member is found that has the same name and contains complete GRID information.

[31] The utility that the manufacturer of your microfilm device provides is also supported by PSF/6000 in the AIX operating system. Microfilm setup resource object containers generated in either of these operating systems can be used by the other operating system.

[32] The object container is not an AFP-managed resource; it is a MO:DCA architecture resource.

Considerations When Using the FLU

Before you run the FLU, read the following key information:

- The FLU does not use user mapping table entries with an FGID or a CPGID that has a value of either X'0000' or X'FFFF'.
- To prevent the FLU from assigning duplicate GRID information to different fonts, you must use a single invocation to run the FLU against *all* fonts that are to be used with the microfilm device[33]—including libraries of different resolutions, font types (for example, raster fonts and outline fonts), system font libraries, security font libraries, font libraries that are to be used by ACIF, and font libraries accessed with the USERLIB keyword.
- If code pages and character sets in different libraries have the same member names, PSF assumes that they are the same code page or character set and will be updated with identical GRID information. If one member contains complete GRID information, and another member with the same name does not, the incomplete member is updated with the additional information.
- When fonts are added or changed on the system after the FLU has run successfully, the FLU must be rerun with all the existing fonts, plus any new fonts. The old fonts are not changed by the FLU on successive runs, but they are used both to propagate GRID information when members with the same name are encountered and to determine what GRID values in the user range are available for assignment.
- The following members of font data sets are not updated by the FLU:
  - Coded fonts
  - Unbounded box fonts (for the 3800 printer)
  - Code pages with complete GRID information
  - Character sets with complete GRID information
  - Non-font resources
  - Non-AFP resource data
- Because the FLU updates fonts in place, no programs should be attempting to access the fonts, and no PSF FSA should be ACTIVE, while the FLU is running.
- The attributes of all font data sets must be as follows (the same requirements as for PSF/MVS):
  - Partitioned (not sequential)
  - Variable (not fixed)
  - Non-spanned
  - Contain carriage control (ANSI or MACHINE)
- If the user-modifiable name-to-grid mapping table, APSRNTGM, is used, it must be link-edited with AMODE=24, RMODE=24.

Running the FLU

A sample job, APSWRFLU, is provided in SYS1.SAMPLIB (see Figure 68 on page 261). A DD statement for each font library on your system that is to be used by the microfilm device must be added to the JCL. Each font DD name must begin with FONT, to identify it to the FLU as the name of a font library. The remaining characters in the DD have no significance to the program. Concatenated data sets for a single DD are not permitted.

[33] The number of font data sets that can be input to the FLU on a single invocation is limited only by the operating system.

```
//APSWRFLU JOB 'account #','name',MSGLEVEL=(1,1)
//* *******************************************************************
//*                                                                   *
//*  USE THIS JOB TO UPDATE ALL FONT LIBRARIES TO BE USED WITH        *
//*  A MICROFILM DEVICE.                                              *
//*                                                                   *
//* *******************************************************************
//STEP1   EXEC PGM=APSRFLU
//SYSPRINT  DD SYSOUT=*
//FONT01    DD DSN=SYS1.FONTLIBB,DISP=OLD
//FONT02    DD DSN=SYS1.FONT300,DISP=OLD
//
```

Figure 68. APSWRFLU Sample Job Provided in SYS1.SAMPLIB

For example, if fonts also existed in COMMON.FONTLIB, PSF.V130.FONTLIB, USER.MY.FONTLIB, and SYS1.SECURE.FONTLIB, the following DDs would be added to the JCL:

```
//FONT03    DD DSN=COMMON.FONTLIB,DISP=OLD
//FONT130   DD DSN=PSF.V130.FONTLIB,DISP=OLD
//FONTUSR1  DD DSN=USER.MY.FONTLIB,DISP=OLD
//FONTSEC   DD DSN=SYS1.SECURE.FONTLIB,DISP=OLD
```

FLU Return Codes

The FLU returns only three valid return codes:

- **0**: The FLU has run successfully, and the appropriate font members have been updated. For information on the updates performed, see the informational messages in the SYSPRINT data set.
- **4**: The FLU updated all the members it could, but found fonts that were invalid and could not be updated. The invalid fonts cannot be used with the microfilm device until they are fixed and the FLU is run again. For information on the font members in error and the updates performed, see the informational messages in the SYSPRINT data set.
- **16**: The FLU updated some or no members. One or more “font” data sets input to the FLU is not a valid font data set. See the informational messages in the SYSPRINT data set for information on the data set in error.

If any other return code is returned, the FLU encountered a logic error. Consult your service representative in the IBM support center. The FLU has error checking and data isolation to avoid contamination of font data in the event of a logic error, and it aborts processing. Fix the error and rerun the FLU.

Printing to Microfilm

After you have attached the microfilm device, provided a startup procedure, run the utility provided by the manufacturer of your microfilm device, and run the FLU utility, users can send AFP data sets to the microfilm device to be printed on microfilm.

Operating a Microfilm Device

If you have two microfilm devices, one with 240-pel resolution and one with 300-pel resolution, you can assign a different destination or class to each of them. You must tell the PSF users what DEST statements to use to direct their output to the destination with the desired resolution.

To switch from one startup procedure (or PRINTDEV) to the other, use the following procedure:

- Drain the active PRINTDEV by using a normal printer stop command—*$PPRTxxx* in JES2, or **V,PRTxxx,OFFLINE* in JES3.
- Start the other PRINTDEV by using a normal printer start command—*$SPRTyyy* in JES2, or **V,PRTyyy,ONLINE* in JES3.

Note: To turn off the microfilm device for normal completion of processing (for example, at the end of the day or for planned maintenance), drain the active PRINTDEV by using a normal printer stop command before powering off the printer.

Some operator commands are not supported for correct operation with microfilm. Tell the operator *not* to use the following:

- JES $I (Interrupt) command
- JES $F (Space Forward) command
- JES $B (Space Backward) command (unless instructed to do so by the microfilm device)

At some point, the microfilm device instructs you to perform operator actions; correct operation is dependent upon your doing so.

Microfilm Data Protection and Security Considerations for PSF

No changes in the procedures for data protection and security considerations have been made for printing to microfilm. The microfilm device is handled in the same way as an AFP1 family printer. For information, see Chapter 12, “Understanding PSF Data Protection and Security.”

Using PSF Installation Exits for Microfilm Devices

One of the tasks of system programmers is writing, changing, and installing exit routines. For more information, see Chapter 17, “Using Installation Exits.”

Understanding PSF Performance Issues That Pertain to Microfilm Devices

IBM recommends that you define no more than two active microfilm devices (PRINTDEVs) in a single startup procedure, because of the large number of pages these devices produce.

Responding to PSF Errors Reported by Microfilm Devices

Handle PSF errors reported by the microfilm device in the same way as you handle PSF errors reported by other printers. For more information, see Chapter 20, “Responding to PSF Errors.”

Recovering or Restarting AFP Print Jobs for Microfilm

The restart and rerun processes have joint PSF and microfilm device interactions. Refer to the documentation provided with your microfilm device. The microfilm device guides you through the restart and rerun process, and instructs the operator to issue commands to PSF when needed.

Appendix A. APSRMARK Sample Jobs

This appendix lists the APSRMARK sample jobs that are shipped with PSF. Table 18 lists the SYS1.SAMPLIB member names and associates each sample job with a corresponding printer resource diskette or cartridge.

Table 18. Sample Jobs in SYS1.SAMPLIB

Member name	**Description**	**Printer that *may* have these resident (some are optional font cards)**
APSWMCPG	AFCCU code pages	Any AFCCU printer, 3930 printer For a list of AFCCU printers, refer to *Advanced Function Presentation: Printer Information*.
APSWMCR	Core Interchange Courier	AFCCU, 3930 printer
APSWMGML	DCF/GML	3812 and 3816 printers
APSWMGR4	3270/IDS, DCF/GML, Language Group 2, Language Group 3, and Language Group 4	3812 and 3816 printers, and any printer with a 3812/3816 compatible font set
APSWMHLV	Core Interchange Helvetica	AFCCU, 3930 printer
APSWMLG2	Language Group 2	3812 and 3816 printers
APSWMLG3	Language Group 3	3812 and 3816 printers
APSWMLG4	Language Group 4	3812 and 3816 printers
APSWMRES	Generic job to mark individual members in a font library as PUBLIC, and replace them	Any printer with resident fonts
APSWMROM	Kanji fonts	3820 printer with ROM Font RPQ, 3130 printer
APSWMRPM	Generic job to mark all members in a font library as PUBLIC	Any printer attached through DPF
APSWMSTD	3270 IDS diskette	3812 and 3816 printers
APSWMTNR	Core Interchange Times New Roman	AFCCU, 3930 printer
APSW4028	4028 resident fonts	4028 printer and any printer with a 4028 compatible font set

Notes:

1. None of the sample jobs listed in Table 18 runs as provided. Each sample job requires modification to the control statements or the JCL.
2. Some of the listings of jobs in the tables include recommended procedures. If the procedures are available, they are given in the comment section at the top of each member.
3. Language Group diskettes or cartridges may not be available in your area. Check with your IBM marketing representative for more information.
4. These sample jobs do not include statements for any font simulations that use font substitutions.

5. For a 300-pel printer, you mut create a 300-pel font library, using APSRCF30, and then mark it with these APSRMARK jobs. For more information, see “Using APSRMARK” on page 133.

Appendix B. Font Resource Tables Supplied by PSF

This appendix contains product-sensitive programming interface information.

The following font resource tables, which are provided with PSF, are intended as samples. The routines as provided may not be suitable for your requirements.

The APSRFTBL font resource table module is provided on the PSF distribution tapes.

APSRFTBL Font Resource Table

The following listing is a copy of the module APSRFTBL, which is shipped with the PSF SAMPLIB. The host code pages and character sets listed in this font resource table are equivalent to the resident symbol sets in the 4224, 4230, 4234, 4247, and 64XX printers, and should be used only with those printers.

PSF searches the table of resident symbol sets to find table entries that match the member names specified in the Map Coded Font structured fields. If PSF finds a match for both the font character set and the code page member names, PSF builds the Load Font Equivalence (LFE) command, using information from the entry in the table of resident symbol sets. If PSF cannot find resident symbol-set table entries that match the names specified in the Map Coded Font structured field, PSF terminates the current page and issues messages saying that the requested resource could not be identified as a resident symbol set.

Unlike other printer-resident font table samples provided, the APSRFTBL font resource table does not represent fonts that can be marked by use of APSRMARK. PSF supports only the resident symbol sets; it does not download fonts to the supported printers.

If you edit any table, do not change its last line (which contains the characters 'FFFFFFFF'); all code for that table must be placed before that last line.

```
       TITLE:   APSRFTBL RESIDENT FONT RESOURCE TABLES
APSRFTBL START 0
*/****START OF SPECIFICATIONS********************EXTPROC**************/
*/*                                                                  */
*/*      MODULE NAME: APSRFTBL                                       */
*/*                                                                  */
*/*      DESCRIPTIVE NAME: RESIDENT FONT RESOURCE TABLES             */
*/*                                                                  */
*/*      FUNCTION: THIS TABLE ENABLES PSF/MVS TO DETERMINE THE       */
*/*                FONT CHARACTER SETS AND CODE PAGES THAT ARE       */
*/*                RESIDENT IN THE PRINTER(S). THE USER MAY UPDATE   */
*/*                THESE TABLES AND CHANGE THE MAPPINGS. IF ANY      */
*/*                CHANGES ARE MADE, THIS FILE MUST BE ASSEMBLED     */
*/*                AND RE-LINKEDITED.                                */
*/*                                                                  */
*/*      NOTES:                                                      */
*/*         DEPENDENCIES: NONE                                       */
*/*         RESTRICTIONS: NONE                                       */
*/*         REGISTER CONVENTIONS: NONE                               */
*/*         PATCH LABEL: NONE                                        */
*/*                                                                  */
*/*      MODULE TYPE:  PROCEDURE                                     */
*/*         PROCESSOR: OS ASSEMBLER H                                */
*/*         MODULE SIZE: 900                                         */
*/*         ATTRIBUTES: READ ONLY, NO EXECUTABLE CODE                */
*/*                                                                  */
*/*                                                                  */
*/*      ENTRY POINT: APSCS24V                                       */
*/*         PURPOSE:  THIS IS THE TABLE OF RESIDENT FONT CHARACTER   */
*/*                   SETS IN THE 4224 OR 4230 PRINTER. IT CAN   @02C*/
*/*                   BE CHANGED BY THE USER.                        */
*/*                   THE TABLE IS TERMINATED BY A STRING OF         */
*/*                   CL8'FFFFFFFF'.                                 */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCP24V                                       */
*/*         PURPOSE:  THIS IS A TABLE OF RESIDENT CODE PAGES IN THE  */
*/*                   4224 OR 4230 PRINTER. IT CAN BE CHANGED BY @02C*/
*/*                   BY THE USER.                                   */
*/*                   THE TABLE IS TERMINATED BY A STRING OF         */
*/*                   CL8'FFFFFFFF'.                                 */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCS34V                                       */
*/*         PURPOSE:  THIS IS THE TABLE OF RESIDENT FONT CHARACTER   */
*/*                   SETS IN THE 4234 PRINTER. IT CAN BE CHANGED    */
*/*                   BY THE USER. THE TABLE IS TERMINATED BY A      */
*/*                   STRING OF CL8'FFFFFFFF'.                       */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCP34V                                       */
*/*         PURPOSE:  THIS IS A TABLE OF RESIDENT CODE PAGES IN THE  */
*/*                   4234 PRINTER. IT CAN BE CHANGED BY THE USER.   */
*/*                   THE TABLE IS TERMINATED BY A STRING OF         */
*/*                   CL8'FFFFFFFF'.                                 */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCS47V                                   @04A*/
*/*         PURPOSE:  THIS IS THE TABLE OF RESIDENT FONT CHARACTER   */
*/*                   SETS IN THE 4247 PRINTER. IT CAN BE CHANGED    */
*/*                   BY THE USER. THE TABLE IS TERMINATED BY A      */
*/*                   STRING OF CL8'FFFFFFFF'.                     */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCP47V                                   @04A*/
*/*         PURPOSE:  THIS IS A TABLE OF RESIDENT CODE PAGES IN THE  */
*/*                   4247 PRINTER. IT CAN BE CHANGED BY THE USER.   */
*/*                   THE TABLE IS TERMINATED BY A STRING OF         */
*/*                   CL8'FFFFFFFF'.                                 */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCS64X                                   @03A*/
*/*         PURPOSE:  THIS IS THE TABLE OF RESIDENT FONT CHARACTER   */
*/*                   SETS IN THE 64XX PRINTER. IT CAN BE CHANGED    */
*/*                   BY THE USER. THE TABLE IS TERMINATED BY A      */
*/*                   A STRING OF CL8'FFFFFFFF'.                     */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      ENTRY POINT: APSCP64X                                   @03A*/
*/*         PURPOSE:  THIS IS A TABLE OF RESIDENT CODE PAGES IN THE  */
*/*                   64XX PRINTER. IT CAN BE CHANGED BY THE USER.   */
*/*                   THE TABLE IS TERMINATED BY A STRING OF         */
*/*                   CL8'FFFFFFFF'.                                 */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*                                                                  */
*/*      TABLES:                                                     */
*/*        APSCS24V - 4224 & 4230 RESIDENT CHAR SET MAPPING TABLE    */
*/*        APSCP24V - 4224 & 4230 RESIDENT CODE PAGE MAPPING TABLE   */
*/*        APSCS34V - 4234 RESIDENT CHAR SET MAPPING TABLE           */
*/*        APSCP34V - 4234 RESIDENT CODE PAGE MAPPING TABLE          */
*/*        APSCS47V - 4247 RESIDENT CHAR SET MAPPING TABLE       @04A*/
*/*        APSCP47V - 4247 RESIDENT CODE PAGE MAPPING TABLE      @04A*/
*/*        APSCS64X - 6408/6412 RESIDENT CHAR SET MAPPING TABLE  @03A*/
*/*        APSCP64X - 6408/6412 RESIDENT CODE PAGE MAPPING TABLE @03A*/
*/*                                                                  */
*/*      MESSAGES: NONE                                              */
*/*                                                                  */
*/*      CHANGE ACTIVITY                                             */
*/*                                                                  */
*/* $00=LAPS0005,HAF1237, 022690, B51PATM: RELEASE 2.1.0         @00A*/
*/* $H7=LAPS0005,HPRF102, 040690, B51PATM: REL 2.1.0 STAGE 2     @H7A*/
*/* $01=OY53959, HPRF102, 060292, BJ13KZB: INCORRECT FGID        @01A*/
*/* $02=OY56466, HPRF102, 080492, BJ13KZB: ADD 4230 CODEPAGES 875 &  */
*/*                                        1026                  @02A*/
*/* $03=OW04104, HPRF220, 040694, BJ13KPC: 6408/6412 SUPPORT     @03A*/
*/* $04=OW19064, HPRF220, 960307, BDKUMLC: 4247 SUPPORT          @04A*/
*/*                                                                  */
*/**********************END OF SPECIFICATIONS************************/
```

```
*/********************************************************************/
*/*                                                                  */
*/*                       BAL CODE FOLLOWS                           */
*/*                                                                  */
*/********************************************************************/
APSRFTBL CSECT                        ENTRY POINT OF THIS MODULE
APSRFTBL AMODE 31                     DO NOT REMOVE/CHANGE
APSRFTBL RMODE ANY                    DO NOT REMOVE/CHANGE
CS4224@  DC    V(APSCS24V)            DO NOT REMOVE/CHANGE
CP4224@  DC    V(APSCP24V)            DO NOT REMOVE/CHANGE
CS4234@  DC    V(APSCS34V)            DO NOT REMOVE/CHANGE
CP4234@  DC    V(APSCP34V)            DO NOT REMOVE/CHANGE
CS4247@  DC    V(APSCS47V)            DO NOT REMOVE/CHANGE        @04A*
CP4247@  DC    V(APSCP47V)            DO NOT REMOVE/CHANGE        @04A*
CS64XX@  DC    V(APSCS64X)            DO NOT REMOVE/CHANGE        @03A*
CP64XX@  DC    V(APSCP64X)            DO NOT REMOVE/CHANGE        @03A*

***********************************************************************
* THE FOLLOWING IS THE CHARACTER SET TABLE FOR THE 4224 PRINTER. THE  *
* FIRST EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE FONT CHARAC- *
* TER SET. THE NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF CORRESPON- *
* DING RESIDENT FONT. THE NEXT TWO (2) BYTE FIELD DEFINES THE FONT    *
* ATTRIBUTES. THE FIRST BYTE IS RESERVED AND THE SECOND BYTE DEFINES  *
* WHETHER THE FONT IS ITALIC, DOUBLE STRIKE, BOLD, OR DOUBLE WIDE.    *
*                                                                     *
* FOR MORE INFORMATION ON THE 4224 RESIDENT FONTS, SEE THE "IBM 4224  *
* PRINTER PRODUCT AND PROGRAMMING DESCRIPTION MANUAL (GC31-2551)."    *
***********************************************************************
         ENTRY APSCS24V                           DO NOT REMOVE/CHANGE
APSCS24V DS    0D                                 DO NOT REMOVE/CHANGE
         DC    CL8'C0L00BOA',XL2'0003',XL2'0000'  OCR-B
         DC    CL8'C0S0CR10',XL2'000B',XL2'0000'  COURIER 10
         DC    CL8'C0S0CB10',XL2'000B',XL2'0002'  COURIER BOLD 10
         DC    CL8'C0S0CI10',XL2'000B',XL2'0008'  COURIER ITALIC 10
         DC    CL8'C0L00AOA',XL2'0013',XL2'0000'  OCR-A
         DC    CL8'C0L0KATA',XL2'001A',XL2'0000'  KATAKANA
         DC    CL8'C0S0AE10',XL2'001A',XL2'0000'  APL 10
         DC    CL8'C0D0GT10',XL2'001A',XL2'0000'  GOTHIC TEXT 10
         DC    CL8'C0D0GB10',XL2'001A',XL2'0002'  GOTHIC BOLD 10
         DC    CL8'C0S0CR12',XL2'0055',XL2'0000'  COURIER 12
         DC    CL8'C0S0CB12',XL2'0055',XL2'0002'  COURIER BOLD 12
         DC    CL8'C0S0CI12',XL2'0055',XL2'0008'  COURIER ITALIC 12
         DC    CL8'C0D0GT12',XL2'0057',XL2'0000'  GOTHIC TEXT 12
         DC    CL8'C0D0GB12',XL2'0057',XL2'0002'  GOTHIC BOLD 12
         DC    CL8'C0D0GI12',XL2'0057',XL2'0008'  GOTHIC ITALIC 12
         DC    CL8'C0S0ESTR',XL2'00A0',XL2'0000'  ESSAY
         DC    CL8'C0S0EBTR',XL2'00A0',XL2'0002'  ESSAY BOLD
         DC    CL8'C0S0EITR',XL2'00A0',XL2'0008'  ESSAY ITALIC
         DC    CL8'C0D0GT15',XL2'00DE',XL2'0000'  GOTHIC TEXT 15
         DC    CL8'C0S0CR15',XL2'00DF',XL2'0000'  COURIER 15
         DC    CL8'C0S0CB15',XL2'00DF',XL2'0002'  COURIER BOLD 15
         DC    CL8'C0S0CI15',XL2'00DF',XL2'0008'  COURIER ITALIC 15
         DC    CL8'C0S0CD15',XL2'00DF',XL2'0001'  COURIER DBL-WIDE 15
         DC    CL8'C0S0CW15',XL2'00DF',XL2'0009'  COURIER DBL-WIDE ITAL
         DC    CL8'FFFFFFFF',XL2'FFFF',XL2'FFFF'  DO NOT REMOVE/CHANGE
```

```
***********************************************************************
* THE FOLLOWING IS THE CODE PAGE TABLE FOR THE 4224 PRINTER. THE FIRST*
* EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE CODE PAGE. THE     *
* NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF THE CORRESPONDING       *
* RESIDENT CODE PAGE.                                                 *
*                                                                     *
* FOR MORE INFORMATION ON THE 4224 RESIDENT FONTS, SEE THE "IBM 4224  *
* PRINTER PRODUCT AND PROGRAMMING DESCRIPTION MANUAL (GC31-2551)."    *
***********************************************************************
         ENTRY APSCP24V                           DO NOT REMOVE/CHANGE
APSCP24V DS    0D                                 DO NOT REMOVE/CHANGE
         DC    CL8'T1D0BASE',XL2'03EA'            CODE PAGE 1002
         DC    CL8'T1L038BA',XL2'03EA'            CODE PAGE 1002
         DC    CL8'T1V10037',XL2'0025'            CODE PAGE 37
         DC    CL8'T1V10273',XL2'0111'            CODE PAGE 273
         DC    CL8'T1V10274',XL2'0112'            CODE PAGE 274
         DC    CL8'T1V10275',XL2'0113'            CODE PAGE 275
         DC    CL8'T1V10277',XL2'0115'            CODE PAGE 277
         DC    CL8'T1V10278',XL2'0116'            CODE PAGE 278
         DC    CL8'T1V10280',XL2'0118'            CODE PAGE 280
         DC    CL8'T1V10281',XL2'0119'            CODE PAGE 281
         DC    CL8'T1V10282',XL2'011A'            CODE PAGE 282
         DC    CL8'T1V10284',XL2'011C'            CODE PAGE 284
         DC    CL8'T1V10285',XL2'011D'            CODE PAGE 285
         DC    CL8'T1000290',XL2'0122'            CODE PAGE 290
         DC    CL8'T1L02773',XL2'0122'            CODE PAGE 290 SUBSET
         DC    CL8'T1L02774',XL2'0122'            CODE PAGE 290 SUBSET
         DC    CL8'T1S0AE10',XL2'0125'            CODE PAGE 293
         DC    CL8'T1V10297',XL2'0129'            CODE PAGE 297
         DC    CL8'T1L0OCR1',XL2'0154'            CODE PAGE 340
         DC    CL8'T1L0OCRB',XL2'0154'            CODE PAGE 340
         DC    CL8'T1V10500',XL2'01F4'            CODE PAGE 500
***********************************************************************

***********************************************************************
* THE FOLLOWING CODE PAGES 875 AND 1026 ARE TO BE USED WITH THE   @02A*
* 4230 PRINTER ONLY.                                              @02A*
***********************************************************************
         DC    CL8'T1000875',XL2'036B'            CODE PAGE 875   @02A
         DC    CL8'T1001026',XL2'0402'            CODE PAGE 1026  @02A
         DC    CL8'FFFFFFFF',XL2'FFFF'            DO NOT REMOVE/CHANGE
```

```
***********************************************************************
* THE FOLLOWING IS THE CHARACTER SET TABLE FOR THE 4234 PRINTER. THE  *
* FIRST EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE FONT CHARAC- *
* TER SET. THE NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF CORRESPON- *
* DING RESIDENT FONT. THE NEXT TWO (2) BYTE FIELD DEFINES THE FONT    *
* ATTRIBUTES. THE FIRST BYTE IS RESERVED AND THE SECOND BYTE DEFINES  *
* WHETHER THE FONT IS ITALIC, DOUBLE STRIKE, BOLD, OR DOUBLE WIDE.    *
*                                                                     *
* FOR MORE INFORMATION ON THE 4234 RESIDENT FONTS, SEE THE "IBM 4234  *
* PRINTER PRODUCT AND PROGRAMMING DESCRIPTION MANUAL (GC31-2554)."    *
***********************************************************************
         ENTRY APSCS34V                            DO NOT REMOVE/CHANGE
APSCS34V DS    0D                                  DO NOT REMOVE/CHANGE
         DC    CL8'C0L00BOA',XL2'0003',XL2'0000'  OCR-B
         DC    CL8'C0S0CR10',XL2'000B',XL2'0000'  COURIER 10
         DC    CL8'C0S0CB10',XL2'000B',XL2'0002'  COURIER BOLD 10
         DC    CL8'C0S0CI10',XL2'000B',XL2'0008'  COURIER ITALIC 10
         DC    CL8'C0L00AOA',XL2'0013',XL2'0000'  OCR-A
         DC    CL8'C0L0KATA',XL2'001A',XL2'0000'  KATAKANA
         DC    CL8'C0S0AE10',XL2'001A',XL2'0000'  APL 10
         DC    CL8'C0D0GT10',XL2'001A',XL2'0000'  GOTHIC TEXT 10
         DC    CL8'C0D0GB10',XL2'001A',XL2'0002'  GOTHIC BOLD 10
         DC    CL8'C0S0CR12',XL2'0055',XL2'0000'  COURIER 12
         DC    CL8'C0S0CB12',XL2'0055',XL2'0002'  COURIER BOLD 12
         DC    CL8'C0S0CI12',XL2'0055',XL2'0008'  COURIER ITALIC 12
         DC    CL8'C0D0GT12',XL2'0057',XL2'0000'  GOTHIC TEXT 12
         DC    CL8'C0D0GB12',XL2'0057',XL2'0002'  GOTHIC BOLD 12
         DC    CL8'C0D0GI12',XL2'0057',XL2'0008'  GOTHIC ITALIC 12
         DC    CL8'C0S0ESTR',XL2'00A0',XL2'0000'  ESSAY
         DC    CL8'C0S0EBTR',XL2'00A0',XL2'0002'  ESSAY BOLD
         DC    CL8'C0S0EITR',XL2'00A0',XL2'0008'  ESSAY ITALIC
         DC    CL8'C0S0D224',XL2'00CC',XL2'0000'  GOTHIC TEXT 13   @01C
         DC    CL8'C0S0D225',XL2'00CC',XL2'0000'  GOTHIC TEXT 13   @01C
         DC    CL8'C0S0D226',XL2'00CC',XL2'0002'  GOTHIC BOLD 13   @01C
         DC    CL8'C0S0D227',XL2'00CC',XL2'0008'  GOTHIC ITAL 13   @01C
         DC    CL8'C0D0GT15',XL2'00DE',XL2'0000'  GOTHIC TEXT 15
         DC    CL8'C0S0CR15',XL2'00DF',XL2'0000'  COURIER 15
         DC    CL8'C0S0CB15',XL2'00DF',XL2'0002'  COURIER BOLD 15
         DC    CL8'C0S0CI15',XL2'00DF',XL2'0008'  COURIER ITALIC 15
         DC    CL8'C0S0CD15',XL2'00DF',XL2'0001'  COURIER DBL-WIDE 15
         DC    CL8'C0S0CW15',XL2'00DF',XL2'0009'  COURIER DBL-WIDE ITAL
         DC    CL8'FFFFFFFF',XL2'FFFF',XL2'FFFF'  DO NOT REMOVE/CHANGE
```

```
***********************************************************************
* THE FOLLOWING IS THE CODE PAGE TABLE FOR THE 4234 PRINTER. THE FIRST*
* EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE CODE PAGE. THE     *
* NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF THE CORRESPONDING       *
* RESIDENT CODE PAGE.                                                 *
*                                                                     *
* FOR MORE INFORMATION ON THE 4234 RESIDENT FONTS, SEE THE "IBM 4234  *
* PRINTER PRODUCT AND PROGRAMMING DESCRIPTION MANUAL (GC31-2554)."    *
***********************************************************************
         ENTRY APSCP34V                           DO NOT REMOVE/CHANGE
APSCP34V DS    0D                                 DO NOT REMOVE/CHANGE
         DC    CL8'T1D0BASE',XL2'03EA'            CODE PAGE 1002
         DC    CL8'T1L038BA',XL2'03EA'            CODE PAGE 1002
         DC    CL8'T1V10037',XL2'0025'            CODE PAGE 37
         DC    CL8'T1V10273',XL2'0111'            CODE PAGE 273
         DC    CL8'T1V10274',XL2'0112'            CODE PAGE 274
         DC    CL8'T1V10275',XL2'0113'            CODE PAGE 275
         DC    CL8'T1V10277',XL2'0115'            CODE PAGE 277
         DC    CL8'T1V10278',XL2'0116'            CODE PAGE 278
         DC    CL8'T1V10280',XL2'0118'            CODE PAGE 280
         DC    CL8'T1V10281',XL2'0119'            CODE PAGE 281
         DC    CL8'T1V10282',XL2'011A'            CODE PAGE 282
         DC    CL8'T1V10284',XL2'011C'            CODE PAGE 284
         DC    CL8'T1V10285',XL2'011D'            CODE PAGE 285
         DC    CL8'T1000290',XL2'0122'            CODE PAGE 290
         DC    CL8'T1L02773',XL2'0122'            CODE PAGE 290 SUBSET
         DC    CL8'T1L02774',XL2'0122'            CODE PAGE 290 SUBSET
         DC    CL8'T1S0AE10',XL2'0125'            CODE PAGE 293
         DC    CL8'T1V10297',XL2'0129'            CODE PAGE 297
         DC    CL8'T1000310',XL2'0136'            CODE PAGE 310
         DC    CL8'T1000420',XL2'01A4'            CODE PAGE 420
         DC    CL8'T1000423',XL2'01A7'            CODE PAGE 423
         DC    CL8'T1000424',XL2'01A8'            CODE PAGE 424
         DC    CL8'T1V10500',XL2'01F4'            CODE PAGE 500
         DC    CL8'T1000838',XL2'0346'            CODE PAGE 838
         DC    CL8'T1000870',XL2'0366'            CODE PAGE 870
         DC    CL8'T1V10871',XL2'0367'            CODE PAGE 871
         DC    CL8'T1L0OCR1',XL2'037C'            CODE PAGE 892
         DC    CL8'T1L0OCRB',XL2'037D'            CODE PAGE 893
         DC    CL8'FFFFFFFF',XL2'FFFF'            DO NOT REMOVE/CHANGE
```

```
****************************************************************
* THE FOLLOWING IS THE CHARACTER SET TABLE FOR THE 4247        *
* PRINTER. THE FIRST EIGHT (8) BYTES OF EACH ENTRY IS THE      *
* NAME OF THE FONT CHARACTER SET. THE NEXT TWO (2) BYTE        *
* FIELD IS THE GLOBAL ID OF * CORRESPONDING RESIDENT FONT.     *
* THE NEXT TWO (2) BYTE FIELD DEFINES THE * FONT ATTRIBUTES.   *
* THE FIRST BYTE IS RESERVED AND THE SECOND BYTE DEFINES       *
* WHETHER THE FONT IS ITALIC, DOUBLE STRIKE, BOLD, OR DOUBLE   *
* WIDE.                                                    @04P*
****************************************************************
         ENTRY APSCS47V                     DO NOT REMOVE/CHANGE
APSCS47V DS 0D                              DO NOT REMOVE/CHANGE
         DC CL8'C0L00BOA',XL2'0003',XL2'0000' OCR-B         @04P
         DC CL8'C0S0CR10',XL2'000B',XL2'0000' COURIER 10    @04P
         DC CL8'C0S0CB10',XL2'000B',XL2'0002' COURIERBOLD10 @04P
         DC CL8'C0S0CI10',XL2'000B',XL2'0008' COURIERITAL10 @04P
         DC CL8'C0L00AOA',XL2'0013',XL2'0000' OCR-A         @04P
         DC CL8'C0L0KATA',XL2'001A',XL2'0000' KATAKANA      @04P
         DC CL8'C0S0AE10',XL2'001A',XL2'0000' APL 10        @04P
         DC CL8'C0D0GT10',XL2'001A',XL2'0000' GOTHIC TEXT10 @04P
         DC CL8'C0D0GB10',XL2'001A',XL2'0002' GOTHIC BOLD10 @04P
         DC CL8'C0S0CR12',XL2'0055',XL2'0000' COURIER 12    @04P
         DC CL8'C0S0CB12',XL2'0055',XL2'0002' COURIER BOLD12@04P
         DC CL8'C0S0CI12',XL2'0055',XL2'0008' COURIER ITAL12@04P
         DC CL8'C0D0GT12',XL2'0057',XL2'0000' GOTHIC TEXT 12@04P
         DC CL8'C0D0GB12',XL2'0057',XL2'0002' GOTHIC BOLD 12@04P
         DC CL8'C0D0GI12',XL2'0057',XL2'0008' GOTHIC ITAL 12@04P
         DC CL8'C0S0ESTR',XL2'00A0',XL2'0000' ESSAY         @04P
         DC CL8'C0S0EBTR',XL2'00A0',XL2'0002' ESSAY BOLD    @04P
         DC CL8'C0S0EITR',XL2'00A0',XL2'0008' ESSAY ITALIC  @04P
         DC CL8'C0D0GT15',XL2'00DE',XL2'0000' GOTHIC TXT 15 @04P
         DC CL8'C0S0CR15',XL2'00DF',XL2'0000' COURIER 15    @04P
         DC CL8'C0S0CB15',XL2'00DF',XL2'0002' COURIER BOLD15@04P
         DC CL8'C0S0CI15',XL2'00DF',XL2'0008' COURIER ITAL15@04P
         DC CL8'C0S0CD15',XL2'00DF',XL2'0001' COURDBL-WID 15@04P
         DC CL8'C0S0CW15',XL2'00DF',XL2'0009' COURDBL-WIDITA@04P
         DC CL8'FFFFFFFF',XL2'FFFF',XL2'FFFF' DO NOT REMOVE
*
```

```
****************************************************************
* THE FOLLOWING IS THE CODE PAGE TABLE FOR THE 4247 PRINTER.   *
* THE FIRST EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE   *
* CODE PAGE. THE NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF   *
* THE CORRESPONDING RESIDENT PAGE.                         @04P*
****************************************************************
         ENTRY APSCP47V
APSCP47V DS    0D                           DO NOT REMOVE/CHANGE
         DC    CL8'T1D0BASE',XL2'03EA'       CODE PAGE 1002 @04P
         DC    CL8'T1L038BA',XL2'03EA'       CODE PAGE 1002 @04P
         DC    CL8'T1GDP256',XL2'0100'        CODE PAGE 256 @04A
         DC    CL8'T1V10037',XL2'0025'        CODE PAGE  37 @04P
         DC    CL8'T1V10273',XL2'0111'        CODE PAGE 273 @04P
         DC    CL8'T1V10274',XL2'0112'        CODE PAGE 274 @04P
         DC    CL8'T1V10275',XL2'0113'        CODE PAGE 275 @04P
         DC    CL8'T1V10277',XL2'0115'        CODE PAGE 277 @04P
         DC    CL8'T1V10278',XL2'0116'        CODE PAGE 278 @04P
         DC    CL8'T1V10280',XL2'0118'        CODE PAGE 280 @04P
         DC    CL8'T1V10281',XL2'0119'        CODE PAGE 281 @04P
         DC    CL8'T1V10282',XL2'011A'        CODE PAGE 282 @04P
         DC    CL8'T1V10284',XL2'011C'        CODE PAGE 284 @04P
         DC    CL8'T1V10285',XL2'011D'        CODE PAGE 285 @04P
         DC    CL8'T1000290',XL2'0122'        CODE PAGE 290 @04P
         DC    CL8'T1L02773',XL2'0122'    CODE PAGE 290 SUB @04P
         DC    CL8'T1L02774',XL2'0122'    CODE PAGE 290 SUB @04P
         DC    CL8'T1S0AE10',XL2'0125'        CODE PAGE 293 @04P
         DC    CL8'T1V10297',XL2'0129'        CODE PAGE 297 @04P
         DC    CL8'T1000310',XL2'0136'        CODE PAGE 310 @04P
         DC    CL8'T1000420',XL2'01A4'        CODE PAGE 420 @04P
         DC    CL8'T1000423',XL2'01A7'        CODE PAGE 423 @04P
         DC    CL8'T1000424',XL2'10A8'        CODE PAGE 424 @04P
         DC    CL8'T1V10500',XL2'01F4'        CODE PAGE 500 @04A
         DC    CL8'T1000803',XL2'0323'        CODE PAGE 803 @04A
         DC    CL8'T1000838',XL2'0346'        CODE PAGE 838 @04P
         DC    CL8'T1000870',XL2'0366'        CODE PAGE 870 @04P
         DC    CL8'T1V10871',XL2'0367'        CODE PAGE 871 @04P
         DC    CL8'T1000875',XL2'036B'        CODE PAGE 875 @04A
         DC    CL8'T1000880',XL2'0370'        CODE PAGE 880 @04A
         DC    CL8'T1000892',XL2'037C'        CODE PAGE 892 @04P
         DC    CL8'T1L0OCR1',XL2'037C'        CODE PAGE 892 @04P
         DC    CL8'T1000893',XL2'037D'        CODE PAGE 893 @04P
         DC    CL8'T1L0OCRB',XL2'037D'        CODE PAGE 893 @04P
         DC    CL8'T1001025',XL2'0401'       CODE PAGE 1025 @04A
         DC    CL8'T1001026',XL2'0402'       CODE PAGE 1026 @04A
         DC    CL8'FFFFFFFF',XL2'FFFF' DO NOT REMOVE/CHANGE
```

```
***********************************************************************
* THE FOLLOWING IS THE CHARACTER SET TABLE FOR THE 64XX PRINTER. THE  *
* FIRST EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE FONT CHARAC- *
* TER SET. THE NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF CORRESPON- *
* DING RESIDENT FONT. THE NEXT TWO (2) BYTE FIELD DEFINES THE FONT    *
* ATTRIBUTES. THE FIRST BYTE IS RESERVED AND THE SECOND BYTE DEFINES  *
* WHETHER THE FONT IS ITALIC, DOUBLE STRIKE, BOLD, OR DOUBLE WIDE.@03C*
***********************************************************************
         ENTRY APSCS64X                           DO NOT REMOVE/CHANGE
APSCS64X DS    0D                                 DO NOT REMOVE/CHANGE
         DC    CL8'C0L00BOA',XL2'0003',XL2'0000'  OCR-B            @03P
         DC    CL8'C0S0CR10',XL2'000B',XL2'0000'  COURIER 10       @03P
         DC    CL8'C0S0CB10',XL2'002E',XL2'0000'  COURIER BOLD 10  @03P
         DC    CL8'C0S0CI10',XL2'0012',XL2'0000'  COURIER ITAL 10  @03P
         DC    CL8'C0L00AOA',XL2'0013',XL2'0000'  OCR-A            @03P
         DC    CL8'C0L0KATA',XL2'001A',XL2'0000'  KATAKANA         @03P
         DC    CL8'C0S0AE10',XL2'001A',XL2'0000'  APL 10           @03P
         DC    CL8'C0D0GT10',XL2'001A',XL2'0000'  GOTHIC TEXT 10   @03P
         DC    CL8'C0D0GB10',XL2'001A',XL2'0002'  GOTHIC BOLD 10   @03P
         DC    CL8'C0S0CR12',XL2'0055',XL2'0000'  COURIER 12       @03P
         DC    CL8'C0S0CB12',XL2'006C',XL2'0000'  COURIER BOLD 12  @03P
         DC    CL8'C0S0LB12',XL2'006E',XL2'0000'  LET GOTH BOLD 12 @03A
         DC    CL8'C0S0CI12',XL2'0055',XL2'0008'  COURIER ITAL 12  @03P
         DC    CL8'C0D0GT12',XL2'0057',XL2'0000'  GOTHIC TEXT 12   @03P
         DC    CL8'C0D0GB12',XL2'0057',XL2'0002'  GOTHIC BOLD 12   @03P
         DC    CL8'C0D0GI12',XL2'0057',XL2'0008'  GOTHIC ITALIC 12 @03P
         DC    CL8'C0S0ESTR',XL2'00A0',XL2'0000'  ESSAY            @03P
         DC    CL8'C0S0EBTR',XL2'00A3',XL2'0000'  ESSAY BOLD       @03P
         DC    CL8'C0S0EITR',XL2'00A2',XL2'0000'  ESSAY ITALIC     @03P
         DC    CL8'C0D0GT13',XL2'00CC',XL2'0000'  GOTHIC TEXT 13   @03A
         DC    CL8'C0S0D224',XL2'00CC',XL2'0000'  GOTHIC TEXT 13   @03P
         DC    CL8'C0S0D225',XL2'00CC',XL2'0000'  GOTHIC TEXT 13   @03P
         DC    CL8'C0S0D226',XL2'00CC',XL2'0002'  GOTHIC BOLD 13   @03P
         DC    CL8'C0S0D227',XL2'00CC',XL2'0008'  GOTHIC ITAL 13   @03P
         DC    CL8'C0D0GT15',XL2'00DE',XL2'0000'  GOTHIC TEXT 15   @03P
         DC    CL8'C0S0CR15',XL2'00DF',XL2'0000'  COURIER 15       @03P
         DC    CL8'C0S0CB15',XL2'00DF',XL2'0002'  COURIER BOLD 15  @03P
         DC    CL8'C0S0CI15',XL2'00DF',XL2'0008'  COURIER ITAL 15  @03P
         DC    CL8'C0S0CD15',XL2'00DF',XL2'0001'  COUR DBL-WID 15  @03P
         DC    CL8'C0S0CW15',XL2'00DF',XL2'0009'  COUR DBL-WID ITAL@03P
         DC    CL8'C0D0GT18',XL2'0102',XL2'0000'  GOTHIC TEXT 18   @03A
         DC    CL8'FFFFFFFF',XL2'FFFF',XL2'FFFF'  DO NOT REMOVE/CHANGE
```

```
***********************************************************************
* THE FOLLOWING IS THE CODE PAGE TABLE FOR THE 64XX PRINTER. THE FIRST*
* EIGHT (8) BYTES OF EACH ENTRY IS THE NAME OF THE CODE PAGE. THE     *
* NEXT TWO (2) BYTE FIELD IS THE GLOBAL ID OF THE CORRESPONDING       *
* RESIDENT CODE PAGE.                                             @03C*
***********************************************************************
         ENTRY APSCP64X                           DO NOT REMOVE/CHANGE
APSCP64X DS    0D                                 DO NOT REMOVE/CHANGE
         DC    CL8'T1D0BASE',XL2'03EA'            CODE PAGE 1002   @03P
         DC    CL8'T1L038BA',XL2'03EA'            CODE PAGE 1002   @03P
         DC    CL8'T1GDP256',XL2'0100'            CODE PAGE 256    @03A
         DC    CL8'T1V10037',XL2'0025'            CODE PAGE 37     @03P
         DC    CL8'T1V10273',XL2'0111'            CODE PAGE 273    @03P
         DC    CL8'T1V10274',XL2'0112'            CODE PAGE 274    @03P
         DC    CL8'T1V10275',XL2'0113'            CODE PAGE 275    @03P
         DC    CL8'T1V10277',XL2'0115'            CODE PAGE 277    @03P
         DC    CL8'T1V10278',XL2'0116'            CODE PAGE 278    @03P
         DC    CL8'T1V10280',XL2'0118'            CODE PAGE 280    @03P
         DC    CL8'T1V10281',XL2'0119'            CODE PAGE 281    @03P
         DC    CL8'T1V10282',XL2'011A'            CODE PAGE 282    @03P
         DC    CL8'T1V10284',XL2'011C'            CODE PAGE 284    @03P
         DC    CL8'T1V10285',XL2'011D'            CODE PAGE 285    @03P
         DC    CL8'T1000290',XL2'0122'            CODE PAGE 290    @03P
         DC    CL8'T1L02773',XL2'0122'            CODE PAGE 290 SUB@03P
         DC    CL8'T1L02774',XL2'0122'            CODE PAGE 290 SUB@03P
         DC    CL8'T1S0AE10',XL2'0125'            CODE PAGE 293    @03P
         DC    CL8'T1V10297',XL2'0129'            CODE PAGE 297    @03P
         DC    CL8'T1000310',XL2'0136'            CODE PAGE 310    @03P
         DC    CL8'T1000420',XL2'01A4'            CODE PAGE 420    @03P
         DC    CL8'T1000423',XL2'01A7'            CODE PAGE 423    @03P
         DC    CL8'T1000424',XL2'01A8'            CODE PAGE 424    @03P
         DC    CL8'T1V10500',XL2'01F4'            CODE PAGE 500    @03P
         DC    CL8'T1000803',XL2'0323'            CODE PAGE 803    @03A
         DC    CL8'T1000838',XL2'0346'            CODE PAGE 838    @03P
         DC    CL8'T1000870',XL2'0366'            CODE PAGE 870    @03P
         DC    CL8'T1V10871',XL2'0367'            CODE PAGE 871    @03P
         DC    CL8'T1000875',XL2'036B'            CODE PAGE 875    @03A
         DC    CL8'T1000880',XL2'0370'            CODE PAGE 880    @03A
         DC    CL8'T1000892',XL2'037C'            CODE PAGE 892    @03P
         DC    CL8'T1L0OCR1',XL2'037C'            CODE PAGE 892    @03P
         DC    CL8'T1000893',XL2'037D'            CODE PAGE 893    @03A
         DC    CL8'T1L0OCRB',XL2'037D'            CODE PAGE 893    @03P
         DC    CL8'T1000905',XL2'0389'            CODE PAGE 905    @03A
         DC    CL8'T1001025',XL2'0401'            CODE PAGE 1025   @03A
         DC    CL8'T1001026',XL2'0402'            CODE PAGE 1026   @03A
         DC    CL8'FFFFFFFF',XL2'FFFF'            DO NOT REMOVE/CHANGE
         END   APSRFTBL                           DO NOT REMOVE/CHANGE
```

Appendix C. Field Descriptions for APSUX07

This appendix describes APSUX07 parameter areas and fields, which contain product-sensitive programming-interface information. The appendix presents the following:

- Descriptions of parameter areas
- A control block listing for each parameter area
- A description of each field in each parameter area
- A processing indicator cross-reference, which you can use to match fields with exit calls

XTP7 Parameter Area (DSECT)

XTP7, a DSECT (dummy section) in APSGEXTP, contains information that APSUX07 needs in order to determine what action to take. The XTPRECP pointer in APSGEXTP provides the address of XTP7. XTP7 is mapped within macro APSGEXTP. For more information on APSGEXTP, see Figure 54 on page 167.

All of the fields in XTP7 can be read by the exit; however, most of those fields are not intended to be written to by the exit, and some fields do not always contain information about the exit call in progress. PSF looks for changes to specific fields only after certain calls to the exit are made.

Figure 69 on page 280 shows the format of XTP7.

```
Offsets  Type       Length   Name           Description

   0    (0) CHARACTER     *  XTP7
   0    (0) CHARACTER     4  XTP7CID        CONTROL BLOCK ID = 'XTP7'
   4    (4) SIGNED        4  XTP7LENG       LENGTH OF XTP7 AREA
   8    (8) BITSTRING     2  XTP7ETYP       FLAGS SET BY PSF-INDICATES
                                            TO APSUX07 THE TYPE OF EXIT
                                            CALL BEING MADE.
        1... ....            XTP7INIT       INITIALIZATION EXIT
        .1.. ....            XTP7BDS        BEGINNING OF DATA SET
        ..1. ....            XTP7ACC        RESOURCE ACCESS TIME EXIT
        ...1 ....            XTP7LDB        LOAD BEGIN EXIT
        .... 1...            XTP7LDE        LOAD END EXIT
        .... .1..            XTP7DSE        DELETION AT DATA SET END
        .... ..1.            XTP7TERM       TERMINATION OF FSA
        .... ...1            *              RESERVED

        1111 1111            *              RESERVED
  10    (A) BITSTRING     2  XTP7RTYP       FLAGS SET BY PSF-INDICATES
                                            TO APSUX07 THE TYPE OF
                                            RESOURCE CURRENTLY BEING
                                            PROCESSED
        1... ....            XTP7PD         PAGEDEF
        .1.. ....            XTP7FD         FORMDEF
        ..1. ....            XTP7CF         CODED FONT
        ...1 ....            XTP7MO         OVERLAY
        .... 1...            XTP7PS         PAGE SEGMENT
        .... .1..            XTP7OC         OBJECT CONTAINER
        .... ..11            *              RESERVED
        1111 1111            *              RESERVED
  12    (C) BITSTRING     1  XTP7DSTY       FLAGS SET BY PSF-INDICATES
                                            TO APSUX07 THE TYPE OF DATA
                                            SET CURRENTLY BEING PROCESSED
        1... ....            XTP7AUX        PROCESSING SYSTEM DEFAULT OR
                                            FOR AUXILIARY DATA SETS
        .1.. ....            XTP7PDS        PRINT DATA SET
        ..11 1111            *              RESERVED
  13    (D) BITSTRING     1  XTP7DSAT       FLAGS SET BY PSF-ATTRIBUTES
                                            OF THE DATA SET CURRENTLY
                                            BEING PROCESSED
        1... ....            XTP7PDFT       SYSTEM DEFAULT
        .1.. ....            XTP7PJHD       JOB HEADER
        ..1. ....            XTP7PJTR       JOB TRAILER
        ...1 ....            XTP7PDSH       DATA SET HEADER
        .... 1...            XTP7PMDS       MESSAGE DATA SET
        .... .1..            XTP7PLDS       LAST PRINT DATA SET IN
                                            JOB VALID IN BDS & DSE
                                            EXITS ONLY
        .... ..11            *              RESERVED
  14    (E) UNSIGNED      1  XTP7ACSE       AUTOMATIC CUT-SHEET
                                            EMULATION CHECKING.
                                            VALID FOR INITIALIZATION
                                            'XTP7INIT' AND BDSC
                                            'XTP7BDS'.
  15    (F) UNSIGNED      1  XTP7IPCC       NUMBER OF COPIES OF
                                            PAGE TO PRODUCE.
                                            - VALID ONLY WITH
                                            THE INTERRUPT MESSAGE
                                            XTP7IPMF.
  16   (10) BITSTRING     4  XTP7NOTY       NOTIFICATION FLAGS SET BY
                                            APSUX07, INDICATES TO PSF WHEN
                                            THE EXIT WANTS CONTROL
  16   (10) BITSTRING     1  XTP7MISC       MISCELLANEOUS TYPE EXITS
        1... ....            XTP7EBDS       BEGINNING OF DATA SET
        .1.. ....            XTP7ETRM       TERMINATION OF FSA
        ..11 1111            *              RESERVED
```

Figure 69 (Part 1 of 5). XTP7 Parameter Area Format

```
    17   (11) BITSTRING     1  XTP7NACC       ACCESS TIME EXITS
          1...  ....           XTP7APD        PAGEDEFS
          .1..  ....           XTP7AFD        FORMDEFS
          ..1.  ....           XTP7ACF        CODED FONTS
          ...1  ....           XTP7AMO        MEDIUM OVERLAYS
          ....  1...           XTP7APS        PAGE SEGMENTS
          ....  .1..           XTP7AOC        OBJECT CONTAINERS
          ....  ..11           *              RESERVED
    18   (12) BITSTRING     1  XTP7NLD        LOAD TIME EXITS
          1...  ....           XTP7LPD        PAGEDEFS
          .1..  ....           XTP7LFD        FORMDEFS
          ..1.  ....           XTP7LCF        CODED FONTS
          ...1  ....           XTP7LMO        MEDIUM OVERLAYS
          ....  1...           XTP7LPS        PAGE SEGMENTS
          ....  .1..           XTP7LOC        OBJECT CONTAINERS
          ....  ..11           *              RESERVED
    19   (13) BITSTRING     1  XTP7NDEL       DELETE TIME (DSE) EXITS
          1...  ....           XTP7DPD        PAGEDEFS
          .1..  ....           XTP7DFD        FORMDEFS
          ..1.  ....           XTP7DCF        CODED FONTS
          ...1  ....           XTP7DMO        MEDIUM OVERLAYS
          ....  1...           XTP7DPS        PAGE SEGMENTS
          ....  .1..           XTP7DOC        OBJECT CONTAINERS
          ....  ..11           *              RESERVED
    20   (14) BITSTRING     2  XTP7RFLG       RESPONSE FLAGS
          1...  ....           XTP7ETFF       EJECT TO FRONT FACING BETWEEN
                                              DOCUMENTS:
                                              0 = ETFF ACTIVE
                                              1 = BYPASS ETFF
          .1..  ....           XTP7TDS        TERMINATE THE DATA SET
                                              0 = DO NOT TERMINATE
                                              1 = TERMINATE
          ..1.  ....           XTP7EFFH       EJECT TO FRONT FACING BEFORE
                                              JOB HEADER:
                                              0 = ETFF ACTIVE
                                              1 = BYPASS ETFF
          ...1  ....           XTP7HDF        SELECT HARDWARE DEFAULT FONT
                                              (VALID FOR 3800 ONLY):
                                              0 = USE HARDWARE DEFAULT
                                              1 = USE 1ST FONT MAPPED
          ....  1...           XTP7HQUE       SEND DATA SET TO JES AND
                                              MARK IT AS UNPRINTABLE
                                              (VALID ONLY WHEN DATA SET
                                              TERMINATED BY XTP7TDS):
                                              0 = PURGE THE DATA SET
                                              1 = MARK DATA SET AS UNPRINTABLE
          ....  .1..           XTP7PRNS       SINGLE BYTE FONT PRUNING FLAG
                                              (VALID ONLY ON INITIALIZATION
                                              CALL 'XTP7INIT')
                                              0 = PRUNE SINGLE BYTE FONTS
                                              1 = BYPASS PRUNING
          ....  ..1.           XTP7PRND       DOUBLE BYTE FONT PRUNING FLAG
                                              (VALID ONLY ON INITIALIZATION
                                              CALL 'XTP7INIT'
                                              0 = PRUNE DOUBLE BYTE FONTS
                                              1 = BYPASS PRUNING
          .... ...1            XTP7IPMF       MARK FORM REQUEST
                                              (VALID FOR INITIALIZATION
                                              'XTP7INIT' AND BDSC 'XTP7BDS')
                                              SEE XTP7IPCC FOR COPIES.
                                              0 = NO MARKING OF INTERRUPT MESSAGE PAGE
                                              1 = MARK INTERRUPT MESSAGE PAGE
                                              MESSAGE PAGE
```

Figure 69 (Part 2 of 5). XTP7 Parameter Area Format

```
                 1... ....          XTP7IPOS       OFFSET STACK INTERRUPT
                                                   MESSAGE PAGE.
                                                   (VALID FOR INITIALIZATION
                                                   'XTP7INIT' AND BDSC 'XTP7BDS')
                                                   0 = NO OFFSET STACKING
                                                   1 = OFFSET STACKING
                 .1.. ....          XTP7MTOF       MAP FONTS TO OUTLINE FONTS
                                                   (VALID ONLY ON INITIALIZATION
                                                   CALL 'XTP7INIT')
                                                   0 = DO NOT MAP FONTS TO OUTLINE FONTS
                                                   1 = MAP FONTS TO OUTLINE FONTS
                 ..1.  ....         XTP7RFNF       RECOVER FROM FONT-NOT-FOUND
                                                   CONDITIONS (VALID ONLY ON
                                                   INITIALIZATION CALL 'XTP7INIT').
                                                   0 = DO NOT QUERY HOST FONT LIBRARIES
                                                       TO ENSURE THAT THE MAPPED FONT
                                                       ACTUALLY EXISTS.
                                                   1 = QUERY THE HOST FONT LIBRARIES FOR
                                                       THE MAPPED FONT.  IF THE MAPPED
                                                       DOES NOT EXIST, GO BACK AND USE
                                                       THE ORIGINAL FONT.
                                                       NOTE:  DUE TO THE ADDITIONAL
                                                       LIBRARY QUERIES FOR EVERY FONT
                                                       RESOURCE, PERFORMANCE WILL BE
                                                       DEGRADED IF THIS OPTION IS
                                                       TURNED ON.
                 ...1  ....         XTP7IPSU       SUPPRESS INTERRUPT MESSAGE PAGES
                                                   (VALID FOR INITIALIZATION
                                                   'XTP7INIT')
                                                   0 = ALLOW INTERRUPT MSG PAGES
                                                   1 = SUPPRESS INTERRUPT MSG PAGES
                 ....  1...         XTP738MO       SET THE MEDIA ORIGIN TO THE
                                                   3800 MEDIA ORIGIN FOR THE USER'S
                                                   DATA SET
                                                   (VALID FOR INITIALIZATION,
                                                   'XTP7INIT')
                                                   0 = DO NOT SET MEDIA ORIGIN TO
                                                       3800 MEDIA ORIGIN
                                                   1 = SET MEDIA ORIGIN TO
                                                       3800 MEDIA ORIGIN
                 ....  .1..         XTP7MDSD       SEND THE MESSAGE DATA SET TO A
                                                   SYSOUT DATA SET (VALID FOR
                                                   INITIALIZATION 'XTP7INIT'
                                                   AND BDSC 'XTP7BDS')
                                                   0 = DO NOT SEND MSG DATA SET TO A
                                                       SYSOUT DATA SET
                                                   1 = SEND MSG DATA SET TO A SYSOUT
                                                       DATA SET
                 ....  ..1.         XTP7HCA        COMPRESS IPDS DATA USING THE
                                                   HUFFMAN COMPRESSION ALGORITHM
                                                   (VALID FOR INITIALIZATION,
                                                   'XTP7INIT')
                                                   0 = NO COMPRESSION
                                                   1 = COMPRESS IPDS DAT
                 ....  ...1         XTP7C2SI       CONVERT A MULTIPLE-CELLED IM1
                                                   IMAGE TO A SINGLE IOCA IMAGE
                                                   (VALID FOR INITIALIZATION
                                                   'XTP7INIT' AND BDSC 'XTP7BDS')
                                                   0 = CONVERT TO MULTIPLE
                                                   1 = CONVERT TO SINGLE
                                                       IOCA IMAGE
                                                       IOCA IMAGES
```

Figure 69 (Part 3 of 5). XTP7 Parameter Area Format

```
     22   (16) CHARACTER    14  XTP7RRLV       REASONABLE RESOURCE LOADED
                                               VALUE-DETERMINES THE NUMBER OF
                                               RESOURCES RETAINED AT DATA SET END
     22   (16) SIGNED        2  XTP7RPD        RRLV FOR PAGEDEFS
     24   (18) SIGNED        2  XTP7RFD        RRLV FOR FORMDEFS
     26   (1A) SIGNED        2  XTP7RMO        RRLV FOR OVERLAYS
     28   (1C) SIGNED        2  XTP7RPS        RRLV FOR PAGE SEGMENTS
     30   (1E) SIGNED        2  XTP7RCF        RRLV FOR FONTS
     32   (20) SIGNED        2  XTP7ROC        RRLV FOR OBJECT CONTAINERS
     34   (22) CHARACTER     2  *              RESERVED
     36   (24) CHARACTER     8  XTP7RLST       APSURLST INFORMATION
     36   (24) ADDRESS       4  XTP7LSTP       POINTER TO APSURLST
     40   (28) SIGNED        2  XTP7RESC       RESOURCE COUNT-# OF DISTINCT
                                               RESOURCES IN RLST CHAIN
     42   (2A) SIGNED        2  XTP7LSTC       LIST COUNT-# OF ENTRIES IN
                                               RLST CHAIN
     44   (2C) CHARACTER     8  XTP7OVLY       NAME OF OVERLAY CURRENTLY
                                               BEING PROCESSED

     52   (34) CHARACTER    20  XTP7PINF       PRINTER INFORMATION
     52   (34) CHARACTER     8  XTP7PRT        PRINTER TYPE AND MODEL
     52   (34) CHARACTER     1  XTP7ATCH       ATTACHMENT TYPE
     53   (35) CHARACTER     4  XTP7FMLY       DEVICE FAMILY
     57   (39) CHARACTER     2  XTP7PRTT       DEVICE TYPE
     59   (3B) CHARACTER     1  XTP7PRTM       MODEL NUMBER
     60   (3C) BITSTRING     1  XTP7PFLG       PRINTER FLAGS
            1... ....           XTP7PSEC       SECURE PRINTER
            .111 1111           *              RESERVED
     61   (3D) CHARACTER     3  *              RESERVED
     64   (40) CHARACTER     8  XTP7PNAM       PRINTER NAME
     72   (48) CHARACTER     6  *              MESSAGE INSERT INFORMATION
                                               POINTER & LENGTH OF TEXT FOR
                                               OPTIONAL MESSAGE INSERT FOR APS954I
     72   (48) ADDRESS       4  XTP7MSGP       POINTER TO USER SUPPLIED, 64
                                               BYTE MESSAGE INSERT
     76   (4C) UNSIGNED      1  XTP7MSGL       LENGTH OF MESSAGE INSERT
                                               (0-64)
     77   (4D) CHARACTER     1  *              RESERVED
     78   (4E) BITSTRING     1  XTP7CRLF       CRL FLAGS
                                               0 = FREE THE STORAGE
                                               1 = KEEP THE CRL
            1...  ....          XTP7CFPD       FREE/KEEP PAGEDEF CRL
            .1..  ....          XTP7CFFD       FREE/KEEP FORMDEF CRL
            ..1.  ....          XTP7CFCF       FREE/KEEP FONT CRL
            ...1  ....          XTP7CFMO       FREE/KEEP OVERLAY CRL
            ....  1...          XTP7CFPS       FREE/KEEP SEGMENT CRL
            ....  .1..          XTP7CSSD       FREE/KEEP SECUR. DEFN CRL
            ....  ..1.          XTP7CSCF       FREE/KEEP SECUR. FONT CRL
            ....  ...1          XTP7CSMO       FREE/KEEP SECUR. OVLY CRL
     79   (4F) BITSTRING     1  XTP7CRF2       CRL FLAGS
                                               0 = FREE THE STORAGE
                                               1 = KEEP THE CRL
            1... ....           XTP7CSPS       FREE/KEEP SECURITY SEG CRL
            .1.. ....           XTP7CFOC       FREE/KEEP OBJECT CONTAINER CRL
            ..11  1111          *              RESERVED
     80   (50) CHARACTER    40  XTP7CRLP       POINTERS TO THE CONCATENATION
                                               RETRIEVAL LISTS
     80   (50) ADDRESS       4  XTP7CLPD       POINTER TO PAGEDEF CRL
     84   (54) ADDRESS       4  XTP7CLFD       POINTER TO FORMDEF CRL
     88   (58) ADDRESS       4  XTP7CLCF       POINTER TO FONT CRL
     92   (5C) ADDRESS       4  XTP7CLMO       POINTER TO OVERLAY CRL
     96   (60) ADDRESS       4  XTP7CLPS       POINTER TO SEGMENT CRL
    100   (64) ADDRESS       4  XTP7CLOC       POINTER TO OBJECT CONTAINER CRL
    104   (68) ADDRESS       4  XTP7CLSD       PTR TO SECURITY DEFN CRL
    108   (6C) ADDRESS       4  XTP7CLSF       PTR TO SECURITY FONT CRL
    112   (70) ADDRESS       4  XTP7CLSO       PTR TO SECURITY OVLY CRL
    116   (74) ADDRESS       4  XTP7CLSP       PTR TO SECURITY SGMT CRL
```

Figure 69 (Part 4 of 5). XTP7 Parameter Area Format

```
      120   (78) BITSTRING     1  XTP7MFLG       MORE RESPONSE FLAGS
             1... ....            XTP7JHMO       SET THE JOB HEADER MEDIA
                                                 ORIGIN TO TOP LEFT CORNER
                                                 (VALID FOR INITIALIZATION
                                                 'XTP7INIT' AND BDSC
                                                 'XTP7BDS')
                                                 0 = DO NOT SET JOB HEADER
                                                     MEDIA ORIGIN
                                                 1 = SET JOB HEADER MEDIA
                                                     ORIGIN TO TOP LEFT
                                                     CORNER
             .1.. ....            XTP7JTMO       SET THE JOB TRAILER MEDIA
                                                 ORIGIN TO TOP LEFT CORNER
                                                 (VALID FOR INITIALIZATION
                                                 'XTP7INIT' AND BDSC
                                                 'XTP7BDS')
                                                     CORNER
                                                 0 = DO NOT SET JOB
                                                     TRAILER MEDIA ORIGIN
                                                 1 = SET JOB TRAILER MEDIA
                                                     ORIGIN TO TOP LEFT
             ..1. ....            XTP7DHMO       SET THE DATA SET HEADER
                                                 MEDIA ORIGIN TO TOP LEFT
                                                 CORNER
                                                 (VALID FOR INITIALIATION
                                                 'XTP7INIT' AND BDSC
                                                 'XTP7BDS')
                                                 0 = DO NOT SET DATASET
                                                     HEADER MEDIA ORIGIN
                                                 1 = SET DATASET HEADER
                                                     MEDIA ORIGIN TO TOP
                                                     LEFT CORNER
             ...1 ....            XTP7MDMO       SET THE MESSAGE DATASET
                                                 MEDIA ORIGIN TO TOP LEFT
                                                 CORNER
                                                 (VALID FOR INITIALIZATION
                                                 'XTP7INIT' AND BDSC
                                                 'XTP7BDS')
                                                 0 = DO NOT SET MESSAGE
                                                     DATASET MEDIA ORIGIN
                                                 1 = SET MESSAGE DATASET
                                                     ORIGIN TO TOP LEFT
                                                     CORNER
             .... 1...            XTP7RSTR       ABEND RESTART FLAG
                                                 (VALID ONLY ON TERMINATION
                                                 CALL 'XTP7TERM')
                                                 0 = NO ABEND OR ABEND IS
                                                     NOT RESTARTABLE
                                                 1 = RESTARTABLE ABEND
                                                     OCCURRED
             .... .111            *              RESERVED
      121   (79) CHARACTER     2  XTP7ABRC       ABEND REASON CODE
                                                 (VALID ONLY ON TERMINATION
                                                 CALL 'XTP7TERM')
      123   (7B) CHARACTER    21  *              RESERVED
```

Figure 69 (Part 5 of 5). XTP7 Parameter Area Format

XTP7 Fields

XTP7CID Contains the control block identifier XTP7.

XTP7LENG Contains the length of the XTP7 DSECT.

XTP7ETYP Contains information flags set by PSF to indicate which type of exit call is in progress. Only one of these flags is set to '1' for each call to the exit.

XTP7RTYP Information flags set by PSF to indicate the type of resource being processed. For calls to the exit that involve resources (RAC, RLC, and RDC), only one XTP7RTYP flag is set to '1'. For calls to the exit that do not involve resources (INIT, BDSC, and TERM), all XTP7RTYP flags are set to '0'.

XTP7DSTY Information flags set by PSF to indicate which type of data set is being processed. Only one of these flags is set to '1' except during the INIT and TERM calls, when all of these flags are set to '0'.

- **XTP7AUX** PSF is processing auxiliary resources. This includes all resources used in auxiliary data sets and the default page definition and form definition. Auxiliary data sets are the job header and trailer, data set header, and message data set.
- **XTP7PDS** PSF is processing a user's print data set.

XTP7DSAT Information flags, set by PSF, that indicate the specific type of data set being processed. Only one of these flags is set to '1' except during the INIT and TERM calls, when all of these flags are set to '0'. These flags give more detail than XTP7DSTY about the data set being processed.

- **XTP7PDFT** PSF is processing the default page definition or form definition.
- **XTP7PJHD** PSF is processing the job header.
- **XTP7PJTR** PSF is processing the job trailer.
- **XTP7PDSH** PSF is processing the data set header.
- **XTP7PMDS** PSF is processing the message data set.
- **XTP7PLDS** PSF is processing the last print data set in the job; a job trailer will be printed.

XTP7ACSE Valid only on the INIT and BDSC calls to APSUX07 when CSE is being used. PSF initially sets this field to X'00'. This value causes PSF not to check to see if a page fits 2 up on the sheet. If this field is set to X'01', PSF checks the first page printed, using a new copy group, to see if it fits 2 up on the sheet. If this field is set to X'02', PSF will check all the font side pages to see if they fit 2 up on the sheet. If XTP7ACSE is set to any other value, it defaults to X'00'.

XTP7IPCC Valid only on the INIT and BDSC calls to APSUX07. PSF initially sets this field to X'00', which causes 1 copy of the interrupt message page to be printed. If flag XTP7IPMF is set, and APSUX07 sets this field to a binary value from 1 to 10 decimal inclusive, PSF produces many copies of the interrupt message page. If XTP7IPCC is set to a value outside the valid range, it defaults to 1. This field is ignored if the XTP7IPMF flag is not set or if the device type is 3800. For details about copies when forms are marked, see “3800 Forms Marking” on page 201.

XTP7NOTY Flags that indicate when the exit wants control. The default state of all XTP7NOTY flags is '0' (inactive). If an XTP7NOTY flag is set to '1', PSF calls the exit when it gets to that particular point in the processing of a data set. An XTP7NOTY flag set to '0' indicates that PSF should not call the exit.

- The state of the XTP7NOTY flags can be changed only by the exit.
- PSF supplies the current settings of the XTP7NOTY flags to the exit before calling the exit.
- All the flags in XTP7NOTY can be set independent of one another.

XTP7MISC

XTP7EBDS A flag that tells PSF the exit is to be called for the beginning of the data set call (BDSC) function.

XTP7ETRM A flag that tells PSF the exit is to be called for FSA termination (TERM) function. This is for both normal and abnormal termination.

XTP7NACC Flags telling PSF that the exit is to be called for the resource access call (RAC) function. Each resource type has an XTP7NACC flag.

XTP7NLD Flags telling PSF that the exit is to be called for the resource load call (RLC) function. Each resource type has an XTP7NLD flag.

XTP7NDEL Flags telling PSF that the exit is to be called for resource delete call (RDC) function. Each resource type has an XTP7NDEL flag.

XTP7RFLG Flags that the exit can modify to request certain functions.

XTP7ETFF PSF sets this flag to B'0' to indicate that an eject to front facing is to be done before the beginning of the next document. APSUX07 can set this field to B'1' to bypass the ETFF function. This flag is valid only for continuous-forms printers, and can be accessed only through INIT and BDSC calls to APSUX07.

XTP7TDS PSF sets this flag to B'0'. If the exit sets this flag to B'1', PSF terminates the current data set. This flag is valid for the BDSC, RAC, and RLC exit calls to APSUX07.

XTP7EFFH PSF sets this flag to B'0' to indicate that an eject to front facing is to be done before the job header page. APSUX07 can set this field to B'1' to bypass the ETFF function. This flag is valid only for continuous-forms printers; it can be accessed only through INIT and BDSC calls to APSUX07.

XTP7HDF PSF sets this flag to B'0' to indicate that the printer should use the hardware default font. If APSUX07 sets this field to B'1', PSF lets the printer substitute the first font in the current font list and replace the hardware default font. This flag is valid only with 3800 printers and during calls to APSUX07.

XTP7HQUE Valid only when APSUX07 has requested the end of processing for the current data set. (XTP7TDS=B'1'). PSF initially sets this field to B'0'. When processing ends, the data set is purged. APSUX07 can set this field to B'1', to request that the data set be sent to JES and marked as unprintable rather than purged.

XTP7PRNS Valid only on the INIT call to APSUX07. PSF initially sets this field to B'0', to indicate that non-3800 single-byte fonts are to be pruned. For more information about font pruning, see “Number and Type of Resources” on page 222 and “Glossary” on page 353. If APSUX07 sets this field to B'1', PSF does not prune single-byte fonts.

XTP7PRND Valid only on the INIT call to APSUX07. PSF initially sets this field to B'0', to indicate that non-3800 double-byte fonts are to be pruned. For more information about font pruning, see “Number and Type of Resources” on page 222 and “Glossary” on page 353. If APSUX07 sets this field to B'1', PSF is not to prune double-byte fonts.

XTP7IPMF Valid only on the INIT and BDSC calls to APSUX07. PSF initially sets this field to B'0', to indicate that no form marks are required for the interrupt message page. If APSUX07 sets this field to B'1', PSF marks the interrupt message page with form marks.

XTP7IPOS Valid only on the INIT and BDSC calls to APSUX07. PSF initially sets this field to B'0', to indicate that no offset stacking is required for the interrupt message page. If APSUX07 sets this field to B'1', the printed output is offset-stacked, starting at the interrupt message page.

XTP7MTOF Valid only on the INIT call to APSUX07. PSF initially sets this field to B'0', to indicate that fonts are not mapped to outline fonts. If APSUX07 sets this flag to B'1', PSF maps fonts to an equivalent outline font if the printer supports outline fonts. PSF does not map fonts used on auxiliary data sets, fonts found inline or in a user library, or fonts substituted by Exit 7. Double-byte raster coded fonts are prefix-mapped to equivalent outline coded fonts. However, a double-byte raster font section requested by character set and code page name is not mapped to outline, because there is no outline equivalent for just one section.

XTP7RFNF Valid only on the INIT call to APSUX07. The Recover-from-Font-Not-Found flag applies only when XTP7MTOF is set ON to B'1'. PSF initially sets this field to B'0', to indicate that PSF is not to query the host libraries to make sure that the mapped font exists before attempting to load it into the printer. If the mapped font does not exist inline or in the host libraries, PSF issues an error message and terminates the page. If APSUX07 sets this flag to B'1', then, when PSF has mapped a raster font to an outline font, it first queries to see if the font exists inline or in the host libraries. If the mapped font does not exist there, PSF uses the original font. Because there are additional library queries for every mapped font resource, PSF performance is degraded.

XTP7IPSU Valid only on the INIT call to APSUX07. PSF initially sets this field to B'0', to indicate that interrupt message pages are desired. For more information, see "Considerations on Suppressing an Interrupt Message Page" on page 197. If APSUX07 sets this flag to B'1', interrupt message pages are suppressed (are not printed). An interrupt message page is printed, for example, at the point at which a data set is backspaced. Intervention conditions, such as a paper jam, can also cause an interrupt message page.

XTP738MO Valid only on the INIT and BDSC calls. PSF initially sets this field to B'0', to indicate that the user's data set media origin is not set to the 3800 media origin. If APSUX07 sets this flag to B'1', PSF sets the user's data set media origin to the 3800 media origin.

XTP7MDSD Valid only on the INIT and BDSC calls. PSF initially sets this field to the PIMSG keyword setting on the OUTPUT statement for the message data set. If APSUX07 sets this flag to B'0', PSF will not send the message data set to a SYSOUT data set. If APSUX07 sets this flag to B'1', PSF will send the message data set to a SYSOUT data set.

XTP7HCA Valid only on the INIT call. PSF initially sets this field to B'0', to indicate that the IPDS data is not to be compressed. If APSUX07 sets this field to B'1' and the printer is SDLC-attached, PSF compresses the IPDS data by use of the Huffman compression algorithm.

XTP7C2SI Valid only on the INIT and BDSC calls. PSF initially sets this field to B'0', to indicate that a multiple-celled IM1 image is to be converted to multiple IOCA images. If APSUX07 sets this flag to B'1', PSF converts a multiple-celled IM1 image to a single IOCA image.

XTP7RRLV PSF provides the current reasonable resource loaded values (RRLVs) in the following fields. PSF uses the RRLVs to determine how many resources should remain loaded at the end of a user's print data set. The exit can write to these fields to change any RRLVs. For the default values of RRLV, see Table 20 on page 307.

XTP7RPD RRLV for page definitions.

XTP7RFD RRLV for form definitions.

XTP7RMO RRLV for medium overlays. PSF always sets this value to '0'. If APSUX07 sets this field to a non-zero value, PSF ignores that setting and sets the value to '0'.

XTP7RPS RRLV for page segments.

XTP7RCF RRLV for fonts. A double-byte font is considered one font.

XTP7ROC RRLV for object containers. PSF always sets this value to '0'. If APSUX07 sets this field to a non-zero value, PSF ignores it and sets the value to '0'.

XTP7RLST PSF supplies information about the resource list (APSURLST) control block in the following fields:

XTP7LSTP Pointer to the first RLST control block in the RLST chain. This field is set to '0' if no RLST chain exists—that is, if this call to the exit does not involve resources.

XTP7RESC Contains the number of distinct resources that are being described in the RLST chain.

XTP7LSTC Contains the total number of RLST entries in the RLST chain. This value differs from XTP7RESC only during the RAC exit, when the RLST chain is describing a font that contains multiple entries in the Map Coded Font (MCF) structured field. In all other cases, this value is the same as XTP7RESC.

XTP7OVLY PSF supplies the name of the overlay being processed by PSF. If PSF is not involved in processing an overlay or any resource nested in the overlay, this field is set to null (X'FFFF000000000000')

XTP7PINF PSF provides the following information on the attached printer:

XTP7PRT Printer type and model

XTP7ATCH Attachment type—C=Channel, S=SNA, T=TCP/IP.

XTP7FMLY Device family—3800, 3820, or, for all other IBM printers, AFP1

XTP7PRTT Device type—not intended for use in user programming.

XTP7PRTM Model number for the IBM 3800—for example, 3800 Model 3.

Note: The IBM 3800 Model 6 is listed as a Model 3.

XTP7PFLG Printer flags

XTP7PSEC PSF sets this flag to B'1',to indicate that the printer is a secure printer—that is, that it supports the Define User Area IPDS command.

XTP7PNAM Printer name, as designated by the installation in JES initialization statements.

PSF message APS954I allows the exit up to 64 bytes of text at the end of the message. APS954I indicates that the resource exit has terminated the data set (see XTP7TDS on page 286). The optional text can be used to explain why the data set has been terminated. To get this additional text, the exit must put the applicable information into the following two fields:

XTP7MSGP Pointer to the text. If this field is '0', no text is added to the message.

XTP7MSGL Length of the text. The domain of this field is '0' to '64', the maximum number of bytes PSF adds to APS954I. If XTP7MSGL is '0', no text is added to the message.

XTP7CRLF, XTP7CRF2
Flags that tell PSF to keep or delete the ten concatenation retrieval lists (CRLs). Before the INIT call to the exit, PSF sets all CRL flags to '0'. If the exit sets a CRL flag to '1', PSF retains the CRL for the entire FSA session. Only after the INIT call to the exit does PSF delete a CRL and free its storage.

XTP7CRLP PSF provides pointers to the ten concatenation retrieval lists (see “APSLCRL Parameter Area (Concatenation Retrieval List)” on page 291) at every call to the exit. If the storage for a CRL has been freed (see XTP7CRLF and XTP7CRF2 on page 290), or if no library has been found for a resource type, PSF sets the pointer to '0'.

XTP7MFLG Flags that the exit can modify to request certain functions.

XTP7JHMO Valid only on the INIT and BDSC calls. PSF initially sets this field to B'0', to indicate that the job header media origin is not set to the top left corner. If APSUX07 sets this flag to B'1', PSF sets the job header media origin to the top left corner.

XTP7JTMO Valid only on the INIT and BDSC calls. PSF initially sets this field to B'0' to indicate that the job trailer media origin is not set to the top left corner. If APSUX07 sets this flag to B'1', PSF sets the job trailer media origin to the top left corner.

XTP7DHMO Valid only on the INIT and BDSC calls. PSF initially sets this field to B'0', to indicate that the data set header media origin is not set to the top left corner. If APSUX07 sets this flag to B'1', PSF sets the data set header media origin to the top left corner.

XTP7MDMO	Valid only on the INIT and BDSC calls. PSF initially sets this field to B'0', to indicate that the message data set media origin is not set to the top left corner. If APSUX07 sets this flag to B'1', PSF sets the message data set media origin to the top left corner.
XTP7RSTR	Valid only on the TERM call. PSF initially sets this field to B'1' if the abend is restartable or to 0 if no abend occurred or if the abend is not restartable. If this field is set to B'1' (restartable abend), APSUX07 can set this field to 0 to cause PSF to not restart.

XTP7ABRC Valid only on the TERM call. This field contains the abend reason code if the FSA is terminating abnormally. It contains X'0000' if the FSA is terminating normally.

APSLCRL Parameter Area (Concatenation Retrieval List)

The concatenation retrieval list (CRL) is a control block that PSF builds to pass information on resource libraries to the exit. Before the INIT call to the exit, PSF builds a CRL for each resource library specified in PSF initialization statements. PSF supplies pointers to the CRLs and to flags that tell PSF whether CRL information is to be saved. The pointers and flags are in the XTP7CRLP, XTP7CRLF, and XTP7CRF2 sections of APSGEXTP (see page 290).

If a CRL does not exist (either because no library was specified or because PSF has freed the storage for it), the CRL pointer is set to '0'. The CRL is not intended to be written to by the exit; however, PSF does not reference the CRL other than to supply the pointers to the exit.

Figure 70 on page 292 shows the format of APSLCRL parameter area, and Figure 71 on page 293 shows the format of the APSLCRL data area.

Note: The CRLs are for the system libraries only. No CRL is passed for a user library.

```
 Offsets  Type     Length  Name          Description
  0    (0) STRUCTURE    40  APSLCRL       CONCATENATION RETRIEVAL LIST
  0    (0) CHARACTER    20  CRLPFIX       PREFIX SECTION
  0    (0) CHARACTER     4  CRLID         CONTROL BLOCK ID
  4    (4) SIGNED        4  CRLLEN        LENGTH INCLUDING DATA SET
                                          ENTRIES
  8    (8) CHARACTER     6  CRLPTF        PTF DATE
 14    (E) CHARACTER     2  CRLREL        RELEASE LEVEL
 16   (10) SIGNED        1  CRLEID        OBJECT ENCODED ID FROM APSGSFX
                                          IDENTIFIES TYPE OF OBJECTS IN
                                          THIS LIBRARY.
 17   (11) BITSTRING     1  CRLSPN        SUBPOOL NUMBER
 18   (12) BITSTRING     1  CRLFLG0       FLAGS
       1... ....            CRLSECUR      SECURITY LIBRARY
       .111  1111                         RESERVED
 19   (13) CHARACTER     1                RESERVED
 20   (14) CHARACTER    20  CRLBASE       BASE SECTION
 20   (14) CHARACTER     8  CRLDDNM       DDNAME FOR DATA SET ENTRIES
 28   (1C) SIGNED        2  CRLRTRVD      NUMBER OF DATA SETS FOR WHICH
                                          JFCBS WERE RETRIEVED. FROM
                                          FIELD ARLRETRV IN IHAARL OR
                                          GENERATED BY APSLOPEN.
                                          DOMAIN = (0  256)
                                          0 = NO DATA SET NAMES RETRIEVED
 30   (1E) SIGNED        2  CRLCONC       NUMBER OF DATA SETS CONCATENATD
                                          TOGETHER. FROM FIELD ARLCONC
                                          IN IHAARL OR GENERATED BY
                                          APSLOPEN. DOMAIN = (1  256)
                                          1 = NOT A CDS
                                          2 TO 256 = CDS
 32   (20) SIGNED        2  CRLLOWCN      LOW CONCATENATION NUMBER FOR
                                          THIS DATA SET MAPPING.
                                          DOMAIN = (0  255)
 34   (22) SIGNED        2  CRLHIGCN      HIGH CONCATENATION NUMBER FOR
                                          THIS DATA SET MAPPING.
                                          DOMAIN = (0  255)
 36   (24) CHARACTER     4  *             RESERVED
 40   (28) CHARACTER        CRLEND
```

Figure 70. APSLCRL Parameter Area Format

```
Offsets  Type     Length  Name           Description

0    (0) STRUCTURE        CRLDATA        DATA SET NAME AND VOLUME
                                         ENTRY. NUMBER OF ENTIRES IS
                                         EQUAL VALUE IN FIELD CRLRTRVD.
0    (0) SIGNED        2  CRLDSELN       LENGTH OF DATA SET AND VOLUME
                                         SERIAL LIST PLUS LENGTH BYTES.
                                         CRLDSELN IS USED TO INCREMENT
                                         TO NEXT ENTRY. MAPPED BY
                                         CRLDATA.
                                         DOMAIN = (0 32767)
2    (2) SIGNED        2  CRLCATN        CONCATENATION NUMBER FOR
                                         DATA SET NAME IN CRLDSNM.
                                         GENERATED BY APSLJFCB OR
                                         APSLOPEN. DOMAIN = (0 255)
                                         0 = FIRST OR ONLY DATA SET
                                         1 = 2ND DATA SET
4    (4) SIGNED        2  CRLNVOL        NUMBER OF VOLUMES FROM
                                         JFCBNVOL. DOMAIN = (1 256)
6    (6) CHARACTER    44  CRLDSNM        JFCB INDEX BY CRLRTRVD.
                                         X'40' IN FIRST BYTE OF ENTRY
                                         INDICATES DATA SET NAME COULD
                                         NOT BE RETRIEVED.
                                         DOMAIN = (ALPHANUMERIC)
50  (32) CHARACTER     0  CRLVOLS        VOLUME LIST
50  (32) CHARACTER     6  CRLAVOLS       VOLUME SERIAL NUMBERS. INDEX
                                         BY CRLNVOL. X'FF' IN FIRST
                                         BYTE OF ENTRY INDICATES VOLUME
                                         SERIAL NOT DETERMINE. ONE
                                         VOLUME FOR PARTITION DATA SET
                                         DOMAIN: 1-6 ALPHANUMERIC
                                         CHARACTERS
```

Figure 71. Format of the CRL Data Area

APSLCRL Fields

CRLPFIX CRL prefix section.

- **CRLID** Contains the control block identifier CRL.
- **CRLLEN** Contains the length of the CRL, including the data set entries.
- **CRLPTF** Contains the date of the PTF.
- **CRLREL** Contains the release level.
- **CRLEID** Contains an encoded object ID.
- **CRLSPN** Contains the subpool number.

CRLBASE CRL base section.

- **CRLDDNM** Contains the DDNAME for the data set entries.
- **CRLRTRVD** Contains the number of data set entries; if no data set names are retrieved, it is set to '0'.
- **CRLCONC** Contains the number of concatenated data sets; if this is not a concatenated data set, CRLCONC is set to '1'.

CRLLOWCN Contains the lowest concatenation number for this data set mapping.

CRLHIGCN Contains the highest concatenation number for this data set mapping.

CRLEND End of the CRL base section; can be used as addressability for CRLDATA.

CRL Data Area

CRLDATA This area contains the data set names and volume entries. The number of entries is equal to the value in CRLRTRVD.

CRLSDELN Contains the length of the data set and volume serial list. This field is used to increment to the next entry mapped by CRLDATA.

CRLCATN Contains the concatenation number for the data set name in CRLDSNM. This number is supplied to the exit in the RLSTPDS, RLSTCPDS, or RLSTFNDS fields during the resource load calls.

CRLNVOL Contains the number of volumes in CRLVOLS; can be used as a maximum index for CRLAVOLS.

CRLDSNM Contains the name of the partitioned data set. A X'40' in the first byte indicates that the data set could not be retrieved.

CRLVOLS Volume list.

CRLAVOLS An array that contains the volume serial numbers. CRLNVOL contains the number of elements to this array. A X'FF' in the first byte indicates that PSF is unable to determine the volume serial number.

APSURLST Parameter Area (Resource List)

APSURLST is a control block that PSF builds when it needs to provide information to the exit about a specific resource. Usually, PSF builds one RLST entry for each resource associated with the current call to the exit. If multiple RLST entries are required to describe all the resources to the exit, the RLST entries are linked to form a single-headed, double-threaded chain. The first, or only, RLST entry is pointed to by XTP7LSTP in the XTP7 control block area.

Note: APSUX07 must never modify the pointers RLSTPRVS, RLSTNEXT, and RLSTEXTP as defined in APSURLST.

All the fields in APSURLST are read by the exit; however, some fields contain information that is valid only for certain exit calls and certain resource types. Also, most fields in APSURLST are not intended to be written to by the exit. PSF looks for changes to specific fields after certain calls to the exit. For more details, see “APSURLST Fields” on page 296.

Figure 72 on page 295 shows the format of APSURLST.

```
 Offsets  Type     Length  Name          Description
 0    (0) CHARACTER     *  APSURLST
 0    (0) CHARACTER     4  RLSTCID        CONTROL BLOCK ID = RLST
 4    (4) SIGNED        4  RLSTLENG       LENGTH OF RLST CONTROL BLOCK
 8    (8) CHARACTER    16  RLSTPTRS       RESOURCE LIST POINTERS
 8    (8) ADDRESS       4  RLSTPRVS       POINTER TO PREVIOUS ENTRY
12    (C) ADDRESS       4  RLSTNEXT       POINTER TO NEXT ENTRY
16   (10) ADDRESS       4  RLSTEXTP       POINTER TO FONT EXTENSION AREA
20   (14) ADDRESS       4  RLSTMSGP       POINTER TO 64 BYTE MESSAGE
                                            INSERT TEXT
24   (18) CHARACTER    16  RLSTRID        RESOURCE IDENTIFICATION
24   (18) CHARACTER     8  RLSTNAME       RESOURCE NAME
24   (18) BITSTRING     2  RLSTGCID       GRAPHIC CHARACTER SET
                                          GLOBAL IDENTIFIER GCSGID
26   (1A) BITSTRING     2  RLSTCPID       CODE PAGE GLOBAL IDENTIFIER
28   (1C) BITSTRING     2  RLSTFGID       FONT GLOBAL IDENTIFIER
30   (1E) BITSTRING     2  RLSTFW         FONT WIDTH
32   (20) UNSIGNED      1  RLSTPDS        PDS NUMBER FOR RESOURCE
                                          IDENTIFIED IN RLSTNAME FIELD
33   (21) UNSIGNED      1  *              RESERVED
34   (22) BITSTRING     2  RLSTATTR       RESOURCE ATTRIBUTES
           1... ....       RLSTDFT        DEFAULT PAGEDEF/FORMDEF
           .1.. ....       *              RESERVED
           ..1. ....       RLSTINL        INLINE RESOURCE
           ...1 ....       RLSTSPS        SOFT PAGE SEGMENT
           .... 1...       RLSTMEF        MULTIPLE ENTRY FONT
           .... .1..       RLSTSEC        SECURITY RESOURCE
           .... ..1.       RLSTUSER       USERLIB RESOURCE
           .... ...1       *              RESERVED
           1111  1111      *              RESERVED
36   (24) BITSTRING     2  RLSTSTAT       RESOURCE STATUS FLAGS
           1... ....       RLSTAUX        AUXILIARY RESOURCE
           .1.. ....       RLSTDEP        HAS DEPENDENCIES
           ..1. ....       RLSTNEST       NESTED RESOURCE
           ...1 ....       RLSTPLR        PRIORITY LEVEL RES.
           ....  1111      *              RESERVED
           1111  1111      *              RESERVED
38   (26) CHARACTER     2  *              RESERVED
40   (28) SIGNED        2  RLSTRESC       RESOURCE COUNT-THE 5TH
                                          RESOURCE IN THE RLST CHAIN
                                          WILL BE #5
42   (2A) UNSIGNED      1  RLSTMSGL       LENGTH OF USER'S MESSAGE
                                          INSERT POINTED AT BY RLSTMSGP
43   (2B) CHARACTER     1  *              RESERVED
```

Figure 72 (Part 1 of 2). APSURLST Parameter Area Format

```
        44   (2C) CHARACTER     4  RLSTFLGS       FLAGS
        44   (2C) BITSTRING     1  RLSTMISC       MISCELLANEOUS FLAGS
              1... ....            RLSTOFNT       PRINTER SUPPORTS
                                                  OUTLINE FONTS
              .1.. ....            RLSTGRID       RESOURCE NAME IS A GRID
              ..11  1111           *              RESERVED
        45   (2D) BITSTRING     1  RLSTAFLG       FLAGS USED ONLY AT ACCESS TIME
              1... ....            RLSTLOAD       FORCE LOAD (RELOAD)
              .1.. ....            RLSTSMSG       ISSUE SUB MESSAGE
              ..1. ....            RLSTRRI        RELOAD REQ. IGNORED (SET BY
                                                  PSF)
              ...1  1111           *              RESERVED
        46   (2E) BITSTRING     1  RLSTLFLG       FLAGS USED ONLY AT LOAD TIME
              1... ....            RLSTLCF        LOADING A CODED FONT
              .1.. ....            RLSTLCP        LOADING A CODE PAGE
              ..1. ....            RLSTLFN        LOADING A FONT CHARACTER SET
              ...1  1111           *              RESERVED
        47   (2F) BITSTRING     1  RLSTDFLG       FLAGS USED ONLY AT DELETE TIME
              1... ....            RLSTKEEP       KEEP THIS RESOURCE
              .111  1111           *              RESERVED
        48   (30) UNSIGNED      2  RLSTVFS        VERTICAL FONT SIZE
        50   (32) UNSIGNED      2  RLSTHFS        HORIZONTAL FONT SIZE
        52   (34) UNSIGNED      2  RLSTHSCF       HORIZONTAL SCALE FACTOR
                                                  SPECIFIED IN 1440THS OF AN
                                                  INCH
        54   (36) CHARACTER    18  *              RESERVED
```

Figure 72 (Part 2 of 2). APSURLST Parameter Area Format

APSURLST Fields

RLSTCID Contains the control block identifier RLST.

RLSTLENG Contains the length of the RLST control block.

RLSTPTRS Are pointers:

- **RLSTPRVS** The pointer to the previous RLST entry in the chain. If this is the first RLST entry, this variable is set to '0'. The exit must never modify this field.

- **RLSTNEXT** The pointer to the next RLST entry in the chain. If this is the last RLST entry, this variable is set to '0'. The exit must never modify this field.

- **RLSTEXTP** The pointer to the font extension area, RLSTFEXT. If a font is described as a code page and character set pair (sometimes abbreviated as CP/FN pair), PSF builds the font extension area. If a FEXT does not exist, this field is '0'. The exit must never modify this field.

- **RLSTMSGP** The pointer to the message insert text. The exit may add up to 64 bytes of text to the end of PSF message APS563I. This field is used with FLSTMSGL and RLSTSMSG. For more information on the optional text for APS563I, see the descriptions of these fields. If the exit leaves this field at '0', no text is added to the message.

RLSTRID Contains information about the resource represented by the RLST entry. All fields contain information that PSF supplies to the exit.

- **RLSTNAME** The resource member name. For resources other than fonts, this name contains all eight characters of the member name—for example, P106060. For coded fonts, this 8-byte field contains all eight characters of the member name or the GRID. Flag RLSTGRID is set when the field is a GRID. The GRID contains 4 fields: RLSTGCID, RLSTCPID, RLSTFGID, and RLSTFW. Sometimes fonts are described not by this 8-character field, but by their code page and character set names. If the coded font name or GRID is not available, this field is set to null, and the code page and character set names are in the font extension area (see RLSTFEXT). The exit can modify this field during the RAC if it wishes to make a substitution of the resource. Substitutions must be name for name, GRID for GRID, and so on.

  - **RLSTGCID** The 2-byte graphic character set global identifier (GCSGID) in the GRID for this font.
  - **RLSTCPID** The 2-byte code page global identifier in the GRID for this font.
  - **RLSTFGID** The font global identifier in the GRID for this font.
  - **RLSTFW** The font width in the GRID for this font.

- **RLSTPDS** The partitioned data set concatenation number. PSF supplies this information during both the resource-load call and the resource-delete call. This field is not valid for inline resources.

  For more information about CRLCATN, see page 294.

  If the resource is from the system library, RLSTPDS is the PDS concatenation number in the system library. If the resource is from the user library, RLSTPDS is the PDS concatenation number in the user library.

- **RLSTATTR** Are attributes of this resource.

  - **RLSTDFT** If set, indicates this is a default page definition or form definition.

    Note: This flag is set only for the page definition or form definition specified in the PRINTDEV statement.

  - **RLSTINL** If set, indicates that this is an inline resource.
  - **RLSTSPS** If set, indicates that this is a soft page segment.

- **RLSTMEF** If set, indicates that this is a multiple-entry font.
- **RLSTSEC** If set, indicates that the resource is from a security library.
- **RLSTUSER** If set, indicates that the resource is from a user library.

RLSTSTAT Contains information that affects the deletion status of the resource.

- **RLSTAUX** If set, indicates that this is an auxiliary resource. PSF never deletes auxiliary resources during data set end processing. The exit cannot change the auxiliary status of a resource.
- **RLSTDEP** If set to '1', indicates that this resource has dependencies; that is, it has mapped (nested) another resource. Overlays are the only resource type that can nest another resource. When an overlay that has dependencies is loaded in the printer, a nested resource cannot be deleted. The exit cannot change the dependency status of a resource.
- **RLSTNEST** If set to '1', indicates that this resource has been mapped (nested) in another resource. When a resource that has dependencies is loaded in the printer, a nested resource cannot be deleted. Fonts and hard page segments are the only resources that can be nested. The exit cannot change the nested status of a resource.
- **RLSTPLR** Indicates that this resource is a priority-level resource. All resources default to a non-PLR (RLSTPLR='0'). By setting this field to '1', the exit can make the resource a PLR; this improves its chance of remaining in the printer at the end of the data set.

RLSTRESC Field set by PSF to indicate the position of this resource in the RLST chain. If a resource requires more than one RLST entry (multiple-entry fonts), all RLST entries for that resource have the same RLSTRESC value. The value of RLSTRESC is never more than XTP7RESC.

RLSTMSGL Indicates the length of the message insert text for APS563I. PSF always sets this field to '0'. If the exit supplies a pointer to some text in RLSTMSGP and sets this field to the length of the text (64 maximum), PSF adds that text to the end of APS563I.

RLSTFLGS Flags.

RLSTMISC General flags

RLSTOFONT Set by PSF to B'1' on entry to the exit when the attached printer supports outline fonts. For printers that do not support outline fonts, set to B'0'.

RLSTGRID Set by PSF to B'1' when the RLSTNAME field contains a GRID for a coded font. Otherwise, set to B'0'.

RLSTAFLG Flags valid only during the resource-access call (RAC).

RLSTLOAD Forces a load of this resource. PSF always sets this flag to '0'. The exit can set it to '1' to force PSF to load this resource. If the resource is already loaded, PSF deletes the resource and then reloads it.

Note: If the resource concerned is a page definition or a form definition, the load involved is from DASD to PSF virtual storage. For other resources, setting this flag causes the resource to be reloaded into printer storage.

RLSTSMSG Issues substitution message APS563I. PSF always sets this flag to '0'. If the exit has requested a substitution of this resource and it sets RLSTSMSG to '1', PSF issues message APS563I.

RLSTRRI Set by PSF to B'1' to indicate that PSF is to ignore a load request (RLSTLOAD) for this resource.

RLSTLFLG Flags valid only during the resource-load call (RLC).

RLSTLCF Informational flag set by PSF to indicate that PSF is loading a coded font.

RLSTLCP Informational flag set by PSF to indicate that PSF is loading a code page.

RLSTLFN Informational flag set by PSF to indicate that PSF is loading a font character set.

RLSTDFLG Flags valid only during the resource-delete call (RDC).

RLSTKEEP Keep or delete this resource. Before calling the exit for the RDC, PSF sets this flag in accordance with its current pruning algorithm. By modifying this field, the exit tells PSF which resources should remain loaded or should be deleted. If the field is set to '0', PSF deletes this resource. If it is set to '1', PSF does not delete this resource.

Note: PSF is not always able to follow the RLSTKEEP flag when deleting resources.

RLSTVFS The vertical size for this font, if a size is specified, can be set by the exit to scale an outline font. This is the recommended field for scaling an outline font.

RLSTHFS The horizontal size for this font, if a size is specified. When the vertical font size has not been specified, RLSTHFS can be changed to scale an outline font.

RLSTHSCF The horizontal scale factor for this font, if one is specified, can be set by the exit to contract or expand an outline font horizontally. This is known as anamorphic scaling. This field is not needed when the exit changes the point of a font but the normal character shape is maintained.

RLSTFEXT Parameter Area (Font Extension Area)

The font extension area is used when the code page and character set names are available for the font. The RLSTFEXT area consists of a header and an area, RLSTCPFM, that describes the font sections. RLSTCPFM is an array with a number of elements equal to the number of sections in the font. For information on defining fonts and the Map Coded Font (MCF) structured field, refer to *Advanced Function Printing: Host Font Data Stream Reference*.

Figure 73 on page 301 shows the format of RLSTFEXT.

```
Offsets  Type     Length  Name          Description
0    (0) CHARACTER     *  RLSTFEXT      FONT RLST EXTENSION
0    (0) CHARACTER     4  RLSTHDR       HEADER
0    (0) SIGNED        2  RLSTEXTC      NUMBER OF FONT EXTENSION
                                        REPEATING GROUPS
2    (2) CHARACTER     2  *             RESERVED
4    (4) CHARACTER    28  RLSTCPFN      CP/FN PAIR REPEATING GROUPS
4    (4) SIGNED        2  RLSTSID       SECTION ID NUMBER
6    (6) CHARACTER     2  RLSTPDSP      PDS PAIR
6    (6) UNSIGNED      1  RLSTCPDS      PDS NUMBER FOR CODE PAGE
7    (7) UNSIGNED      1  RLSTFNDS      PDS NUMBER FOR CHARACTER SET
8    (8) CHARACTER    16  RLSTPAIR      FONT PAIR
8    (8) CHARACTER     8  RLSTCPNM      CODE PAGE NAME
16  (10) CHARACTER     8  RLSTFNNM      FONT CHARACTER SET NAME
24  (18) BITSTRING     1  RLSTPFLG      PAIR FLAGS
     1... ....            RLSTCPUS      USERLIB CODE PAGE
     .1.. ....            RLSTFNUS      USERLIB FONT CHAR SET
     ..1. ....            RLSTCPIL      INLINE CODE PAGE
     ...1 ....            RLSTFNIL      INLINE FONT CHAR SET
     .... 1111            *             RESERVED
25  (19) CHARACTER     7  *             RESERVED
```

Figure 73. Format of the RLSTFEXT Parameter Area

RLSTFEXT Fields

RLSTHDR Header of the font extension area.

> **RLSTEXTC** Extension count field. PSF sets this field equal to the number of elements in the RLSTCPFN array. For a single-byte font, there is never more more than one element in the array. The exit must never modify this field.

RLSTCPFN An array in which each element describes a section of the font as a code page–character set pair.

> **RLSTSID** Section ID. PSF sets this field equal to the coded font section identifier from the MCF structured field.
>
> **RLSTPDSP** Contains the partitioned data set concatenation numbers for the code page and character set in this section. These numbers are available only during the resource-load call to the exit.
>
> > **RLSTCPDS** Contains the PDS concatenation number for the code page. If the code page resides in the system library, RLSTCPDS is the PDS concatenation number in the system library. If the code page resides in the user library, RLSTCPDS is the PDS concatenation number in the user library.
> >
> > **RLSTFNDS** Contains the PDS concatenation number for the character set. If the character set resides in the system library, RLSTFNDS is the PDS

concatenation number in the system library. If the character set resides in the user library, RLSTFNDS is the PDS concatenation number in the user library.

RLSTPAIR Contains the code page and character set names for this section. These fields are similar to the RLSTNAME field.

- **RLSTCPNM** Contains the code page name for this section. During the resource-access call to the exit, a new code page name is substituted if the exit changes the name in this field.
- **RLSTFNNM** Contains the character set name for this section. During the resource-access call to the exit, a new character set is substituted if the exit changes the name in this field.

RLSTPFLG Contains flags about the code page and character set for this field.

- **RLSTCPUS** If set, indicates that the code page is from the user library.
- **RLSTFNUS** If set, indicates that the character set is from the user library.
- **RLSTCPIL** If set, indicates that the code page is inline.
- **RLSTFNIL** If set, indicates that the character set is inline.

Processing Indicator Cross-Reference for APSUX07

Table 19 (Page 1 of 3). Input and Output Summary for APSUX07 Processing Indicators. Input is marked "I," and output is marked "O." If the indicator is both, it is marked "IO." For definitions of these fields, see "XTP7 Fields" on page 284 and "APSURLST Fields" on page 296.

	INIT	BDSC	RAC	RLC	RDC	TERM
XTP7ETYP	I	I	I	I	I	I
XTP7RTYP			I	I	I	
XTP7DSTY						
XTP7AUX			I	I		
XTP7PDS		I	I	I	I	
XTP7DSAT						
XTP7PDFT			I	I		
XTP7PJHD			I	I		
XTP7PJTR			I	I		
XTP7PDSH			I	I		
XTP7PMDS			I	I		

Table 19 (Page 2 of 3). Input and Output Summary for APSUX07 Processing Indicators. Input is marked "I," and output is marked "O." If the indicator is both, it is marked "IO." For definitions of these fields, see "XTP7 Fields" on page 284 and "APSURLST Fields" on page 296.

	INIT	BDSC	RAC	RLC	RDC	TERM
XTP7PLDS		I			I	
XTP7ACSE	IO	IO				
XTP7IPCC	IO	IO				
XTP7NOTY	IO	IO	IO	IO	IO	
XTP7RFLG						
XTP7ETFF	IO	IO				
XTP7TDS		O	O	O		
XTP7EFFH	IO	IO				
XTP7HDF	IO	IO	IO	IO	IO	IO
XTP7HQUE		O	O	O		
XTP7PRNS	IO					
XTP7PRND	IO					
XTP7IPMF	IO	IO				
XTP7IPOS	IO	IO				
XTP7MTOF	IO					
XTP7RFNF	IO					
XTP7IPSU	IO					
XTP738MO	IO	IO				
XTP7MDSD	IO	IO				
XTP7HCA	IO					
XTP7C2SI	IO	IO				
XTP7RRLV	IO	IO	IO	IO	IO	
XTP7RLST			I	I	I	
XTP7OVLY			I	I		
XTP7PINF	I	I	I	I	I	I
XTP7MSGP and MSGL		O	O	O		
XTP7CRLF	IO					
XTP7CRF2	IO					
XTP7CRLP	I					
XTP7MFLG						
XTP7JHMO	IO	IO				
XTP7JTMO	IO	IO				
XTP7DHMO	IO	IO				
XTP7MDMO	IO	IO				
XTP7RSTR						IO
XTP7ABRC						I
RLSTPTRS						
RLSTPRVS			I	I	I	
RLSTNEXT			I	I	I	
RLSTEXTP			I	I	I	
RLSTMSGP			O			
RLSTRID			I	I	I	

Table 19 (Page 3 of 3). Input and Output Summary for APSUX07 Processing Indicators. Input is marked "I," and output is marked "O." If the indicator is both, it is marked "IO." For definitions of these fields, see "XTP7 Fields" on page 284 and "APSURLST Fields" on page 296.

	INIT	BDSC	RAC	RLC	RDC	TERM
RLSTNAME			IO	I	I	
RLSTPDS			I	I	I	
RLSTATTR			I	I	I	
RLSTSTAT						
RLSTAUX			I	I	I	
RLSTDEP			I	I	I	
RLSTNEST			I	I	I	
RLSTPLR			IO	IO	IO	
RLSTRESC			I	I	I	
RLSTMSGL			O			
RLSTFLGS						
RLSTMISC			I	I	I	
RLSTAFLG			IO			
RLSTLFLG				I		
RLSTDFLG					IO	

Appendix D. APTRCONV Resource Conversion Utility for VSE

APTRCONV is a utility program that resides in PSF. Its purpose is to convert resources so that VSE can use them.

You can run APTRCONV with JCL to convert an AFP resource in a partitioned data set to a VSE job stream. APTRCONV assembles and link-edits the resource into a VSE library**.**sublibrary.

Converting a Resource

Figure 74 shows an example of the APTRCONV control statements by which a PAGEDEF resource can be converted for use by VSE.

```
//USER1T  JOB (SYS10000),CLASS=A,MSGCLASS=T,
//    USER=USER1T,REGION=1024K
//S1      EXEC PGM=APTRCONV,PARM=(P1075A0,PAGEDEF,TEST)
//RESC      DD DSN=SYS1.PAGEDEF(P1075A0),DISP=SHR
//SYSPCH    DD DSN=USER1T.CONVOUT,
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120),
//    SPACE=(TRK,(5,5)),DISP=(,CATLG),
//    UNIT=3380,VOL=SER=TEMP00
```

Figure 74. Sample JCL for Converting the MVS PAGEDEF Resource for Use by VSE

The sample JCL in Figure 74 uses the following parameters:

P1075A0
Specifies the name of the resource converted.

PAGEDEF
Specifies the VSE library in which the converted resource is to be catalogued.

TEST
Specifies the VSE sublibrary in which the converted resource is to be catalogued.

Note: The converted resource is entered in the VSE library.sublibrary under the same name as the MVS resource.

When a job is being networked from JES to POWER, a special POWER utility name should be used to route the job to the reader rather than to the punch. For example:

```
//OUTPUT1   OUTPUT DEST=xxx.PWR$JOB
//SYSPCH    DD SYSOUT=x,OUTPUT=*.OUTPUT1
```

Figure 75. Sample JCL to Route Job to Reader

The sample JCL in Figure 75 uses the following parameters:

xxx
Specifies the POWER NJE name.

PWR$JOB
Specifies a special utility to route the job to the VSE reader rather than to the punch.

x Specifies the MVS installation-defined punch class, for example, *b*.

Note: With this utility, the converted resource job stream runs without operator intervention.

Appendix E. Reasonable Resource Loading Values (RRLV)

This appendix lists the default reasonable resource loading values. (See also “Partial Control of Resource Deletion” on page 211.)

Note: PSF deletes page definitions and form definitions from virtual storage, not from printer storage.

Table 20. Reasonable Resource Loading Values (RRLV)

Resource	Device	Default value
Page definitions	3800 3820 All other printers	6 6 6
Form definitions	3800 3820 All other printers	6 6 6
Overlays	3800 3820 All other printers	0 0 0
Page segments	3800 3820 All other printers	0 0 0
Coded fonts	3800 3800 printer with storage expansion feature 3820 All other printers	10 64 24 PRT[34]
Object containers	3800 3800 with storage expansion feature 3820 All other printers	0 0 0 0

[34] PRT is derived from printer information (storage size).
Six fonts are left in the printer storage for each 256K bytes of pattern storage available—for example, 48 fonts for 2M bytes.

Appendix F. Font Conversion Program, 240-pel to 300-pel (APSRCF30)

Some JCL jobs that convert fonts are provided with PSF. They are intended as samples; as provided, they might not be suitable for your requirements.

APSRCF30 is a font conversion utility program provided with PSF. The program uses your currently installed 240-pel fonts as input, converts (resizes) them to 300-pel fonts, and places the converted output into a partitioned data set. APSRCF30 uses 24-bit addressing; it does not use XA instructions.

Understanding APSRCF30

If you install a 300-pel printer, you must have a 300-pel font library to use with the printer. Use the APSRCF30 font conversion utility program only if you do not have a 300-pel font library containing the fonts needed for your printer. For more information about 300-pel printers, refer to *AFP: Printer Information*.

Using APSRCF30

APSRCF30 is shipped as part of PSF. After PSF has been installed, if you do not have a 300-pel font library, you can run APSRCF30 to convert 240-pel fonts into 300-pel fonts.

Note: The IBM Core interchange raster fonts are shipped in both 240-pel and 300-pel resolutions; therefore, you do not have to convert them. Outline fonts, of course, are resolution-independent and do not need to be converted.

Before converting fonts with APSRCF30, you need to:

- Define a new font library for 300-pel fonts, if one does not already exist. SYS1.FONT300 is the recommended name for your 300-pel system font library.

  You should use the existing 240-pel font library as a model, and allocate the new 300-pel font library similarly. The 300-pel font library will require more space than the 240-pel font library.

- Create a list of 240-pel fonts that are to be converted.

  Only the character set members whose filenames start with *C0* have to be converted with APSRCF30. See “Converting All 240-Pel Fonts in Source Library (APSRAL30)” on page 312; the sample JCL will convert all members whose filenames start with *C0*.

  – Coded font members whose filenames start with *X0* should not be used as input to APSRCF30.
  – Code page members whose filenames start with *T1* should not be used as input to APSRCF30.

- Make any changes in the JCL for APSRCF30 that are needed for your installation. See “APSRCF30 Sample Jobs” on page 311.

  The JCL supplied with APSRCF30 should be changed to indicate the font member input required. The JCL uses the font member names for a few sample fonts as input to NAME on the PROC statement. You must specify the desired font member names, using NAME on the PROC statement. For example, in the JCL for APSRCF30, specify:

```
C0L0AD10  EXEC  PROC=CON,NAME=C0L0AD10
C0L0AD12  EXEC  PROC=CON,NAME=C0L0AD12
⋮
C0S0AP13  EXEC  PROC=CON,NAME=C0S0AP13
```

- Submit the JCL.

After converting fonts with APSRCF30, you need to:

- Change the PSF startup procedure to use the new 300-pel font library.
- Arrange for PSF to have access to the code pages and coded fonts that have not been converted. Either copy the code pages and the coded fonts from the 240-pel library to the 300-pel library, or specify font library concatenation so that the 300-pel library is accessed before the 240-pel library. For example:

```
//FONT01    DD DSN=SYS1.FONT300,DISP=SHR
//          DD DSN=SYS1.FONTLIBB,DISP=SHR
```

- If you have specified both a 240-pel printer and a 300-pel printer in the same PSF startup procedure, create DD statements for each font library. The FONT01 DD statement should contain only the 240 concatenation list, and the FONT02 DD statement should contain the 300-pel library (concatenated ahead of the 240-pel libraries). The PRINTDEV statement for each printer must contain a separate reference to the correct FONT DD statement.

Error Handling

APSRCF30 detects each of the errors listed below, and places the return code (RC) for it in register 15. The return code prints on the screen when you are using the APSRCF30 EXEC. When an error occurs, the font-conversion program may create an incomplete output file, which you should discard.

Return Code **Meaning**

4 One or more of the conversion character data set members required by the conversion program could not be found. The DD of the required conversion character data set is FONTB300.

20 An I/O error occurred while APSRCF30 was writing to the output data set.

28 The input font data set either could not be found or could not be opened. This error can occur if APSDAT00 (the conversion character data set) was allocated with record format FBM rather than VBM.

32 The font named in the input data set is invalid, for any of the following reasons:

- The member contains invalid structured fields.
- The font is not bounded-box.
- The font is not 240-pel.

44 An unexpected end-of-file occurred on the input data set.

88 Insufficient virtual storage was available for program execution.

99 Invalid font data. This return code is presented if any of the following structured fields in a font has a length less than 1:

FNI	Font index
FNM	Font patterns map
FNG	Font patterns

APSRCF30 Considerations

Do not convert the Bar Code|Optical Character Recognition font (Product Number 5688-021). The conversion process usually distorts the font, making it unreadable by an optical scanning device. Check all the JCL for this font before running APSRCF30.

APSRCF30 uses a comparatively large amount of CPU resources, especially if started on a smaller System/390 processor complex. To run APSRCF30, you should have a minimum region size of 5MB.

Before you invoke APSRCF30, you should allocate the target font library for the conversion.

For an example of JCL that can be changed, see "APSRCF30 Sample Jobs."

APSRCF30 has no PARM statement; it uses the following ddnames:

- SYSUT1: Input font library (including member name); uses DSORG=PS.
- SYSUT2: Output font library (including member name); uses DSORG=PS.
- FONTB300: Conversion character data set, installed with PSF (used for conversion of special characters); uses DSORG=PO. The default name is SYS1.APSDAT00.

APSRCF30 Sample Jobs

Sample font conversion jobs are shipped with PSF in SYS1.SAMPLIB. You can change them to convert bounded-box fonts from 240-pel fixed metrics to 300-pel relative metrics by using a PROC invocation of APSRCF30. The PROC statements supplied are provided as samples to assist you in tailoring JCL to reflect the bounded-box fonts that require converting.

You should allocate the font library used by the PROC before invoking APSRCF30.

You can tailor APSWAL30, which is in SYS1.SAMPLIB, to convert all the 240-pel fonts in your library to 300-pel fonts. The sample JCL for APSWAL30 is given in "Converting All 240-Pel Fonts in Source Library (APSRAL30)" on page 312.

If you select some of your 240-pel fonts to be converted, rather than converting all the fonts in your source library, see "Example for Selected Font Conversion" on page 314 for sample JCL (APSWCF30) that you can modify and use when tailoring the JCL from your source library or from the individual sample jobs in SYS1.SAMPLIB. The JCL you use may differ from the example.

Converting All 240-Pel Fonts in Source Library (APSRAL30)

You can change the following sample JCL and use it to convert all the font character sets (all members with prefix *C0*) in your source library from 240-pel fixed metrics to 300-pel relative metrics.

```
//APSWAL30 JOB 'account #','name,MSGLEVEL=(1.1)
//************************************************************
//* APSWAL30 SAMPLE JOB TO CREATE JCL FOR ALL PDS MEMBERS
//************************************************************
//*
//NAME    EXEC PGM=APSRAL30
//*
//* SYSUT1 IS 240-PEL FONTS TO BE CONVERTED TO 300-PEL
//SYSUT1    DD DSN=SYS1.FONTLIBB,DISP=SHR
//*
//* SYSUT2 IS USER'S JCL IN PDS OR FILE SEQ OR DUMMY
//SYSUT2    DD DSN=SYS1.SAMPLIB(APSWALHD),DISP=SHR
//*
//* SYSUT3 IS JCL AND EXEC PROCEDURE CARDS, FILE SEQ OR PDS WITH MEMBER NAME
//SYSUT3    DD DSN=your.runjob.dataout(out),DISP=SHR
//*
```

The following inline procedure (“CON”) is referenced by SYSUT2, and is shipped in SYS1.SAMPLIB under the name APSWALHD. This JCL is not submitted; it is copied into the output data set specified in SYSUT3 by APSRAL30.

```
//APSWALHD JOB 'account #','name,MSGLEVEL=(1.1)
//*
//CON       PROC NAME=
//S1        EXEC PGM=APSRCF30,REGION=5120K
//*                         INSERT YOUR INPUT 240-PEL FONT LIBRARY
//SYSUT1    DD DISP=SHR,DSN=SYS1.FONTLIBB(&NAME)
//*                         INSERT YOUR OUTPUT 300-PEL FONT LIBRARY
//SYSUT2    DD DISP=SHR,DSN=SYS1.FONT300(&NAME)
//*                         CONVERSION CHARACTER DATA SET DDNAME
//FONTB300  DD DISP=SHR,DSN=SYS1.APSDAT00
//          PEND
//*
//*
```

In the JCL to run APSRCF30, you must list every character set member that is to be converted. However, since the APSRAL30 utility is provided, you do not have to list the name of every character set to be converted. APSRAL30 creates the JCL for APSRCF30 (including the list of character set members) for a 240-pel font data set. To convert the font data set, submit the JCL created by APSRAL30.

Input to APSRAL30

SYSUT1 Specifies the 240-pel font data set to be converted.

SYSUT2 Specifies the data set containing the JCL that is to be copied into the APSRCF30 JCL. SYS1.SAMPLIB (APSWALHD) provides an example of the JCL required.

Output from APSRAL30

SYSUT3 Specifies the output data set. It contains the APSRCF30 JCL to convert every character set member in a font data set.

APSRAL30 Processing

APSRAL30 copies the JCL in the data set specified by SYSUT2 into the SYSUT3 output data set. This is the JCL to invoke APSRCF30. APSRAL30 then reads the directory of the 240-pel font data set. It creates an entry in the SYSUT3 output data set for every character set found.

Note: A character set is identified by the prefix *C0*.

After APSRAL30 is run, the output data set contains the JCL to convert all the font character set members of a font data set. To convert the character sets, you must submit the output data set JCL. You can modify the output data set JCL before you submit it. For example, if you want to leave some of the font character sets unconverted, delete their member names from the output data set.

To run APSRAL30:

1. Allocate the output data set specified in SYSUT3.

   This should be a fixed-block data set with a record length of 80. It can be sequential, or partitioned with a member name; the member name does not need to be predetermined.

2. Tailor the JCL the data set specified by SYSUT2.

   A sample JCL file, APSWALHD, is provided in SYS.SAMPLIB.

3. Tailor the sample APSRAL30 JCL to your installation.
4. Submit APSRAL30.
5. Optionally, edit the APSRCF30 JCL created by APSRAL30.
6. Submit the APSRCF30 JCL created by APSRAL30.
7. Make sure that PSF has access to the code page and coded font members from the 240-pel font data set.

   You can copy the code page and coded font members from the 240-pel font data set to the 300-pel font data set. Or, you can concatenate the 240-pel font data set after the 300-pel font data set in the PSF startup procedure.

APSRAL30 Considerations

Some considerations include:

- SYSUT1 cannot contain a concatenation list, because only the PDS directory is being processed.
- The JCL contained in the input data set SYSUT2 is output to your SYSUT3 data set. SYSUT2 can be a PDS with a member name or a sequential data set. It can also be a JCL DUMMY statement, in which case only the EXEC PROC statements will be output to your SYSUT3 data set.
- The output on SYSUT3 contains the JCL from SYSUT2, followed by up to 255 EXEC PROC statements representing the character sets in SYSUT1.

  If the input PDS directory on SYSUT1 has more than 255 members whose names begin with *C0*, SYSUT3 will contain the SYSUT2 JCL repeated with up to another 255 EXEC PROC statements. This process will continue until all the SYSUT1 PDS members whose names begin with *C0* are processed. MVS has a limitation of 255 EXEC statements in a job.

- You can edit the SYSUT3 output before submitting the job.
- The output of this program always refers to the inline procedure “CON.”

Example for Selected Font Conversion

Using the following format, and updating the JCL as needed for the font you have selected, you can convert font character sets from 240-pel fixed metrics to 300-pel relative metrics.

```
//APSWCF30 JOB 'account #','name',MSGLEVEL=(1,1)
//*********************************************************************
//* THIS IS A SAMPLE JOB USED TO CONVERT A GIVEN BOUNDED-BOX          *
//* CHARACTER SET FROM 240-PEL FIXED METRICS TO 300-PEL RELATIVE      *
//* METRICS, USING A PROC INVOCATION OF THE APSRCF30 FONT SCALING     *
//* PROGRAM.  THE SUPPLIED PROC STATEMENTS AND CHARACTER SETS ARE     *
//* PROVIDED AS SAMPLE ONLY, AND SHOULD BE TAILORED TO REFLECT THE    *
//* BOUNDED-BOX CHARACTER SETS THAT REQUIRE SCALING.                  *
//*                                                                   *
//* THE FONT LIBRARY USED BY THIS PROC SHOULD BE ALLOCATED BEFORE     *
//* THIS PROCEDURE IS INVOKED.                                        *
//*                                                                   *
//* NOTE: IF CONCATENATION OF FONT LIBRARIES WILL NOT BE PERFORMED    *
//*       WITH PSF, THE CODE PAGES AND CODED FONTS SHOULD BE COPIED   *
//*       TO THE 300-PEL FONT LIBRARY.                                *
//*                                                                   *
//* DDNAME USAGE:                                                     *
//* ============                                                      *
//* SYSUT1   - INPUT FONT LIBRARY (MEMBER NAME REQUIRED BY PROC)      *
//* SYSUT2   - OUTPUT FONT LIBRARY (MEMBER NAME REQUIRED BY PROC)     *
//* FONTB300 - CONVERSION CHARACTER DATA SET                          *
//*                                                                   *
//*********************************************************************
//*
//CON       PROC NAME=
//S1        EXEC PGM=APSRCF30,REGION=5120K
//*              INSERT YOUR INPUT 240-PEL FONT LIBRARY
//SYSUT1    DD DISP=SHR,DSN=SYS1.FONTLIBB(&NAME)
//*              INSERT YOUR OUTPUT 300-PEL FONT LIBRARY
//SYSUT2    DD DISP=SHR,DSN=SYS1.FONT300(&NAME)
//*              CONVERSION CHARACTER DATA SET DDNAME
//FONTB300  DD DISP=SHR,DSN=SYS1.APSDAT00
//          PEND
//*
//*********************************************************************
//* SAMPLE - CONVERT APL2 BOUNDED-BOX CHARACTER SET FROM 240-PEL      *
//*         FIXED METRICS TO 300-PEL RELATIVE METRICS                 *
//* PRODUCT NAME  : APL2                                              *
//* PRODUCT NUMBER: 5771-ADB                                          *
//* PARTLIST      : CBIPOADB PARTLIST                                 *
//*********************************************************************
//C0L0AD10  EXEC PROC=CON,NAME=C0L0AD10
//C0L0AD12  EXEC PROC=CON,NAME=C0L0AD12
//C0L0AG10  EXEC PROC=CON,NAME=C0L0AG10
```

```
//C0L0AG12  EXEC PROC=CON,NAME=C0L0AG12
//C0L0AG15  EXEC PROC=CON,NAME=C0L0AG15
//C0L0AI10  EXEC PROC=CON,NAME=C0L0AI10
//C0L0AI12  EXEC PROC=CON,NAME=C0L0AI12
//C0L0AT10  EXEC PROC=CON,NAME=C0L0AT10
//C0L0AT12  EXEC PROC=CON,NAME=C0L0AT12
//C0L00APL  EXEC PROC=CON,NAME=C0L00APL
//C0S0AP10  EXEC PROC=CON,NAME=C0S0AP10
//C0S0AP13  EXEC PROC=CON,NAME=C0S0AP13
//C0S0AP20  EXEC PROC=CON,NAME=C0S0AP20
```

Appendix G. Installation Verification Program (IVP) Example

The following is a sample of output obtained by running the installation verification program (IVP) on a 240-pel resolution printer.

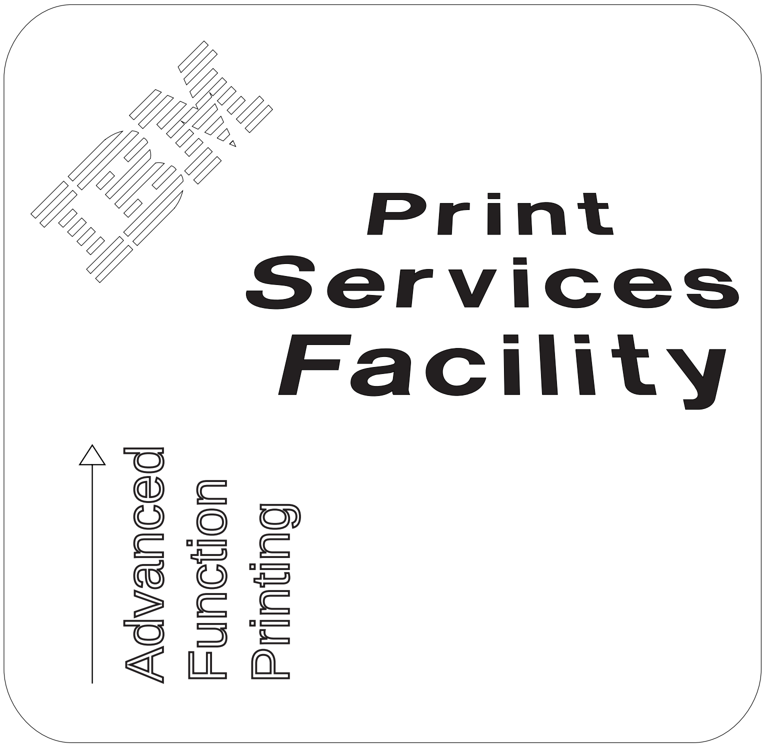


Figure 76. IVP Output Printed with 240-Pel Resolution

Appendix H. PSF Programming Interfaces

The interfaces identified in this appendix are provided for use by a customer installation in writing programs that use the services of PSF. Only the interfaces identified in this appendix should be used to request or receive the services of PSF.

General-Use Programming Interfaces

None

Product-Sensitive Programming Interfaces

APSGEXTP

APSLCRL

APSXUCOM

APSUBTBL

APSUECA

APSURLST

APSUXP16

Interfaces for Tailoring PSF/MVS

APSRFTBL

APSRGTNM

APSRNTGM

Installation-Wide Exits

APSUX01

APSUX02

APSUX03

APSUX04

APSUX05

APSUX06

APSUX07

APSUX15

APSUX16

Appendix I. Mapping Tables for Outline Fonts

This appendix contains two PSF tables:

- APSRGTNM, which maps GRIDs (global resource identifiers) to font names (for either raster fonts or outline fonts)
- APSRNTGM, which maps font names to GRIDs

For information about using these tables, see “Using the Font Mapping Tables” on page 146.

The following fields are in the font mapping tables:

FGID Font global identifier

FW Font width (1440ths of an inch)

CPGID Code page global identifier

GCSGID Graphic character set global identifier

FLAGS Reserved

TEKID The technology identifier for the font character set is:

X'05' Raster
X'1F' Single-byte outline

VFS Vertical font size for the font character set (1440ths of an inch). To find the VFS, multiply the point size of the raster font, 1/72 of an inch, by 20.

NAME The member name of the code page or character set

Tables 21 and 22 give samples of the information in the tables:

Table 21. Font-Name-to-Grid Table, APSRNTGM

FCS NAME	FLAGS	TEKID	FGID[1]	GCSGID	FW	VFS
C0D0GT10	0	X'05'	0040	0000	000	000
CZN200	0	X'1F'	2308	2039	0	0
C0N200B0	0	X'05'	2308	2039	80	240
FFFFFFFF	X'FF'	X'FF'	X'FFFF'	X'FFFF'	X'FFFF'	X'FFFF'

[1] Specifying zeros (0) for FGID means do *not* map fonts.

Table 22. Grid-to-Font-Name Table, APSRGTNM

FGID	GCSGID	FW	FLAGS	TEKID	VFS	FCS NAME
416	1261	120	0	X'05'	200	C0420200
416	2039	120	0	X'05'	200	C0420000
416	2039	0	0	X'1F'	0	CZ4200
X'FFFF'	X'FFFF'	X'FFFF'	X'FF'	X'FF'	0	FFFFFFFF

Figure 77 maps global resource identifiers (GRIDs) to font names (for either raster fonts or outline fonts). Use this module to add mapping entries to PSF's default grid-to-member-names tables, to change entries, or to disable entries.

```
APSRGTNM - GRID PARTS TO FONT MEMBER NAMES TABLES
*/****START OF SPECIFICATIONS********************EXTPROC**************/
*/*                                                                  */
*/*      MODULE NAME: APSRGTNM                                       */
*/*                                                                  */
*/*      DESCRIPTIVE NAME: GRID-TO-NAMES USER MAPPING TABLE          */
*/*                                                                  */
*/*      FUNCTION: THIS MODULE PROVIDES THE ABILITY FOR THE USER     */
*/*                TO ADD ENTRIES FOR CODE PAGE AND CHARACTER SET    */
*/*                GRID-TO-MEMBER-NAME MAPPING.                  @01C*/
*/*                                                                  */
*/*      NOTES:                                                      */
*/*         DEPENDENCIES: NONE                                       */
*/*         RESTRICTIONS: NONE                                       */
*/*         REGISTER CONVENTIONS: NONE                               */
*/*         PATCH LABEL: NONE                                        */
*/*                                                                  */
*/*      MODULE TYPE:  PROCEDURE                                     */
*/*         PROCESSOR: OS ASSEMBLER H                                */
*/*         ATTRIBUTES: NO EXECUTABLE CODE                           */
*/*                                                                  */
*/*      ENTRY POINT: APSRGTFN                                       */
*/*         PURPOSE:  THIS TABLE MAPS GRID PARTS (FGID,              */
*/*                   GCSGID, FW) AND VERTICAL FONT SIZE TO THEIR    */
*/*                   CORRESPONDING MEMBER NAMES.  EACH ENTRY ALSO   */
*/*                   HAS A TECHNOLOGY ID ASSOCIATED WITH IT. THE    */
*/*                   TABLE IS TERMINATED BY A STRING OF XL2'FFFF'.  */
*/*                                                              @01C*/
*/*                   THE FORMAT OF EACH ENTRY IS AS FOLLOWS.        */
*/*                   BYTES  0-1  FONT GLOBAL ID (FGID)              */
*/*                   BYTES  2-3  GRAPHIC CHARACTER SET GLOBAL ID    */
*/*                               (GCSGID)                       @01C*/
*/*                   BYTES  4-5  FONT WIDTH                         */
*/*                   BYTES   6   FLAGS - RESERVED, MUST BE ZERO @01A*/
*/*                   BYTES   7   TECHNOLOGY ID                      */
*/*                               VALID TECHNOLOGY IDS ARE:          */
*/*                               X'05' RASTER (SINGLE-BYTE)         */
*/*                               X'1F' TYPE 1 OUTLINE (SINGLE-BYTE) */
*/*                   BYTES  8-9  VERTICAL FONT SIZE                 */
*/*                   BYTES 10-17 FONT CHARACTER SET NAME            */
*/*         LINKAGE:  NONE                                           */

                                          */
```

Figure 77 (Part 1 of 3). Global Resource Identifiers (GRIDs) to Font Names

```
*/*      ENTRY POINT: APSRGTCP                                        */
*/*         PURPOSE:  THIS TABLE MAPS GRID PARTS (CPGID          @01C*/
*/*                   AND GCSGID) TO THEIR CORRESPONDING CODE PAGE   */
*/*                   MEMBER NAMES.  THE TABLE IS TERMINATED BY A    */
*/*                   STRING OF XL2'FFFF'.  THE FORMAT OF EACH       */
*/*                   ENTRY IS AS FOLLOWS.                           */
*/*                   BYTES  0-1  CODE PAGE GLOBAL ID (CPGID)        */
*/*                   BYTES  2-3  CODE PAGE GRAPHIC CHARACTER SET    */
*/*                               GLOBAL ID (GCSGID)                 */
*/*                   BYTES  4-11 CODE PAGE NAME                     */
*/*         LINKAGE:  NONE                                           */
*/*                                                                  */
*/*      MESSAGES: NONE                                              */
*/*                                                                  */
*/*      CHANGE ACTIVITY                                             */
*/*                                                                  */
*/* $00=LASP0006,HPRF220, 930903, BJ13ATF: RELEASE 2.2.0         @00A*/
*/* $01=OW08340, HPRF220, 941020, BJ13DDA: Outline Font D-APAR   @01A*/
*/*                                                                  */
*/**********************END OF SPECIFICATIONS*************************/
*/*                                                                  */
*/*                       BAL CODE FOLLOWS                           */
*/*                                                                  */
*/********************************************************************/
APSRGTNM CSECT              ENTRY POINT OF THIS MODULE
APSRGTNM AMODE 31           DO NOT REMOVE/CHANGE
APSRGTNM RMODE ANY          DO NOT REMOVE/CHANGE
GTFN@    DC    V(APSRGTFN)  DO NOT REMOVE/CHANGE
GTCP@    DC    V(APSRGTCP)  DO NOT REMOVE/CHANGE
******************************************************************
* THE FOLLOWING IS THE GRID-PARTS-TO-CHARACTER-SET TABLE.        *
******************************************************************
         ENTRY APSRGTFN     DO NOT REMOVE/CHANGE
APSRGTFN DS    0D           DO NOT REMOVE/CHANGE
******************************************************************
**THE FOLLOWING 6 LINES ARE FOR FGID 2308 (X'0904').             *
**THIS ENTRY PROVIDED TO ILLUSTRATE THE FORMAT OF A TABLE ENTRY. *
******************************************************************
**       DC   H'2308'       FGID AS DECIMAL NUMBER
**                           (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'2039'       GCSGID AS DECIMAL NUMBER
**                           (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0080'       FONT WIDTH AS DECIMAL NUMBER
**                           (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC XL1'00'         RESERVED, MUST BE ZERO.
**       DC XL1'05'         TECHNOLOGY ID
**       DC   H'0240'       VERTICAL FONT SIZE AS DECIMAL NUMBER
**                           (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC CL8'C0N200B0'   FONT CHARACTER SET NAME
                                          */
```

Figure 77 (Part 2 of 3). Global Resource Identifiers (GRIDs) to Font Names

```
******************************************************************
**THE FOLLOWING 6 LINES MARK THE END OF APSRFNTG.                *
******************************************************************
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL1'FF'          DO NOT REMOVE/CHANGE
         DC XL1'FF'          DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC CL8'FFFFFFFF'    DO NOT REMOVE/CHANGE
******************************************************************
* THE FOLLOWING IS THE CODE-PAGE-TO-GRID PARTS TABLE.            *
******************************************************************
         ENTRY APSRGTCP      DO NOT REMOVE/CHANGE
APSRGTCP DS    0D            DO NOT REMOVE/CHANGE
******************************************************************
**THE FOLLOWING 3 LINES ARE FOR CPGID 500 (X'01F4').             *
**THIS ENTRY PROVIDED TO ILLUSTRATE THE FORMAT OF A TABLE ENTRY. *
******************************************************************
**       DC   H'0500'        CPGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0697'        CODE PAGE GCSGID AS DECIAML NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC CL8'T1V10500'    CODE PAGE NAME
******************************************************************
**THE FOLLOWING 3 LINES MARK THE END OF APSRCPTG.                *
******************************************************************
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC CL8'FFFFFFFF'    DO NOT REMOVE/CHANGE
******************************************************************
**THE FOLLOWING LINE MARKS THE END OF APSRGTNM.                  *
******************************************************************
         END   APSRGTNM      DO NOT REMOVE/CHANGE
```

Figure 77 (Part 3 of 3). Global Resource Identifiers (GRIDs) to Font Names

Figure 78 maps font names to global resource identifiers (GRIDs). Use this module to add mapping entries to PSF's default member-names-to-grid tables, change entries, or disable entries.

```
APSRNTGM - FONT MEMBER NAMES TO GRID PARTS TABLES
*/****START OF SPECIFICATIONS********************EXTPROC**************/
*/*                                                                  */
*/*      MODULE NAME: APSRNTGM                                       */
*/*                                                                  */
*/*      DESCRIPTIVE NAME: NAMES-TO-GRID USER MAPPING TABLE          */
*/*                                                                  */
*/*      FUNCTION: THIS TABLES PROVIDES THE ABILITY FOR THE USER     */
*/*                TO ADD ENTRIES FOR CODE PAGE AND CHARACTER SET    */
*/*                MEMBER NAME-TO-GRID MAPPING.                  @01C*/
*/*                                                                  */
*/*      NOTES:                                                      */
*/*         DEPENDENCIES: NONE                                       */
*/*         RESTRICTIONS: NONE                                       */
*/*         REGISTER CONVENTIONS: NONE                               */
*/*         PATCH LABEL: NONE                                        */
*/*                                                                  */
*/*      MODULE TYPE:  PROCEDURE                                     */
*/*         PROCESSOR: OS ASSEMBLER H                                */
*/*         MODULE SIZE: 900                                         */
*/*         ATTRIBUTES: READ ONLY, NO EXECUTABLE CODE                */
*/*                                                                  */
*/*      ENTRY POINT: APSRFNTG                                       */
*/*         PURPOSE:  THIS TABLE MAPS FONT CHARACTER SET             */
*/*                   MEMBER NAMES TO THE CORRESPONDING GRID PARTS   */
*/*                   (FGID, GCSGID, FW) AND A VERTICAL FONT SIZE.   */
*/*                   A TECHNOLOGY ID IS ALSO ASSOCIATED WITH EACH   */
*/*                   ENTRY.                                     @01C*/
*/*                   THE FORMAT OF EACH ENTRY IS AS FOLLOWS:        */
*/*                   BYTES  0-7  FONT CHARACTER SET MEMBER NAME     */
*/*                   BYTE     8  FLAGS -RESERVED, MUST BE ZERO. @01A*/
*/*                   BYTE     9  TECHNOLOGY ID                      */
*/*                               VALID TECHNOLOGY IDS ARE:          */
*/*                               X'05'  RASTER (SINGLE-BYTE)        */
*/*                               X'1F'  TYPE 1 OUTLINE (SINGLE-BYTE)*/
*/*                   BYTES 10-11 FONT GLOBAL ID (FGID)              */
*/*                   BYTES 12-13 GRAPHIC CHARACTER SET GLOBAL ID    */
*/*                               (GCSGID)                       @01C*/
*/*                   BYTES 14-15 FONT WIDTH                         */
*/*                   BYTES 16-17 VERTICAL FONT SIZE                 */
*/*                   THE TABLE IS TERMINATED BY A STRING OF         */
*/*                   CL8'FFFFFFFF'.                                 */
*/*         LINKAGE:  NONE                                           */
```

Figure 78 (Part 1 of 3). Font Names to Global Resource Identifiers (GRIDs)

```
*/*      ENTRY POINT: APSRCPTG                                      */
*/*         PURPOSE:  THIS TABLE MAPS CODE PAGES TO                 */
*/*                   THE CORRESPONDING GRID PARTS(CPGID AND GCSGID).*/
*/*                   THE FORMAT OF EACH ENTRY IS AS FOLLOWS        */
*/*                   BYTES  0-7  CODE PAGE NAME                    */
*/*                   BYTES  8-9  CODE PAGE GLOBAL ID (CPGID)       */
*/*                   BYTES 10-11 CODE PAGE GRAPHIC CHARACTER SET   */
*/*                               GLOBAL ID (GCSGID)                */
*/*                   THE TABLE IS TERMINATED BY A STRING OF        */
*/*                   CL8'FFFFFFFF'.                                */
*/*         LINKAGE:  NONE                                          */
*/*                                                                 */
*/*      MESSAGES: NONE                                             */
*/*                                                                 */
*/*      CHANGE ACTIVITY                                            */
*/*                                                                 */
*/* $00=LAPS0006,HPRF220, 090393, BJ13ATF: RELEASE 2.2.0        @00A*/
*/* $01=OW08340, HPRF220, 102094, BJ13DDA: Outline Font D-APAR   @01A*/
*/*                                                                 */
*/**********************END OF SPECIFICATIONS*************************/
*/*                                                                 */
*/*                       BAL CODE FOLLOWS                          */
*/*                                                                 */
*/********************************************************************/
APSRNTGM CSECT               ENTRY POINT OF THIS MODULE
APSRNTGM AMODE 31            DO NOT REMOVE/CHANGE
APSRNTGM RMODE ANY           DO NOT REMOVE/CHANGE
FNTG@    DC    V(APSRFNTG)   DO NOT REMOVE/CHANGE
CPTG@    DC    V(APSRCPTG)   DO NOT REMOVE/CHANGE
*******************************************************************
* THE FOLLOWING IS THE CHARACTER-SET-TO-GRID-PARTS TABLE.         *
*******************************************************************
         ENTRY APSRFNTG      DO NOT REMOVE/CHANGE
APSRFNTG DS    0D            DO NOT REMOVE/CHANGE
*******************************************************************
**CORE TIMES NEW ROMAN - MEDIUM - RASTER 12 POINT                 *
**THE FOLLOWING 6 LINES ARE FOR LATIN.  THIS ENTRY IS PROVIDED    *
**TO ILLUSTRATE THE FORMAT OF EACH ENTRY.                         *
*******************************************************************
**       DC CL8'C0N200B0'    FONT CHARACTER SET NAME
**       DC XL1'00'          RESERVED, MUST BE ZERO
**       DC XL1'05'          TECHNOLOGY ID
**       DC   H'2308'        FGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'2039'        GCSGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0080'        FONT WIDTH AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0240'        VERTICAL FONT SIZE AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
```

Figure 78 (Part 2 of 3). Font Names to Global Resource Identifiers (GRIDs)

```
******************************************************************
**CORE TIMES NEW ROMAN - MEDIUM - OUTLINE FONT                   *
**THE FOLLOWING 6 LINES ARE FOR LATIN.  THIS ENTRY IS PROVIDED   *
**TO ILLUSTRATE THE FORMAT OF EACH ENTRY.                        *
******************************************************************
**       DC CL8'CZN200  '    FONT CHARACTER SET NAME
**       DC XL1'00'          RESERVED, MUST BE ZERO
**       DC XL1'1F'          TECHNOLOGY ID
**       DC   H'2308'        FGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'1269'        GCSGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0000'        FONT WIDTH AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0000'        VERTICAL FONT SIZE AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
******************************************************************
**THE FOLLOWING 6 LINES MARK THE END OF APSRFNTG.                *
******************************************************************
         DC CL8'FFFFFFFF'    DO NOT REMOVE/CHANGE
         DC XL1'FF'          DO NOT REMOVE/CHANGE
         DC XL1'FF'          DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
***********************************************************************
* THE FOLLOWING IS THE CODE-PAGE-TO-GRID PARTS TABLE.                 *
***********************************************************************
         ENTRY APSRCPTG      DO NOT REMOVE/CHANGE
APSRCPTG DS    0D            DO NOT REMOVE/CHANGE
******************************************************************
**THE FOLLOWING 3 LINES ARE FOR EBCDIC INTERNATIONAL CODE PAGE   *
**THIS ENTRY IS PROVIDED TO ILLUSTRATE THE FORMAT OF ONE ENTRY.  *
******************************************************************
**       DC CL8'T1V10500'    CODE PAGE NAME
**       DC   H'0500'        CPGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
**       DC   H'0697'        CODE PAGE GCSGID AS DECIMAL NUMBER
**                            (OR USE " DC XL2'NNNN' " FOR HEX NUM)
******************************************************************
**THE FOLLOWING 3 LINES MARK THE END OF APSRCPTG.                *
******************************************************************
         DC CL8'FFFFFFFF'    DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
         DC XL2'FFFF'        DO NOT REMOVE/CHANGE
******************************************************************
**THE FOLLOWING LINE MARKS THE END OF APSRNTGM.                  *
******************************************************************
         END   APSRNTGM      DO NOT REMOVE/CHANGE
```

Figure 78 (Part 3 of 3). Font Names to Global Resource Identifiers (GRIDs)

Appendix J. Installing the i-data 7913 IPDS Printer LAN Attachment

This appendix describes how to install and configure an i-data 7913 IPDS Printer LAN Attachment.

Note: You must configure an i-data 7913 IPDS Printer LAN Attachment on a system—either AIX or OS/2—on which the **bootp** daemon program is running.

For a list of the printers that require the i-data 7913 IPDS Printer LAN Attachment, refer to *Advanced Function Presentation: Printer Information*, G544-3290.

One i-data 7913 IPDS Printer LAN Attachment is required for each printer you install, and each 7913 is installed and configured separately. The tasks of installation and configuration are as follows:

- Understanding how the i-data 7913 IPDS Printer LAN Attachment relates to PSF/MVS
- Checking prerequisites for the i-data 7913 IPDS Printer LAN Attachment
- Configuring the **bootp** daemon
- Setting up the **bootp** daemon on AIX or OS/2
- Configuring a twinax printer
- Setting the controls for the i-data 7913 IPDS Printer LAN Attachment
- Attaching and starting the i-data 7913 IPDS Printer LAN Attachment
- Testing the i-data 7913 IPDS Printer LAN Attachment
- Troubleshooting the i-data 7913 IPDS Printer LAN Attachment

For additional information, and for instructions for downloading corrective service 7913 microcode, refer to *i-data 7913 IPDS Printer LAN Attachment for Token Ring Installation Guide* or to *i-data 7913 IPDS Printer LAN Attachment for Ethernet Installation Guide*.

Understanding How the i-data 7913 IPDS Printer LAN Attachment Works

The i-data 7913 IPDS Printer LAN Attachment is a hardware protocol converter you can use to communicate between PSF/MVS and a coaxial or twinaxial IPDS printer.

The i-data 7913 IPDS Printer LAN Attachment requires TCP/IP communications protocols, and runs on either an IBM Token-Ring network or an Ethernet LAN. The i-data 7913 IPDS Printer LAN Attachment must be configured on a system, either AIX or OS/2, that is running the TCP/IP **bootp** daemon program. The **bootp** daemon controls the 7913 initialization process.

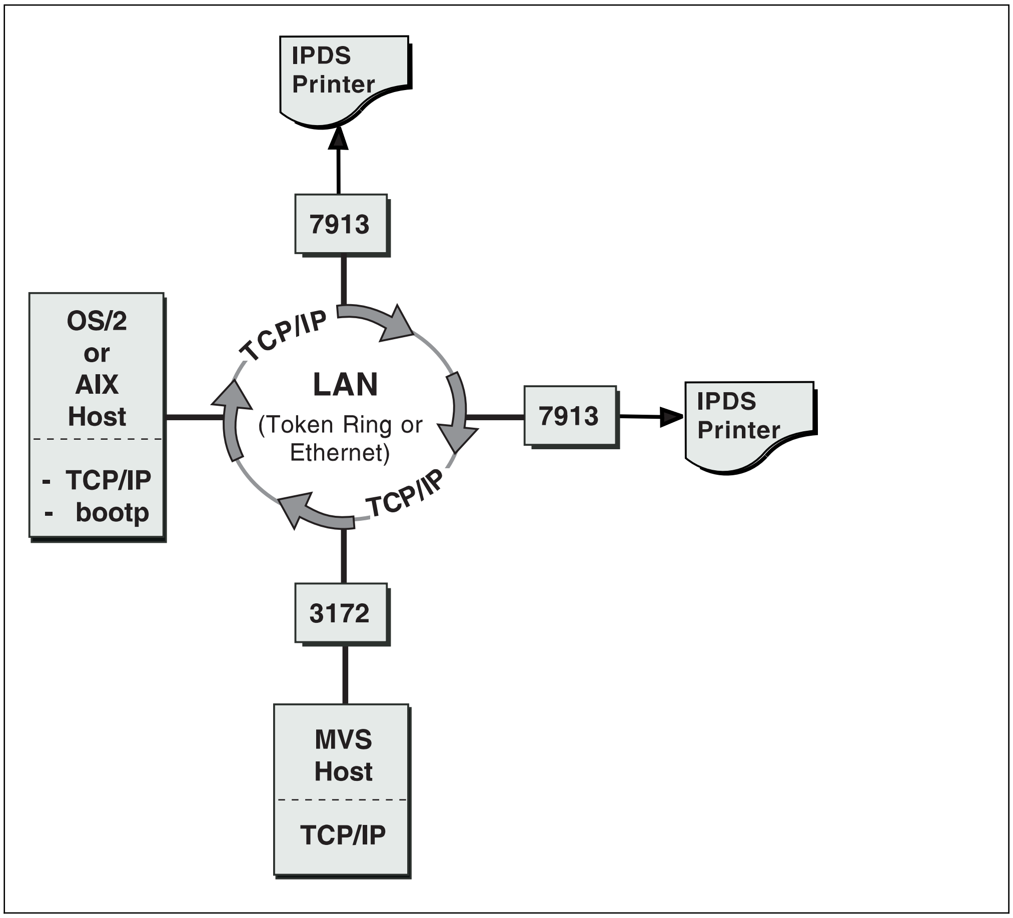


Figure 79. i-data 7913 IPDS Printer LAN Attachment in a TCP/IP LAN Environment

Figure 79 shows a sample TCP/IP LAN configuration, with an AIX or OS/2 host running TCP/IP and the TCP/IP **bootp** daemon. The MVS host and the AIX or OS/2 host can be attached directly to the same LAN as the i-data 7913 IPDS Printer LAN Attachment, as shown in Figure 79, or they can be attached to a different LAN connected through bridges, routers, and gateways.

Checking the i-data 7913 IPDS Printer LAN Attachment

Before you install the i-data 7913 IPDS Printer LAN Attachment, you must satisfy the following conditions:

__ • You must have one i-data 7913 IPDS Printer LAN Attachment for each printer you plan to install. You can order the i-data 7913 IPDS Printer LAN Attachment as a printer feature.

__ • If you are installing the i-data 7913 IPDS Printer LAN Attachment on an AIX system, you must have the following AIX hardware installed:

  – IBM AIX Version 3.2.5 for the RS/6000 (program number 5756-030) or higher. If the **bootp** and the i-data 7913 IPDS Printer LAN Attachment are on different token-ring LAN token-ring segments, PTF U422283 is required.

  – The TCP/IP (**bosnet.tcpip.obj**) option of the AIX operating system.

__ • If you are installing the i-data 7913 IPDS Printer LAN Attachment on an OS/2 system, you must have the following OS/2 software already installed:

  – OS/2 Version 2.11 or above

  – TCP/IP Version 2 or above

__ • You must have either an IBM Token-Ring network or an Ethernet LAN installed and operating correctly.

__ • Your AIX or OS/2 LAN must coexist with TCP/IP. If not, refer to *TCP/IP Tutorial and Technical Reference*.

__ • You must have obtained values for the following parameters from your network administrator for each i-data 7913 IPDS Printer LAN Attachment you plan to install:

  – Internet Protocol address
  – Subnet mask
  – Gateway address

Configuring the bootp Daemon

To configure the system, you need to configure the **bootp** daemon on the TCP/IP network. This section describes these tasks:

- Understanding how the **bootp** daemon works
- Completing the **bootp** configuration worksheet
- Editing the **bootptab** file

Understanding How the bootp Daemon Works

The TCP/IP **bootp** daemon on the server supports one or more i-data 7913 IPDS Printer LAN Attachments. When you start the 7913, it broadcasts an initialization request to all LAN systems. When the **bootp** daemon responds to this request, it accesses the **bootptab** file to send back configuration information and activate the 7913 as shown in Figure 80. The data flow is as follows:

1. On startup, the 7913 sends its universal MAC address, a unique address for each 7913, to the **bootp** daemon. The universal MAC address is printed on the side panel of each 7913 in the network.

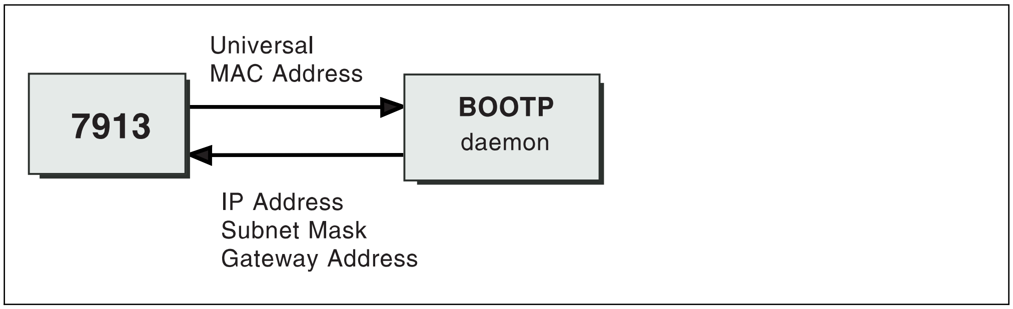


Figure 80. Data Flow between the i-data 7913 IPDS Printer LAN Attachment and the LAN

IP address
The Internet Protocol (IP) address—the full address of the 7913 as defined to the network. This address is used for configuration of the **bootp** daemon. You obtained this address earlier from your network administrator.

Note: The Internet Protocol address *must* be unique for each 7913 in your network.

Subnet mask
The part of the Internet Protocol address that represents the network identifier. You obtained this address earlier from your network administrator.

Gateway address
The address of the gateway system used for routing to Internet Protocol hosts on other networks. You obtained this address earlier from your network administrator.

For more information about **bootp** and related topics, refer to the following TCP/IP Requests for Comments (RFCs):

- RFC 951: The Bootstrap Protocol (**bootp**)
- RFC 1060: Assigned Numbers

Completing the bootp Configuration Worksheet

Before you modify the **bootptab**, complete a **bootp** configuration worksheet for each 7913 you are installing. Enter the values for each parameter name.

Note: To obtain the Universal MAC address, look at the side panel of the i-data 7913 IPDS Printer LAN Attachment. Obtain the remaining values from your TCP/IP network administrator.

Parameter name	Value
Ring speed (for IBM Token-Ring network model only)	
Universal MAC address	
IP address	
Subnet mask	
Gateway address	

Table 23. Configuration Worksheet for bootp. Complete a separate worksheet for each 7913 you attach.

Editing the bootptab File

The **bootptab** file contains a separate entry with configuration information for each 7913. You can use the same **bootptab** file in UNIX, AIX, and OS/2. In UNIX or AIX, the **bootptab** file is accessed under:

```
/etc
```

In OS/2, the **bootptab** file is in the following subdirectory:

```
\tcpip\etc
```

To configure the **bootptab** file for the 7913, you need to add the values from the worksheet (see Table 23).

Notes:

1. You must have a separate entry with a unique Internet Protocol (IP) address for each 7913.
2. On most AIX systems, you must have **root** user authority to update the **/etc/bootptab** file.

This section contains two examples of **bootptab** files:

- Example 1: A basic **bootp** configuration
- Example 2: A more complex **bootp** configuration

Example 1: A Basic bootp Configuration

This example shows how to configure two i-data 7913 IPDS Printer LAN Attachments with the same subnet mask and gateway address. The configuration is as follows:

A basic bootp configuration

```
P4028: ht=6: ha=00036EFFFFFF; ip=9.99.12.33:\
   sm=255.255-255.0: gw=9.99.12.254:

P3812: ht=6: ha=00036E00049F: ip=9.99.12.33:\
   sm=255.255.255.0: gw=9.99.12.254
```

The values used in Example 1 are as follows:

first field=host name
Identifies the entry for the 7913 in the **bootptab** file. For example, the name P4028 identifies an entry for a 4028 printer attached to the LAN by use of a 7913. The host name can be anything you want; it has no meaning to the system.

ht=host hardware type
Identifies whether the LAN is Token Ring or Ethernet. This entry must precede the host hardware address.

For an IBM Token-Ring network, enter **6**.
For an Ethernet LAN, enter **ethernet**.

ha=host hardware address
Identifies the universal MAC address for the 7913. Each 7913 has a unique MAC address specified in its hardware. The MAC address is printed on the side panel of the 7913.

ip=IP address
For a description of the Internet Protocol (IP) address, see page 331.

Note: The ID address *must* be unique for each 7913 in your network.

sm=subnet mask
For a description of the subnet mask, see page 332.

gw=gateway address
For a description of the gateway address, see page 332.

Example 2: A More Complex bootp Configuration

This example shows how to configure two 7913s in accordance with a more complex specification scheme. In this example, pointers (*tc*) are used to indicate an entry for the gateway and the subnet mask.

A More Complex bootp Configuration

```
# global entry with information that every host uses
global.lan:\
   sm=255.255.255.-:
# the subnet entry with the gateway address
subnetida.trlan:\
       tc=global.lan: gw=9.9912.254:
# entries for 7913 units
 P4028: tc=subnetida.trlan: ht=6: ha=00036EFFFFFF:\
        ip=subnetida.trlan: ht=6: ha=00036E00049F:\
 P3812: tc=subnetida.trlan: ht=6: ha=00036E00049F:\
        ip=9.99.12.33:
```

The values used in Example 2 are as follows:

first field=host name
Identifies the entry for the 7913 in the **bootptab** file. For example, the name P4028 identifies an entry for a 4028 printer attached to the LAN by use of a 7913. The host name can be anything you want; it has no meaning to the system.

tc=template host
Points to another entry that specifies addressing information. For exmaple, **tc=subnetida.trlan** points to the **subnetida.trlan** entry that contains the gateway address.

ht=host hardware type
Identifies whether the LAN is token-ring or Ethernet. This entry must precede the host hardware address.

For an IBM Token-Ring network, enter **6**.
For an Ethernet LAN, enter **ethernet**.

ha=host hardware address
Identifies the universal MAC address for the 7913. Each 7913 has a unique MAC address specified in its hardware. The MAC address is printed on the side panel of the i-data 7913 IPDS Printer LAN Attachment.

ip=IP address
For a description of the Internet Protocol (IP) address, see page 331.

Note: The IP address must be unique for each 7913 in your network.

sm=subnet mask
For a description of the subnet mask, see page 332.

gw=gateway address
For a description of the gateway address, see page 332.

Setting Up the bootp Daemon on AIX

After you have configured the **bootp** daemon, you need to set it up in the AIX environment. In order to do this, you must have **root** user authority.

Usually, the 7913 and the IP host running the **bootp** daemon belong to the same Internet Protocol subnetwork and also are on the same LAN segment; that is, they are not separated by a bridge. If you have token-ring LANs and bridges, and if your BOOTP daemon is running AIX 3.2.5 (or higher) with PTF U422283, the **bootp** daemon and the 7913 can be on different LAN segments.

The procedure for setting up the **bootp** daemon on AIX is as follows:

1. Add the following two lines to your **/etc/services** file, if they are not already there:

   ```
   bootps    67/udp   # bootp server port
   bootpc    68/udp   # bootp client port
   ```

   The # character indicates that the items to its right are a comment, and do not affect the **inetd** super-daemon.

2. Start the **bootp** daemon. There are two methods for doing this:

   - You can use the **inetd** super-daemon; this is the usual way to start and manage the **bootp** daemon.

     The **/etc/inetd.conf** file indicates which daemons are controlled by the **inetd** super-daemon. Use the AIX **more'.more** command to view the contents of the file. If the following line appears and is not preceded by a # character, the **bootp** daemon is controlled by the **inetd** super-daemon:

     ```
     bootps  dgram  udp  wait  root  /etc/bootpd bootpd
     ```

     If that line does not appear and you want the **bootpd** daemon to be managed by the **inetd** super-daemon, add that line to your **/etc/inetd.conf** file.

     If you have updated the **etc/inetd.conf** file, enter the following commands at the AIX command line prompt, so that the **inetd** super-daemon recognizes the updates:

     ```
     inetimp
     refresh -s inetd
     ```

   - You can start the **bootp** daemon manually, if it is not already being managed by the **inetd** daemon. If the **bootp** daemon is being managed by the **inetd** super-daemon, do not start the **bootp** daemon manually.

     To start the **bootp** daemon manually, enter the following at the AIX command line prompt:

     ```
     bootpd -s
     ```

     or

     ```
     bootpd -s -d -d -d -d -d
     ```

     where -s signifies that the **bootp** daemon is not controlled by the **inetd** super-daemon, and -d displays debugging information, such as the hardware address. Each additional -d increases the amount of debugging information that AIX displays. You can specify -d up to five times.

Setting Up the bootp Daemon on OS/2

To use the **bootp** daemon under OS/2, the SERVICES file in the ETC directory (`tcpip\etc\services`) must contain the following entries:

```
sbootp   67/udp    #bootp server
cbootp   68/udp    #bootp client
```

To test the **bootpd** command, enter the following from an OS/2 command line:

```
bootpd -d -d -d -d -d
```

where `-d` displays debugging information, such as the hardware address. Each additional `-d` increases the amount of debugging information displayed. You can specify `-d` up to five times.

When you are satisfied that the **bootpd** command is working correctly, you can add the command to your start procedure.

Add the **bootpc** command to the `tcpip\bin\tcpstart.cmd` file (assuming that TCP/IP was installed in the tcpip subdirectory). If the **bootp** daemon and the 7913 are on the same LAN segment (no bridges), enter the following:

```
start bootpd
```

If the **bootp** daemon and the 7913 are on the different LAN segments (bridges required), enter the following:

```
start bootpd -b
```

Configuring a Twinax Printer

For twinax printers, ensure that:

```
Station address = 0
```

Setting the Controls for the i-data 7913 IPDS Printer LAN Attachment

Two models of the i-data 7913 IPDS Printer LAN Attachment are available:

- Model 03 for the IBM Token-Ring network
- Model 04 for the Ethernet LAN

Before installing the i-data 7913 IPDS Printer LAN Attachment, set the controls on the side panel of each 7913 by following the appropriate procedure:

- Setting the controls for IBM Token-Ring network models
- Setting the controls for Ethernet LAN models

Note: Before you change the controls, make sure that the i-data 7913 IPDS Printer LAN Attachment is disconnected from the electrical outlet.

For the IBM Token-Ring Network Model

For the IBM Token-Ring LAN model of the i-data 7913 IPDS Printer LAN Attachment, set the RINGSPEED switch and the PRINTERTYPE switch as shown in Figure 81 on page 337.

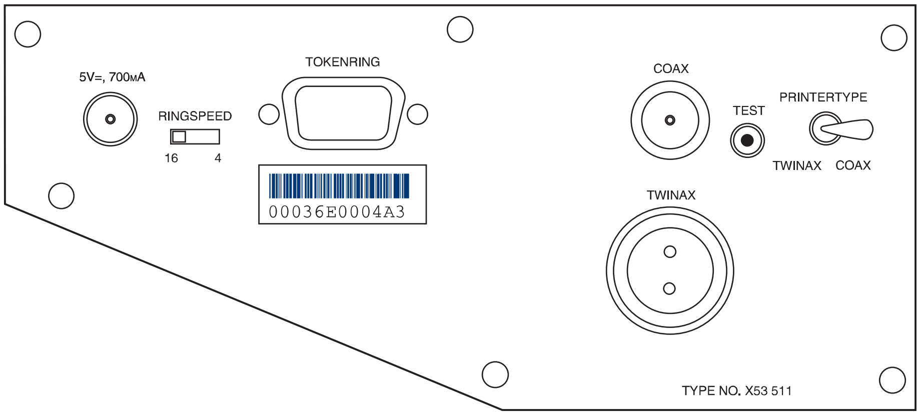


Figure 81. IBM Token-Ring Network Model of the i-data 7913 IPDS Printer LAN Attachment

To set the controls for the IBM Token-Ring network model of the i-data 7913 IPDS Printer LAN Attachment, follow these steps:

1. Before you change the controls, make sure that the i-data 7913 IPDS Printer LAN Attachment is disconnected from the electrical outlet.
2. Find out whether the LAN is 4 Mbps or 16 Mbps. You can get this information from your TCP/IP LAN network administrator.

   Note: If the RINGSPEED switch is set incorrectly, the LAN may malfunction.
3. Set the RINGSPEED switch to one of the following:

   4 For 4-Mbps LANs

   16 For 16-Mbps LANs
4. Set the PRINTERTYPE switch to:

   COAX (for an IBM coax printer)
   TWINAX (for an IBM twinax printer)

For the Ethernet LAN Model

For the Ethernet LAN model of the i-data 7913 IPDS Printer LAN Attachment, set the TWINAX/COAX rotary switch as shown in Figure 82.

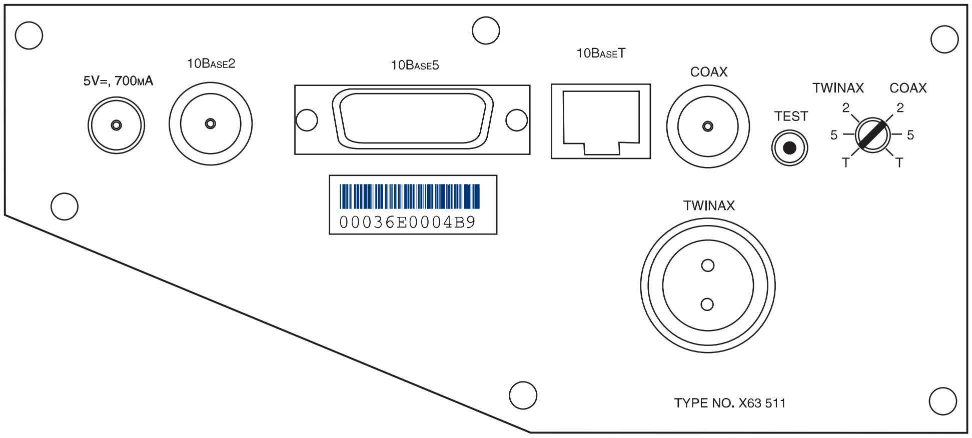


Figure 82. Ethernet LAN Model of the i-data 7913 IPDS Printer LAN Attachment

To set the control for the Ethernet model of the i-data 7913 IPDS Printer LAN Attachment, follow these steps:

1. Before you change the controls, make sure that the i-data 7913 IPDS Printer LAN Attachment is disconnected from the electrical outlet.
2. Set the TWINAX/COAX rotary switch to one of the following positions:

Coax 2	10base2 to coax
Coax 5	10base5 to coax
Coax T	10baseT to coax
Twinax 2	10base2 to twinax
Twinax 5	10base5 to twinax
Twinax T	10baseT to twinax

Operating the i-data 7913 IPDS Printer LAN Attachment

Before you start the i-data 7913 IPDS Printer LAN Attachment, you must attach the 7913 to the LAN and to the printer. This section describes the following tasks:

- Attaching the i-data 7913 IPDS Printer LAN Attachment
- Starting the i-data 7913 IPDS Printer LAN Attachment
- Shutting down the i-data 7913 IPDS Printer LAN Attachment

Attaching the i-data 7913 IPDS Printer LAN Attachment

To attach the i-data 7913 IPDS Printer LAN Attachment, cable it to the LAN and to the printer. Also install the ferrite cable block, which is shipped with the 7913. The ferrite cable block reduces the possibility of electrical interference.

Notes:

1. Cables to attach the 7913 to the LAN and to the printer are not provided with the i-data 7913 IPDS Printer LAN Attachment, PSF for AIX, or PSF for OS/2.
2. If you move a coaxial printer, move the ferrite cable block too.

To attach the i-data 7913 IPDS Printer LAN Attachment, follow these steps:

1. Attach the i-data 7913 IPDS Printer LAN Attachment to the LAN, using a LAN connection cable.
2. Attach the i-data 7913 IPDS Printer LAN Attachment to the printer, using the coax or twinax connection.
3. For coaxial printers, clamp the ferrite cable block to the coaxial cable, about 10 cm (4 inches) from the 7913.

Starting the i-data 7913 IPDS Printer LAN Attachment

To start the i-data 7913 IPDS Printer LAN Attachment, follow these steps:

1. Turn on the printer.
2. Plug the 7913 power cable into an electrical outlet. Because the 7913 is sensitive to power fluctuations, IBM recommends that you use an Uninterruptible Power Supply (UPS).

   The POWER light comes on, and several seconds later the LAN light blinks. This means that the 7913 is active on the IBM Token-Ring or Ethernet LAN, as in Figure 83 on page 339.

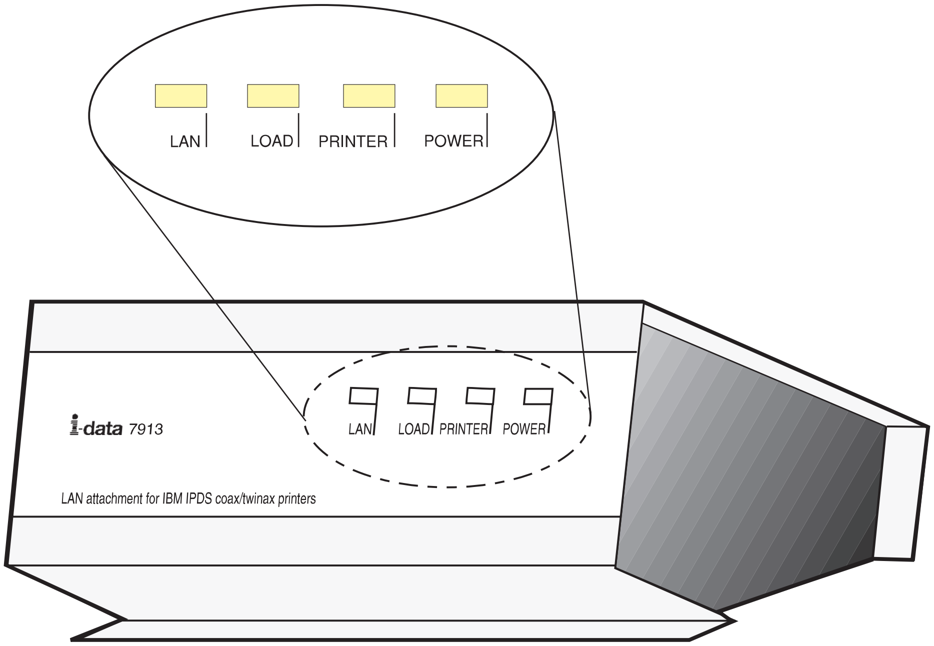


Figure 83. Status indicators on the i-data 7913 IPDS Printer LAN Attachment

Note: If the LAN light is not on, see "Troubleshooting the i-data 7913 IPDS Printer LAN Attachment" on page 342 about possible problems.

When the 7913 is powered on, the unit sends a **bootp** broadcast across the LAN with the MAC address. The **bootp** responds with:

- IP address
- Subnet mask
- Gateway address

If this information arrives successfully, the LAN light progresses from blinking status to a continuously lighted status.

For a more detailed explanation of the data flow that occurs when the i-data 7913 IPDS Printer LAN Attachment is set up, see "Understanding How the bootp Daemon Works" on page 331.

Shutting Down the i-data 7913 IPDS Printer LAN Attachment

You can shut down the i-data 7913 IPDS Printer LAN Attachment in several ways, some preferable to others.

- The preferred method is first to stop the PSF FSA for the printer from the MVS console by entering one of the following commands:
  - For JES2:

    ```
    $Pprinter-name
    ```

  - For JES3:

    ```
    *VARY printer-name,OFF
    *CANCEL printer-name
    ```

    where *printer-name* is the name of the PSF FSA for the printer.

    Then you can turn off power to the i-data 7913 IPDS Printer LAN Attachment and to the printer.

- Turn off power to the printer.

  The i-data 7913 IPDS Printer LAN Attachment notifies PSF that the printer is powered off, and the PSF FSA eventually ends normally.

- Turn off power to the i-data 7913 IPDS Printer LAN Attachment without turning off power to the printer.

  This procedure is not recommended, because PSF may not recognize for a long that the attachment has been powered off. The length of that time depends on the value specified for the keep-alive transmissions, described on page 238.

  To end the PSF FSA for the printer, use the JES commands shown in the first bullet. If you are unable to purge or cancel the printer by use of the JES commands, enter the following command:

  ```
  MODIFY FSSname,FORCE,printer-name
  ```

  Note: Before powering the i-data 7913 IPDS Printer LAN Attachment back on, be sure to power the printer off and then back on.

Testing the i-data 7913 IPDS Printer LAN Attachment

Before you test the installation of the i-data 7913 IPDS Printer LAN Attachment, make sure that the 7913 is plugged into an electrical outlet and attached to the LAN. These procedures are described in "Operating the i-data 7913 IPDS Printer LAN Attachment" on page 338.

To test the i-data 7913 IPDS Printer LAN Attachment, follow these steps:

1. Make sure that the POWER light is on and that the LAN light is constantly on (not blinking). This means that the 7913 is connected to the LAN.
2. Press the TEST button on the side panel of the 7913.

   A test page prints out on the printer. Figure 84 on page 341 shows a sample test page for the IBM Token-Ring network model; Figure 85 on page 341 shows a sample test page for an Ethernet LAN mode. If no text page prints, see "Troubleshooting the i-data 7913 IPDS Printer LAN Attachment" on page 342 for possible problems.

   Note: Pressing the TEST button breaks the connection between PSF/MVS and the 7913.

```
i-data 7913-03 IPDS Printer LAN Attachment.
Version S80 085.000*05 / 00307.004
PROM Version S80 086.075 / 00380.043
FLASH Version S80 086.075 / 00308.043 (Active)
(c) 1993 i-data international a-s DK, phonr: +45 444 007
Ring Speed: 16 Mbps
LAN Type: Token Ring 16 Mbps
Early Token Ring Release: Yes
Boot Status: Booted
Printer Attachment: Coax
MAC Address: 00:03:6E:00:04:A1 Universal
IP Address: 9.99.12.33
Netmask: FFFFFF00
Default Gateway: 9.99.12.254
Broadcast Address: 9.99.12.255
Port: 5001
Response Timeout: 20 seconds
IP Defaults: None
Remote IP: Port:
Remote IP: Port:
```

Figure 84. Sample Test Page for IBM 7913 Token-Ring Network Model

```
i-data 7913-03 IPDS Printer LAN Attachment.
Version S80 086.000*05 / 00308.004
PROM Version S80 086.075 / 00380.043
FLASH Version S80 086.075 / 00308.043 (Active)
(C) 1993 i=data international a-s DK, phone: +45 444 007
LAN Type: Ethernet 10Base2
Boot Status: Booted
Printer Attachment: Coax
MAC Address: 00:03:6E:00:04:BC Universal
IP Address: 9.99.12.33
Netmask: FFFFFF00
Default Gateway: 9.99.10.22
Broadcast Address: 9.99.10.255
Port: 5001
IP Defaults: 9.99.10.132,255.255.255.0, 9.99.10.220
Remote IP: Port:
Remote IP: Port:
```

Figure 85. Sample Test Page for Ethernet Model

3. Ping the 7913 from the PSF/MVS to verify the connectivity both of the physical LAN and of the IP.

   At the TSO command line, enter the **ping** command and the Internet Protocol address or host name of the 7913. For example:

   ```
   tso ping 9.99.12.33
   ```

   Or, from the SDSF menu 6:

   ```
   ping 9.99.12.33
   ```

   If the ping is successful, you receive a system response that looks like this:

```
EZA0458I Ping V3R1: Pinging host 192.35.12.128.
(Use ATTN to interrupt.)
EZA0463I PING: Ping #1 response took 0.084 seconds.
Successes so far=1.
```

If the ping is not successful, or if problems occur with another LAN device, see "Troubleshooting the i-data 7913 IPDS Printer LAN Attachment" for potential problems.

A successful ping usually indicates that the MVS system can communicate with the i-data 7913 IPDS Printer LAN Attachment; however, you might receive a successful ping even though the IP address of the i-data 7913 IPDS Printer LAN Attachment is a duplicate of another IP address. If PSF is unable to establish a network connection with the i-data 7913 IPDS Printer LAN Attachment, or if PSF output for the printer prints elsewhere, follow these steps to verify that the IP address of the i-data 7913 IPDS Printer LAN Attachment is unique:

1. Turn off the i-data 7913 IPDS Printer LAN Attachment.
2. Wait at least five minutes for TCP/IP to clear the Address Resolution Protocol (ARP) tables.[35]
3. Enter the **ping** command again from the MVS system to determine whether you receive a response.

If you receive a successful response to the **ping** command, indicating that there are duplicate IP addresses, consult your TCP/IP network administrator to resolve the IP address problem.

Troubleshooting the i-data 7913 IPDS Printer LAN Attachment

This section contains symptoms and solutions for common problems you may encounter when installing or operating an i-data 7913 IPDS Printer LAN Attachment. For additional problem-solving information you may need during installation, refer to the i-data publication that comes with the 7913 unit. For any further problem-solving information you need during operation, see the description of the PSF error message that is issued.

Table 24 on page 343 describes problems that might occur during the installation of the i-data 7913 IPDS Printer LAN Attachment.

Table 25 on page 344 describes problems that might occur during the operation of the i-data 7913 IPDS Printer LAN Attachment.

[35] You may need to wait longer if your installation specified a longer interval on the ARPAGE configuration statement in the TCP/IP profile. For information about the ARPAGE statement, refer to *TCP/IP for MVS: Customization and Administration Guide*.

Table 24. Problems in Installing the 7913 Attachment

Symptoms	Possible problems	Solutions
LAN light goes on briefly, then goes off. You have attached the 7913 to a LAN and turned it on.	Token-ring speed set incorrectly.	Set the ring speed to: **4** for 4-Mbps LANs **16** for 16-Mbps LANs To determine whether your LAN is a 4-Mbits LAN or a 16-Mbits LAN, see your network administrator. **Warning:** Turn off the 7913 before changing the ring speed.
The LAN light blinks continuously. You have attached the 7913 to a LAN and turned it on.	The **bootp** daemon is not started.	Start the **bootp** daemon on an OS/2 or AIX system.
	The 7913 and the BOOTP daemon are separated by a LAN bridge.	**OS/2:** Specify the **-b** option when starting the **bootp** daemon.
		AIX: Make sure that you have PTF U422283 installed on AIX 3.2.5.
	The 7913 and the **bootp** daemon are separated by an IP router.	Configure the IP router to pass certain broadcast frames.
	The hardware type in the **bootptab** entry is set incorrectly.	Set the hardware type to: **ht=6:** or **ht=tr:** for Token-Ring **ht=ethernet:** for Ethernet
	Cable attached incorrectly on the Ethernet LAN model.	Attach the 10Base2 cable to the 10Base2 connection, not to the COAX or TWINAX connection.
No test page prints. You have pressed the TEST button on the side of the 7913. The LAN and POWER lights are on. A printer is powered on, cabled to the 7913, and ready.	Coax/Twinax switch set incorrectly.	For the Ethernet LAN model, set the Coax/Twinax to: **Coax 2:** 10base2 to coax **Coax 5:** 10base5 to coax **Coax T:** 10baseT to coax **Twinax 2:** 10base2 to twinax **Twinax 5:** 10base5 to twinax **Twinax T:** 10baseT to twinax For the Token-Ring LAN model, set the PRINTERTYPE switch correctly, either to COAX or TWINAX. **Warning:** Turn off the 7913 before changing the switch.
No response on ping to the 7913.	The IP address of the MVS system is not unique.	Consult your network administrator to resolve the IP address problem.
The printer immediately displays an error when powered on. A twinax printer is cabled to the 7913, and the 7913's switches indicate twinax.	The twinax printer's station address is incorrect.	Set the twinax printer's station address to 0.

Table 25. Problems in Operating the 7913 Attachment

Symptoms	Possible problems	Solutions
The 7913 indicator lights indicate repeated power losses during operation. PSF repeatedly loses the connection to the 7913.	Power supply insufficient or overloaded.	Use an Uninterruptible Power Supply (UPS).
PSF fails to establish a connection to the 7913, or loses the connection to the 7913.	The 7913 is powered on, but the printer is powered off.	Turn the printer on and restart the PSF FSA for the printer.
	Duplicate address for the 7913.	By following the steps on page 342, verify that the address of the 7913 is unique.
	Duplicate address for the MVS system.	Consult your network administrator.
	The MTU size for the MVS system is too large.	For the recommended values of MTU. see “GATEWAY Statement” on page 238.

Appendix K. Related Publications

This appendix provides the titles of publications containing additional information about Advanced Function Presentation, the MVS operating system, PSF/MVS, and related products.

The titles and order numbers may change from time to time. To verify the current title or order number, consult your IBM marketing representative.

Print Services Facility/MVS (PSF/MVS)

Title	Order Number
IBM IP PrintWay Guide	S544-5379
IBM NetSpool Guide	G544-5301
PSF/MVS: Application Programming Guide	S544-3673
PSF/MVS: Diagnosis Guide and Reference	LH40-0205
Print Services Facility/MVS: Licensed Program Specifications	G544-3674
PSF/MVS: Messages and Codes	S544-3675
PSF/MVS: MVS Download Guide	G544-5294
Print Services Facility/MVS: Program Directory	None
Print Services Facility: Security Guide	S544-3291
Program Directory for IP PrintWay Feature of PSF/MVS	None
Program Directory for NetSpool Feature of PSF/MVS	None

Advanced Function Presentation (AFP)

Title	Order Number
Guide to Advanced Function Presentation	G544-3876
Mixed Object Document Content Architecture Reference	SC31-6802
Advanced Function Presentation: Programming Guide and Line Data Reference	S544-3884
Advanced Function Printing: Diagnosis Guide	LH40-0201
Advanced Function Printing: Host Font Data Stream Reference	S544-3289
Advanced Function Presentation: Printer Information	G544-3290
Advanced Function Presentation: Printer Summary	G544-3135
AFP Application Programming Interface: Programming Guide and Reference	S544-3872
AFP Application Programming Interface: COBOL Language Reference	S544-3873
AFP Application Programming Interface: PL/1 Language Reference	S544-3874
AFP Conversion and Indexing Facility: User's Guide	S544-5285
Printing and Publishing Cluster Collection CD/ROM	SK2T-2921

Title	Order Number
Page Printer Formatting Aid/370: User's Guide and Reference	S544-3700
Intelligent Printer Data Stream Reference	S544-3417

Advanced Function Presentation for Microfilm

To order any of the following microfilm publications, see your representative for Anacomp, Inc..

Title	Order Number
DataMASTER II for XFP2000-Linked Systems Operator Guide	DMIIR21
XFP2000 Bit-Mapped Imaging User's Guide	NW-04-9403
XFP2000 COM System Operator Guide	8000R21
XFP2000 Message Guide	8KMSGR11
XFP2000 Reference	XF-07-9201, XF-07-9302
XFP2000 Software Release Notes	XF-14-9301
XFP2000 User's Guide	XF-04-9302
XFP2000 Workstation Executive Installation and Operations Guide	XF-12-9201

MVS/Enterprise Systems Architecture (MVS/ESA) Version 4

Title	Order Number
MVS/ESA SP V4 Application Development Reference: Services for Authorized Assembler Language Programs, Volume 1 (ALESERV-DYNALLOC)	GC28-1647
MVS/ESA SP V4 Application Development Reference: Services for Authorized Assembler Language Programs, Volume 2 (ENFREQ-IXCTERM)	GC28-1648
MVS/ESA SP V4 Application Development Reference: Services for Authorized Assembler Language Programs, Volume 3 (LLACOPY-SDUMPX)	GC28-1649
MVS/ESA SP V4 Application Development Reference: Services for Authorized Assembler Language Programs, Volume 4 (SETFRR-WTOR)	GC28-1650
MVS/ESA SP V4 Authorized Assembler Programming Guide	GC28-1645
MVS/ESA SP V4 Configuration Program	GC28-1615
MVS/ESA SP V4 Diagnosis: Using Dumps and Traces	LY28-1813
MVS/ESA SP V4 Hardware Configuration Definition: User's Guide	GC33-6457
MVS/ESA SP V4 Initialization and Tuning Reference	GC28-1635
MVS/ESA SP V4 IPCS User's Guide	GC28-1631
MVS/ESA SP V4 JCL Reference	GC28-1654
MVS/ESA SP V4 JCL User's Guide	GC28-1653

Title	Order Number
MVS/ESA SP V4 JES2 Commands	GC23-0084
MVS/ESA SP V4 JES2 Initialization and Tuning Guide	SC23-0082
MVS/ESA SP V4 JES2 Initialization and Tuning Reference	SC23-0083
MVS/ESA SP V4 JES2 Messages	GC23-0085
MVS/ESA SP V4 JES3 Commands	GC23-0090
MVS/ESA SP V4 JES3 Initialization and Tuning Guide	SC23-0088
MVS/ESA SP V4 JES3 Initialization and Tuning Reference	SC23-0089
MVS/ESA SP V4 JES3 Messages	GC23-0091
MVS/ESA SP V4 Planning: B1 Security	GC28-1171
MVS/ESA SP V4 Planning: Dynamic I/O Configuration	GC28-1674
MVS/ESA SP V4 System Codes	GC28-1664
MVS/ESA SP V4 System Commands	GC28-1626
MVS/ESA SP V4 System Management Facilities	GC28-1628
MVS/ESA SP V4 System Messages, Volume 1	GC28-1656
MVS/ESA SP V4 System Messages, Volume 2	GC28-1657
MVS/ESA SP V4 System Messages, Volume 3	GC28-1658
MVS/ESA SP V4 System Messages, Volume 4	GC28-1659
MVS/ESA SP V4 System Messages, Volume 5	GC28-1660
MVS/ESA SP V4 Using the Functional Subsystem Interface	GC28-1638
RACF V2 General User's Guide	SC23-3728
RACF V2 Security Administrator's Guide	SC23-3726
RACF V2 System Programmer's Guide	SC23-3725

MVS/Enterprise Systems Architecture (MVS/ESA) Version 5

Title	Order Number
MVS/ESA SP V5 Authorized Assembler Services Guide	GC28-1467
MVS/ESA SP V5 Hardware Configuration Definition: User's Guide	SC33-6468
MVS/ESA SP V5 Planning: Operations	GC28-1441
MVS/ESA SP V5 Programming: Authorized Assembler Services Reference, Volume 1 (ALESERV-DNALLOC)	GC28-1475
MVS/ESA SP V5 Programming: Authorized Assembler Services Reference, Volume 2 (ENFREQ-ITTFMTB)	GC28-1476
MVS/ESA SP V5 Programming: Authorized Assembler Services Reference, Volume 3 (LLACOPY-SDUMPX)	GC28-1477
MVS/ESA Programming: Authorized Assembler Services Reference, Volume 4 (SETFRR-WTOR)	GC28-1478
MVS/ESA SP V5 Diagnosis: Tools and Service Aids	LY28-1845
MVS/ESA SP V5 Initialization and Tuning Reference	GC28-1452

Title	Order Number
MVS/ESA SP V5 IPCS User's Guide	GC28-1490
MVS/ESA SP V5 JCL Reference	GC28-1479
MVS/ESA SP V5 JCL User's Guide	GC28-1473
MVS/ESA SP V5 JES2 Commands	GC23-1443
MVS/ESA SP V5 JES2 Initialization and Tuning Guide	SC23-1453
MVS/ESA SP V5 JES2 Initialization and Tuning Reference	SC23-1454
MVS/ESA SP V5 JES2 Messages	GC28-1488
MVS/ESA SP V5 JES3 Commands	GC23-1444
MVS/ESA SP V5 JES3 Initialization and Tuning Guide	SC23-1455
MVS/ESA SP V5 JES3 Initialization and Tuning Reference	SC23-1456
MVS/ESA SP V5 JES3 Messages	GC28-1489
MVS/ESA SP V5 Planning: B1 Security	GC28-1440
MVS/ESA SP V5 System Codes	GC28-1486
MVS/ESA SP V5 System Commands	GC28-1442
MVS/ESA SP V5 System Management Facilities	GC28-1457
MVS/ESA SP V5 System Messages, Volume 1	GC28-1480
MVS/ESA SP V5 System Messages, Volume 2	GC28-1481
MVS/ESA SP V5 System Messages, Volume 3	GC28-1482
MVS/ESA SP V5 System Messages, Volume 4	GC28-1483
MVS/ESA SP V5 System Messages, Volume 5	GC28-1484
MVS/ESA SP V5 Using the Functional Subsystem Interface	SC28-1505
RACF V2 General User's Guide	SC23-3728
RACF V2 Security Administrator's Guide	SC23-3726
RACF V2 System Programmer's Guide	SC23-3725
System Display and Search Facility Guide and Reference	SC23-0408

| ## OS/390 Release 3

Title	Order Number
HCD User's Guide	SC28-1848
OS/390 HCD Planning	GC28-1750
OS/390 JES2 Commands	GC28-1790
OS/390 JES2 Initialization and Tuning Guide	SC28-1791
OS/390 JES2 Initialization and Tuning Reference	SC28-1792
OS/390 JES2 Messages	GC28-1796
OS/390 JES3 Commands	GC28-1798
OS/390 JES3 Initialization and Tuning Guide	SC28-1802
OS/390 JES3 Initialization and Tuning Reference	SC28-1803

Title	Order Number
OS/390 JES3 Messages	GC28-1804
OS/390 MVS Auth Assembler Services Guide	GC28-1763
OS/390 MVS Auth Assembler Services Reference ALE-DYN	GC28-1764
OS/390 MVS Auth Assembler Services Reference ENF-ITT	GC28-1765
OS/390 MVS Auth Assembler Services Reference LLA-SDU	GC28-1766
OS/390 MVS Auth Assembler Services Reference SET-WTO	GC28-1767
OS/390 MVS Diagnosis: Tools and Service Aids	LY28-1845
OS/390 MVS Initialization and Tuning Reference	SC28-1752
OS/390 MVS IPCS User's Guide	GC28-1756
OS/390 MVS JCL Reference	GC28-1757
OS/390 MVS JCL User's Guide	GC28-1758
OS/390 MVS System Codes	GC28-1780
OS/390 MVS System Commands	GC28-1781
OS/390 MVS System Management Facilities (SMF)	GC28-1783
OS/390 MVS System Messages, Vol 1 (ABA-ASA)	GC28-1784
OS/390 MVS System Messages, Vol 2 (ASB-EWX)	GC28-1785
OS/390 MVS System Messages, Vol 3 (GDE-IEB)	GC28-1786
OS/390 MVS System Messages, Vol 4 (IEC-IFD)	GC28-1787
OS/390 MVS System Messages, Vol 5 (IGD-IZP)	GC28-1788
OS/390 MVS Using the Functional Subsystem Interface	SC28-1911
OS/390 Security Server (RACF) General User's Guide	SC28-1917
OS/390 Security Server (RACF) Security Administrator's Guide	SC28-1915
OS/390 Security Server (RACF) System Programmer's Guide	SC28-1913
OS/390 V1R3.0 SDSF Guide and Reference	SC28-1622

VTAM and NCP

Title	Order Number
VTAM Customization	LY43-0075
VTAM Resource Definition Reference	SC31-8377
Network Control Program: System Support Program Emulation Program Resource Definition Guide	SC30-3447
Network Control Program Resource Definition Reference	SC30-3448
ACF/NCP/SSP Version 3 Resource Definition Reference	SC30-3254

Fonts

Title	Order Number
ABOUT TYPE: IBM's Technical Reference for 240-pel Digitized Type	S544-3516
IBM AFP Fonts: Font Samples	S544-3792
IBM AFP Fonts: Font Summary	G544-3810
IBM AFP Fonts: IBM's Typographic Primer for Digitized Type	G544-3183
IBM AFP Fonts: Introduction to Typography	G544-3122
IBM AFP Fonts: Technical Reference for Code Pages	S544-3802
IBM AFP Fonts: Technical Reference for IBM Chinese, Japanese, and Korean Fonts	S544-5330
IBM AFP Fonts: Technical Reference for IBM Expanded Core Fonts	S544-5228
IBM AFP Fonts: Type Transformer User's Guide	G544-3796

Print Services Facility for OS/2 (PSF/2)

Title	Order Number
IBM Print Services Facility for OS/2: Network Configuration Guide for System/370 and Communication Manager/2	S544-3911
IBM Print Services Facility for OS/2: An Installation Cookbook for System/370 and Token Ring Networks	G544-3965
IBM Print Services Facility for OS/2: An Installation Cookbook for AS/400 and Token Ring Networks	G544-3966
IBM Print Services Facility for OS/2: Printer Attachments Guide	G544-5215

Print Services Facility for AIX (PSF for AIX)

Title	Order Number
IBM Page Printer Formatting Aid/6000: User's Guide Version 2.1	*S544-3918*
IBM Print Services Facility for AIX: Print Services Facility for AIX Users	*G544-3814*
IBM Print Services Facility for AIX: AIX for Print Services Facility Users	*G544-3766*
IBM Print Services Facility for AIX: Print Administration	*S544-3817*
IBM Print Services Facility for AIX: Print Submission	*S544-3878*
IBM Print Services Facility for AIX: AFP Conversion and Indexing Facility Version 2.1	*G544-3930*

Printers

For a more complete listing of printer publications, refer to *Advanced Function Presentation: Printer Information*, G544-3290.

Title	Order Number
Attachment Configuration Handbook for Printers That Use the Advanced Function Common Control Unit	S544-3977
IOCP User's Guide and ESCON Channel-to-Channel Reference	GC38-0401
Reference Manual for the IBM 3800 Printing Subsystem Models 3 and 6	GA32-0050
IBM PagePrinter 3812 Introduction and Planning Guide	G544-3265
IBM 3816 Page Printer Operating Instructions	GA34-2075
IBM 3820 Page Printer Introduction and Planning Guide	G544-3520
3825 Page Printer Product Description	G544-3482
3827 Page Printer Product Description	G544-3194
IBM 3828 Advanced Function MICR Printer Product Description	G544-3361
IBM 3829 Advanced Function Printer Product Description	GA32-0255
3835 Page Printer Product Description	G544-3498
3835 Advanced Function Printer Model 002 Product Description	GA32-0249
IBM 3900 Advanced Function Printer Product Description	GA32-0135
IBM 3912 and 3916 Page Printer Getting Started	S544-3898
IBM 3930 Page Printer Models 02D, 02S, 03D, and 03S Introduction and Planning Guide	GA24-4375
IBM 3935 Advanced Function Printer Production Description	G544-3894
IBM LaserPrinter 4028 Introduction and Planning Guide	S544-4258
User's Guide for the IBM LaserPrinter 4029 Series	ZA40-0542
IBM 4224 Printer Models 1xx and 2xx Product and Programming Description Manual	GC31-2551
IBM 4230 Printer Product and Programming Description Models 102 and 202	GC40-1701
IBM 4234 Printer Models 007, 008, 011, and 012 Product and Programming Description	GC31-3879

TCP/IP

Title	Order Number
Internetworking with TCP/IP, Principles, Protocols, and Architecture	SC31-6144
TCP/IP Tutorial and Technical Reference	GG24-3376
TCP/IP for MVS: Application Programming Interface Reference	SC31-7187
TCP/IP for MVS: Customization and Administration Guide	SC31-7134
TCP/IP: Performance Tuning Guide	SC31-7188
TCP/IP for MVS: Programmer's Reference	SC31-7135

i-data

Title
i-data 7913 IPDS Printer LAN Attachment for Ethernet Installation Guide
i-data 7913 IPDS Printer LAN Attachment for Token Ring Installation Guide

Glossary

Sources

This publication includes terms and definitions from the *IBM Dictionary of Computing,* ZC20-1699.

Definitions reprinted from the *American National Dictionary for Information Processing Systems* are identified by the symbol (A) following the definition.

Definitions reprinted from a published section of the International Organization for Standardization (ISO) *Vocabulary—Information Processing* or from a published section of *Vocabulary—Office Machines* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1) are identified by the symbol (I) following the definition. Because many ISO definitions are also reproduced in the *American National Dictionary for Information Processing Systems,* ISO definitions may also be identified by the symbol (A).

Definitions reprinted from working documents, draft proposals, or draft international standards of ISO Technical Committee 97, Subcommittee 1 (Vocabulary), Joint Technical Committee 1 are identified by the symbol (T) following the definition, indicating that final agreement has not yet been reached among its participating members.

Definitions that are specific to IBM products are so labeled—for example, "In SNA," or "In the 3820 printer."

References

The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposite or substantively different meaning.

See. This refers to multiple-word terms in which this term appears.

See also. This refers to related terms that have similar, but not synonymous, meanings.

Synonym for. This appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

Synonymous with. This appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

A

abend. An abnormal end of task before completion of the task because of an error condition that cannot be recovered while the task is running.

absolute coordinate. One of the coordinates that identify the location of an addressable point with respect to the origin of a specified coordinate system. Contrast with *relative coordinate*.

absolute positioning. The establishment of a position within a coordinate system as an offset from the origin of the coordinate system. Contrast with *relative positioning*.

access control. In computer security, the methods and facilities used to ensure that a computer system and the data, system software, and application programs stored in it can be accessed only by authorized users in authorized ways.

access method. A method for moving data between main storage and I/O devices.

accumulator. (1) A 3800 hardware feature that supplies a separate storage area to hold data in raster form. It can be used either for composing a sheet of data that combines a large amount of variable and constant data, or for storing an electronic overlay in raster form so that the overlay is merged with variable data as the page is printed. (2) A register in which the result of an operation is formed.

ACIF. (1) AFP conversion and indexing facility. (2) A PSF utility program that converts a print file into MO:DCA (AFP), creates an index file for input data, and collects resources used by an AFP document into a separate file.

ACK. Acknowledgment. See also *Positive acknowledge reply.*

acknowledge reply. A printer-to-host reply that returns printer information (a positive reply) or reports exceptions (a negative reply).

acknowledgment request. A request from the host for information from the printer. An example of an acknowledgment request is the use of the ARQ flag by a host system to request an acknowledge reply from an attached printer.

action code. A software-generated or hardware-generated code that indicates a recovery action. The hardware action code is byte 2 of the sense data.

active environment group. A collection of mapping structured fields, positioning controls, and data descriptors that define the environment for a page. These structured fields form an internal object in MO:DCA data, page definition, or overlay.

addressable point. For page printers, any point in a presentation surface that can be addressed. Synonymous with *picture element*.

Advanced function common control unit (AFCCU). Converts the IPDS into a presentation format that is usable by the COM; transfers the setup data, document pages, and text-related information to COM.

Advanced Function Presentation (AFP). A set of licensed programs, together with user applications, that use the all-points-addressable concept to print on presentation devices. AFP includes creating, formatting, archiving, retrieving, viewing, distributing, and printing information. See *presentation device*.

AEA. Alternative exception action.

AFCCU. Advanced function common control unit.

AFP. Advanced Function Presentation.

AFP data stream. A presentation data stream that is processed in AFP environments. MO:DCA-P is the strategic AFP data stream. IPDS is the strategic AFP printer data stream.

AFPDS. A term formerly used to identify the MO:DCA-P-based data stream interchanged in AFP environments. See also *MO:DCA-P* and *AFP data stream*.

all-points-addressable (APA). The ability to address, reference, and position text, overlays, and images at any defined position or pel on the printable area of the paper. This capability depends on the ability of the hardware to address and to display each picture element.

| **AIX**. Advanced Interactive Executive

| **AIX operating system**. IBM's implementation of the
| UNIX operating system. RS/6000 runs the AIX
| operating system.

AOS. Alternative offset stacker.

APA. All-points-addressable.

APAR. Authorized program analysis report.

API. Application program interface

application program. (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll processing.

AR. Activate resource (IPDS command).

ARP. Address resolution protocol.

ARQ. (1) Acknowledgment required. (2) A flag in the header of an IPDS command, requesting the subsystem to inform PSF when that command and all previous commands have been processed (a way of synchronizing PSF with subsystems). The NOP command following an End Object Container and Print Buffered Data command sequence is sent with an ARQ so that the COM can check the setup data and determine whether a resolution change is necessary.

asynchronous. Without regular time relationship; unexpected or unpredictable with respect to running program instructions.

authorized program analysis report (APAR). A report of a problem caused by a suspected defect in a current unaltered release of a program.

auxiliary data set. A data set that contains job header, data set header, job trailer, or message data. Contrast with *print data set*.

auxiliary resource. Fonts, page segments, overlays, page definitions, or form definitions associated with auxiliary data sets.

B

baseline. In a font, the imaginary line on which successive characters are aligned in the inline direction.

BCOCA. Bar Code Object Content Architecture.

bin. A paper supply on cut-sheet printers. See also *cassette*.

BIND. The command sent by VTAM to a logical unit to establish a VTAM program-to-logical unit session.

bitmap. A coded representation in which each bit, or group of bits, represents or corresponds to an item:umdashfor example, a configuration of bits in main storage in which each group of bits corresponds to one pel of a display image.

BOC. Begin Object Container. Marks the beginning of the object.

boiler plate. Presentation data that is replicated on a page or within a group of pages, serving as instructions for (or auxiliary information about) the page or group of pages—for example, instructions on how to fill out a form.

bounded-box font. A font designed to use bounded character boxes. Contrast with *unbounded-box font*.

bounded-box relative metrics. Units of measure that are expressed in 1 000 units per Em-square.

bounded character box. A character-box design that does not require character positioning information to be included (as untoned pels) in the box. Contrast with *unbounded character box*.

BTS. Burster-trimmer-stacker.

buffer pool. An area of storage in which all the buffers of a program are kept.

buffered pages. Pages kept in printer control storage, waiting to be printed.

burst. To separate continuous-forms paper into single sheets.

burster-trimmer-stacker (BTS). A printer hardware feature that separates continuous forms into single sheets, trims the carrier strip from both edges of the forms, and stacks the sheets.

byte multiplexer channel. A multiplexer channel that interleaves bytes of data.

C

cache. To store in a special-purpose buffer storage, which is smaller and faster than main storage. PSF holds a copy of data that was obtained from main storage and that will be needed next for processing.

| **capture**. Some printers can capture downloaded fonts. These saved fonts automatically become new temporary printer-resident fonts. This improves performance on future jobs that use the same fonts. Both raster fonts and outline fonts can be captured.

carriage control character. An optional character in an input data record that specifies a write, space, or skip operation.

cassette. In a cut-sheet printer, a movable paper storage enclosure. See also *bin*.

CCGSID. Coded character set identifier.

CCM. Common Conversion Machine.

CCW. Channel command word.

CDS. Control data set.

CF. Coded font.

channel-attached. In PSF, a device that is linked to the host system exclusively by System/390 channel protocols. For example, a 3800 printer cabled to the host system with an IBM System/390 channel adapter is considered a channel-attached printer. Contrast with *SNA-attached*.

channel code. A number from 1 to 12 that identifies a position in the forms control buffer or a page definition. A carriage control character can select a position defined by a particular channel code.

channel command word (CCW). A command that directs data channel operations. CCWs built by Print Services Facility are sent to channel-attached printers.

channel counter. A counting device that identifies how many pages have been successfully received. Synonymous with *received page counter*.

channel station. In the printer, a page-counting device that counts the number of pages queued for printing.

CHAR. A data type for architecture syntax, which indicates one or more bytes to be interpreted as character information.

character. (1) A member of a set of elements that is used for the representation, organization, or control of data. Characters may be letters, digits, punctuation marks, or other symbols represented in the form of a spatial arrangement of adjacent or connected strokes or in the form of other physical conditions in data media. (2) A letter, numeral, punctuation mark, or special graphic used for the production of text. (3) A byte of data. (4) See also *graphic character*.

character box. The area that completely contains the graphic character raster pattern.

character data. Data in the form of letters and special characters, such as punctuation marks. See also *numeric data*.

character graphic. A visual representation of a character, other than a control character, that is normally produced by writing, printing, or displaying. (T)

character group. Any number of character graphics and character properties.

character identifier. The standard identifier for a character regardless of its style. For example, all uppercase A's have the same character identifier. Synonymous with *graphic character identifier*.

character increment. The distance between the current print position and the next print position.

character rotation. The alignment of a character with respect to the baseline, measured in degrees in a clockwise direction. Examples are 0°, 90°, 180°, and 270°. This term usually applies to individual characters. Synonym for *rotation*. See also *orientation*.

character set. (1) A finite set of characters upon which agreement has been reached and which is considered complete for some purpose—for example, each of the characters in ISO Recommendation R646 "6- and 7-bit coded character sets for information processing interchange." (2) For page printers, the font library member that contains the character graphics and their descriptions.

checkpoint. A point at which information about the status of a job and the system can be recorded so that the job step can be restarted later. For any print job, the user can specify that PSF/MVS take checkpoints based on either the number of logical pages processed or the number of processing seconds elapsed.

code page. (1) A set of assignments, each of which assigns a code point to a character. Each code page has a unique name or identifier. Within a given code page, a code point is assigned to one character. More than one character set can be assigned code points from the same code page. See also *coded font*. (2) A font component that associates code points and character identifiers. A code page also identifies how undefined code points are handled.

Code Page Global Identifier (CPGID). A unique code page identifier that can be expressed either as a 2-byte binary value or as a 5-digit decimal value.

code point. A 1-byte code representing one of 256 potential characters.

coded font. A font library member that associates a code page and a font character set. For double-byte fonts, a coded font associates more than one pair of code pages and font character sets.

coded font local identifier. A 1-byte identifier that the Map Coded Font structured field assigns to each coded font it selects. The identifier is then specified in the text-control sequence that precedes the string of text to be printed with the particular font.

coded font section. A font character set and code page pair. A single-byte coded font consists of only one coded font section; a double-byte coded font can consist of more than one.

coded overlay. An overlay that is stored in the printer in a coded (not raster) format. Contrast with *raster pattern overlay*.

color selection. The ability to specify a color other than black to print data in more than one color. Some printers support selection of several colors, depending upon the color of the ribbon installed in the printer. Other printers support the selection of black or "color of media" that can cause white lettering on a background that has been shaded black, for example.

COM. Computer output microfilm. COM is the hardware controller that manages the microfilm print engine and processes the functions unique to COM.

COM device. See *microfilm device*.

COM setup data. Data that enables the PSF user to designate unique microfilm printing functions for AFP print jobs. The parameter values, structure, syntax, and semantics are defined by the COM manufacturer, not by IBM-controlled architectures.

COM Setup Utility. An MVS-based utility that creates an AFP object containing the COM setup data for a microfilm device.

command. A request from a terminal or a specification in a batch-processing print file for the performance of an operation or the execution of a particular program.

committed copy counter. A counting device that identifies the number of copies of a page that have been committed to printing but have not been counted by the committed page counter.

committed page counter. A counting device that identifies the number of pages that have been committed to printing and have been removed from the page buffer. Synonym for *transfer counter*.

common-use forms. A set of paper sizes selected as being the most common in use throughout the world.

communication. See *data communication*.

communication-attached. In PSF, pertaining to a device that is SNA-attached and that uses a communication controller. For example, a 3812 printer attached to a 3174 control unit that is attached to a 37X5 Communication Controller is considered a communication-attached printer. Contrast with *local-attached*.

compatibility font. An AFP font designed to emulate the uniformly spaced and fixed-pitch fonts used with line printers. Synonym for *IBM Compatibility font*.

compatibility mode. A mode of operation for printing the output of 3800 Model 1 application programs on a 3800 page printer with little or no change to the application or the JCL.

completion code. An indication that reflects the status of a task set at the time of its termination.

composed-text data. An obsolete term for text data and text control information that dictates the format, placement, and appearance of data to be printed. Currently called *MO:DCA-P data*.

compressed pattern storage (CPS). Storage that holds the extended (double-byte) fonts for the 3800 Model 6 or Model 8 printers.

Computer output microfilm. See *COM*.

computing system RPQ. A customer request for a price quotation on alterations or additions to the functional capabilities of a computing system, hardware product, or device. The RPQ can be used in conjunction with programming RPQs to solve unique problems in data processing. See also *programming RPQ* and *RPQ*.

concatenate. (1) To link together. (2) To join two character strings.

concatenated data set. In MVS, a group of logically connected data sets that are treated as one data set for the duration of a job step. See also *data set*, *partitioned data set*, and *library*.

conditional processing. A page definition function that enables input data records to partially control their own formatting.

condition code. A code that reflects the result of an input/output, arithmetic, or logical operation.

console. A part of a computer used for communication between the operator or maintenance engineer and the computer. (A)

continuous forms. A series of connected forms that feed continuously through a printing device. The connection between the forms is perforated so that the user can separate them. Before printing, the forms are folded in a stack arrangement, with the folds along the perforations. Contrast with *cut-sheet paper*.

continuous-forms stacker. An output assembly on a continuous-forms printer that refolds and stacks the continuous forms after printing.

control character. A character that starts, changes, or stops any operation that affects the recording, processing, transmitting, or interpreting data (such as carriage return, font change, and end of transmission).

control record. A subset of structured fields that can be intermixed with line data records in a print data set.

copy group. (1) One or more copies of a page of paper. Each copy within a group can have modifications, such as text suppression, page position, forms flash, and overlays. (2) Synonym for *medium map*.

copy modification. The process of adding, deleting, or replacing data on selected copies of certain pages of a print job.

copy separation. The method for distinguishing between consecutive copies of a single data set. In an continuous-forms stacker, the method consists of the alternation between one, two, or three vertical bars placed on the left carrier strip at the end of a copy. In the burster-trimmer-stacker, or on cut-sheet paper, copies are separated by offset stacking.

core interchange font. A uniformly spaced typographic font with specialized characters for different languages.

CPGID. See *Code Page Global Identifier.*

CPS. Compressed pattern storage.

current print position. The picture element that defines the character reference point or the upper-left corner of an image.

cut-sheet paper. Paper that is cut into separate, uniform-size sheets before being loaded into the printer. Contrast with *continuous forms*.

D

daemon. In the AIX and OS/2 operating systems, a process begun by the root user or by the root shell that can be stopped only by the root user. Daemon processes generally provide services that must be available at all times, such as sending data to the printer. A daemon runs continuously, looking for work to do, performing that work, and waiting for more work. A daemon does not have a controlling terminal associated with it.

DASD. Direct access storage device.

data check. A synchronous indication of a condition caused by invalid data or incorrect positioning of data. The reporting of some data checks can be suppressed.

data communication. Transfer of information between functional units by means of data transmission according to a protocol. (I)

data control block (DCB). A control block used by access method routines in storing and retrieving data.

data link. Connection between two terminals operating according to a link protocol. Contrast with *telecommunication line*.

data map. An internal object in a page definition that specifies fonts, page segments, fixed text, page size, and the placement and orientation of text. Synonym for *page format*.

data map transmission subcase. An internal object that specifies the control information for printing line data. One data-map transmission subcase must appear in each data map of a page definition.

data set. (1) Synonym for *file*. (2) A named set of records stored and processed as a unit. See also *concatenated data set*, *partitioned data set*, and *sequential data set*.

data set header. A page in printed output that separates multiple data sets or multiple copies of a data set within a print job. See also *job header*.

DCB. Data control block.

DCF. Document Composition Facility.

default. Pertaining to an attribute, value, or option that is assumed when none is explicitly specified. (I)

deferred-printing mode. A printing mode that spools output through JES to a data set instead of printing it immediately. Output is controlled using JCL statements. Contrast with *direct-printing mode*.

define group boundaries. An IPDS command used to assign a common set of processing functions (such as “Keep as one entity”) and/or characteristics (such as job source identification) to a group of pages, one document, or several documents.

device manager. (1) For channel-attached printers, the subcomponent of PSF that manages the interface to the printer, builds CCWs, issues I/O requests, and processes I/O interrupts. (2) For SNA-attached printers, the device manager is the PSF subcomponent that loads the PPCC during device initialization, processes completed I/O, establishes and controls an SNA session through the PPCC, and supplies printer information in the CCB. (3) For TCP/IP-attached printers, the PSF subcomponent that processes completed I/O, establishes and controls a network session through the TCP/IP subcomponent, and supplies printer information in the CCB.

DGB. See *define group boundaries*.

direct access storage device (DASD). A computer storage device in which access time is, in effect, independent of the location of the data.

Direct Printer Services Subsystem (DPSS). The PSF subcomponent that acts as the interface between PSF and an application program when the job entry subsystem is not spooling jobs for a printer. DPSS attaches PSF as a subtask for use in the direct-printing mode.

direct-printing mode. A printing mode that allows PSF exclusive use of a channel-attached printer. Output is printed immediately and is not spooled through JES. Contrast with *deferred-printing mode*.

disabled mechanism. A function of a printer that is temporarily out of operation. In such a case, PSF may use alternative options.

Distributed Print Function (DPF). A component of PSF/2 that you can install and use to print jobs from MVS, VSE, VM, or OS/400 systems. DPF receives PSF output and resources for spooling and printing with PSF/2. DPF allows PSF/MVS, PSF/VSE, PSF/VM, and OS/400 to send print files to a PS/2. DPF also stores PSF/MVS and PSF/VSE resources in the DPF resource library, so that the host system does not have to send PSF resources each time documents are spooled.

document. (1) One or more pages collected into a single job. (2) Data that has already been composed into pages and that contains a Begin Document and an End Document structured field. (3) In word processing, a collection of one or more lines of text that can be named and stored as a separate entity. (4) A publication or other written material pertaining to a specific subject or related subjects.

Document Composition Facility (DCF). An IBM licensed program that provides a text formatter called SCRIPT/VS. SCRIPT/VS can process files marked up with a unique set of controls and tags.

document environment group. An internal object that identifies overlay and suppression usage and defines the placement of the page or pages on the form. A document environment group is a required part of every form definition.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

double-byte coded font. A font in which the characters are defined by 2 bytes, the first defining a coded font section, and the second defining a code point. Double-byte coded fonts are required to support languages requiring more than 256 graphic characters. Two bytes are required to identify each graphic character. A double-byte font is used to print Kanji. Contrast with *single-byte coded font*.

double-dot image. In Advanced Function Presentation, an image that is enlarged by doubling the pel pattern horizontally and vertically.

download. To transfer data from a processing unit to an attached device such as a microcomputer for processing.

DPF. Distributed Print Function.

DPSS. Direct Printer Services Subsystem.

drain. An operator action to halt the flow of jobs to a printer, usually to stop the printer or to change print options.

drained state. The condition in which job flow to the printer is stopped until the operator enters a command, such as START, that instructs PSF to begin sending jobs to the printer.

DRM. Dynamic reconfiguration management.

DSC. Data synchronous command

duplex printing. Printing on both sides of a sheet of paper. Contrast with *simplex printing*. See also *normal duplex printing* and *tumble duplex printing*.

dynamic reconfiguration management (DRM). A capability that enables a system programmer to add, delete, or change AFP1 or 3820 channel-attached printers from an I/O configuration. The system must be MVS/ESA SP 4.2.0 and above, and must be running JES2 4.2.0 and above.

E

EBCDIC. Extended binary-coded decimal interchange code.

ECB. Event control block.

EID. Event ID.

electronic forms. A collection of constant data that is electronically composed in the host processor and can be merged with variable data on a page during printing.

electronic overlay. A collection of constant data, such as lines, shading, text, boxes, or logos, that is electronically composed in the host processor and stored in a library, and that can be merged with variable data during printing. Contrast with *page segment*. See also *page overlay* and *medium overlay*.

embedded text control. One or more bytes of control information within a set of text. The text control itself is not printed, but it controls certain operations on following text.

enabled. (1) Pertaining to a state of the processing unit that allows the occurrence of certain types of interruptions. (2) A condition of the printer (physically selected) in which the printer is available to the host processor for normal work. Contrast with *disabled*.

end-user interface. Any product method by which a customer can obtain the services of a product—for example, coding samples, commands and command lists, control statements, display panels, job control language, listings of data streams, messages, program link-editing control statements, and text-formatting commands. Not all products have end-user interfaces; some products provide their services through programming interfaces, and others provide their services only to other products. Contrast with *programming interface for customers*.

EOC. End object container, which marks the end of the object.

equipment check. An asynchronous indication of a printer malfunction.

ERP. Error recovery procedures.

error recovery procedures (ERP). Procedures designed to help isolate and, where possible, to recover from errors in equipment. The procedures are often used in conjunction with programs that record information on machine malfunctions.

ES for OS/2. Extended Services for OS/2 2.0 or later.

ESA. Enterprise System Architecture.

escape character. In a text-control sequence, the control character X'2BD3' that indicates the beginning of the sequence and the end of any preceding text.

ESCON. Enterprise system connection

ESTAE. Extended specify task abnormal exit.

ETFF. Eject to front facing.

Ethernet. A 10-Mb baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission. Ethernet uses carrier sense multiple access with collision detection.

EUI. End user interface.

event control block (ECB). A control block used to represent the status of an event.

EWS. Early warning system.

execution. The process by which a computer carries out one or more of the instructions in a computer program. (I) (A)

exception. A condition in which the printer:

- Detects an invalid or unsupported command, order, control, or parameter value from the host.
- Finds a condition about which the host system must be notified.
- Detects a condition that makes it necessary for the host system to resend data.

exception highlighting. The markings placed on the printed page to indicate the location of a data-stream error.

exit. See *installation exit*.

Extended Architecture (XA). An MVS operating system.

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 eight-bit characters.

extended specify task abnormal exit (ESTAE). A macroinstruction that a user can issue to intercept an abend.

F

FCB. See *Forms Control Buffer*.

FDEF. Form definition.

FGID. See *Font Typeface Global Identifier*.

fiche. See *microfiche*.

file. A member of a partitioned data set or a sequential data set.

fixed metrics. Measurement information in specific units such as pels, inches, or centimeters for individual or collections of graphic characters. Contrast with *relative metrics*. See also *font metrics*.

flash. See *forms flash*.

floating overlay. Synonym for *page overlay*.

FLU. See *Font Library Update utility*.

FOCA. Font object content architecture.

font. (1) A family or assortment of characters of a given size and style—for example, 9-point Bodoni Modern. (A) (2) One size and one typeface in a particular type family, including letters, numerals, punctuation marks, special characters, and ligatures. (3) A paired character set and code page that can be used together for printing a string of text characters. A double-byte font can consist of multiple pairs of character sets and code pages. See *coded font* and *double-byte coded font*.

font character set. Synonym for *character set*.

Font Library Update utility (FLU). An MVS-based utility processed against the AFP font library to ensure that the appropriate GRID information is available for processing COM text.

font mapping. PSF has internal tables that equate core raster fonts to core outline fonts. This comparing and matching activity is called mapping. The default is for PSF/MVS not to map fonts, but you can set XTP7MTOF to ON (B‘1’) in Exit 7 (APSUX07) to enable PSF to map raster fonts to outline fonts.

font metrics. Measurement information that defines individual character values, such as height, width, and space, as well as overall font values, such as averages and maximums. Font metrics may be expressed in specified fixed units, such as pels, or in relative units that are independent of both the resolution and size of the font. See also *character metrics*.

font section. Synonym for *coded font section*.

Font Typeface Global Identifier (FGID). A unique font identifier that can be expressed as either a 2-byte binary value or a 5-digit decimal value. The FGID is used to identify a type style and the following characteristics: posture, weight, and width.

form. A division of the physical medium. Multiple forms can exist on a physical medium. For example, a roll of paper might be divided by a printer into rectangular pieces of paper, each representing one form. Envelopes are an example of a physical medium that comprises only one form. The IPDS architecture defines four types of form: cut-sheets, continuous forms, envelopes, and computer output on microfilm. Each type of form has a top edge. A form has two sides—a front side and a back side. Synonymous with *sheet*. See also *medium*, *sheet*, and *page*.

format. (1) A specified arrangement of such things as characters, fields, and lines; usually used for displays, printouts, or files. (2) To arrange such things as characters, fields, and lines. (3) (v.) To prepare a document for printing in a specified format.

formatted print records. Line data made up of records that are formatted for printing on line printers. PSF uses a page definition to print formatted records on page printers.

FORMDEF. A JCL parameter that specifies a form definition. See *form definition*.

form definition. A resource used by PSF that defines the characteristics of the form, which includes such functions as overlays to be used (if any), paper source (for cut-sheet printers), duplex printing, text suppression, the position of MO:DCA-P data on the form, and the number and modifications of a page.

forms control buffer (FCB). A buffer for controlling the vertical format of printed output. The forms control buffer is a line-printer control that is similar to the punched-paper, carriage-control tape used on IBM 1403 printers. On AFP page printers, the forms control buffer is replaced by the page definition. See *page definition*.

forms flash. In the 3800 printer, the function of the printer that enables it to print user-prepared photographic images with variable MO:DCA-P data. When forms flash is used, the printer operator must insert a frame containing a photographic negative into the printer.

frame. A rectangular area on microfilm, bounded by imaginary intersecting, orthogonal lines that form a grid pattern, within which data can be recorded. The grid pattern does not actually appear on the microfilm.

FSA. Functional subsystem application.

FSI. Functional subsystem interface.

FSS. Functional subsystem.

functional subsystem (FSS). The PSF address space created by JES.

functional subsystem application (FSA). An area within the PSF functional subsystem that drives and manages a single printer.

functional subsystem interface (FSI). A set of services that provides for communication between the JES address space or direct printer services subsystem and the PSF functional subsystem.

fuser counter. A counting device in a printer that identifies how many pages have successfully passed the jam-recovery point defined by the printer. See *jam-recovery page counter*.

fuser station. The printer assembly that bonds the toned image to the paper by heat and pressure.

FW. Font width.

G

gateway. A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T)

GCGID. See *Graphic Character Global Identifier*.

GCSGID. See *Graphic Character Set Global Identifier*.

GDDM. Graphical Data Display Manager.

generalized trace facility (GTF). An optional OS/VS service program that records significant system events, such as supervisor calls and start I/O operations, for use in problem determination.

global resource identifier (GRID). An 8-byte identifier that identifies a coded font resource. A GRID contains the following fields, in the order shown:

1. GCSGID of a minimum set of graphic characters required for presentation. It can be a character set that is associated with the code page, with the font character set, or with both.
2. CPGID of the associated code page.
3. FGID of the associated font character set.
4. Font width, in 1440ths of an inch.

GOCA. Graphics object content architecture.

graphic character. See *character graphic*.

Graphic Character Global Identifier (GCGID). An alphanumeric character string used to identify a specific graphic character. A GCGID can be from 4 to 8 bytes long.

graphic character identifier. See *character identifier*.

Graphic Character Set Global Identifier (GCSGID). A unique graphic character set identifier that can be expressed as either a 2-byte binary value or a 5-digit decimal value.

GRID. See *global resource identifier*.

grid. Format for the frame output on a microfiche—row, column, numeric, alpha, start alphanumeric value.

GTF. Generalized trace facility.

guaranteed print labeling. A method of print labeling that ensures the integrity of the identification label by preventing the user from writing over the label.

H

HAID. Host-assigned ID.

hard page segment. (1) A page segment that is declared in the Map Page Segment structured field and loaded in the printer as a resource that can be reused during the job without being reloaded to the printer. (2) A page segment that is contained within another element, as in an inline resource. Contrast with *soft page segment*.

hardcopy. (1) A copy of a display image generated on an output device such as a printer or a plotter, in a form that can be carried away. (T) (2) A printed copy of machine output in a visually readable form—for example, printed reports, listings, documents, and summaries.

hardware configuration definition. HCD.

hardware default font. The font that the printer uses if no other font is specified.

HCD. Hardware configuration definition.

hexadecimal. Pertaining to a numbering system with base of 16; valid numbers use the digits 0 through 9 and the characters A through F, where A represents 10 and F represents 15.

host font. See *host resource*.

host processor. The processing unit to which the page printers are attached through a data-transfer interface.

host resource. A resource found either in a system library, in a user library, or inline in the print data set.

host system. (1) A data processing system that prepares programs and the operating environments for another computer or controller. (2) The data processing system to which a network is connected and with which the system can communicate.

HRI. Human readable interpretation.

I

IBM compatibility fonts. A group of fonts supplied as part of Print Services Facility, Print Management Facility, and Application System/400. Many of these fonts are derived from fonts created for specific IBM printers (such as the IBM 3800 Model 1) or applications (such as Document Composition Facility). The fonts are called compatibility fonts because they make it possible for applications created for the 3800 Model 1 printer to be migrated to page printers with no need to change the fonts specified in the applications. Examples of IBM compatibility fonts are APL, Boldface, Document, Essay, Format, Gothic, Letter Gothic, Orator, Prestige, Roman, Script, Serif, and Text type families, as well as a set of Proprinter Emulation fonts.

ID. Identification.

identification label. In PSF, a protected set of resources (fonts, overlays, page segments) that are used to label PSF output for security purposes. Contrast with *security label*.

image. A pattern of toned and untoned pels that form a picture.

image block. A structure that contains a raster pattern and the instructions for placing the pattern on the page.

image cell. A part of an image that saves storage by defining only part of a raster pattern. Each image cell must also contain information defining the placement of its raster pattern within the complete image. An image cell can be replicated to fill a defined area.

image data. A pattern of bits with 0 and 1 values that define the pels in an image. (A 1-bit is a toned pel.)

imbedded text control. See *embedded text control*.

IML. Initial microprogram load.

IMM. Invoke Medium Map.

impact printer. A printer in which printing results from mechanical impacts. Contrast with *nonimpact printer*.

impression. The data printed on one side of a sheet. Printer speed is often measured in impressions per minute (ipm).

initial microprogram load (IML). (1) The action of loading microprograms into computer storage. (2) The action of loading licensed internal code from a diskette to writable control storage.

initial program load (IPL). (1) The initialization procedure that causes an operating system to begin. (2) The process of loading system programs and preparing a system to run jobs.

initialize. (1) In programming languages, to give a value to a data object at the beginning of its lifetime. (I) (2) To set counters, switches, addresses, or contents of storage to zero or other starting values at the beginning of, or at prescribed points in, the operation of a computer routine. (A) (3) To prepare for use—for example, to initialize a diskette. See also *initial program load*.

inline. The direction of successive characters in a line of text. Synonymous with *inline direction*.

inline direction. The direction of successive characters in a line of text.

inline resource. A resource contained in the print data set.

input/output (I/O). Pertaining to a device whose parts can perform an input process and an output process at the same time. (I)

installation exit. A subcomponent, installed and maintained for or by a customer installation, that provides control or functions specific to that installation, such as calling exits; providing defaults for job header, trailer, and data-set header separator-page exits; and supporting customer-written exits for logical records, SMF records, message processing, and resource management. PSF invokes these exits at certain predetermined times.

intelligent printer data stream (IPDS). An all-points-addressable data stream that makes it possible to position text, images, and graphics at any defined point on a printed page.

interface. A shared boundary. An interface can be a hardware component used to link two devices, or it can be a portion of storage or registers accessed by two or more computer programs.

interleaving. (1) The simultaneous accessing of two or more bytes or streams of data from distinct storage units. (2) The alternating of two or more operations or functions through the overlapped use of a computer family.

intermediate device. In the IPDS architecture, a device that operates on the data stream and is situated between a printer and a presentation services program in the host. Examples are devices that capture and cache resources and devices that spool the data stream.

internal object. Groups of structured fields that can be included as part of a resource or a print data set but cannot be accessed separately.

intervention-required exception. An error that causes printing to stop until an operator performs a required action.

Internet Protocol (IP). In TCP/IP, a protocol that routes data from its source to its destination in an Internet environment.

Internet Protocol (IP) address. The 32-bit address defined by the Internet Protocol, standard 5, Request for Comment (RFC) 791. It is usually represented in dotted decimal notation.

I/O. Input/output.

I/O error manager. The PSF subcomponent that analyzes I/O errors, determines the recovery action, and directs I/O error recovery and cleanup.

IOCA. Image object content architecture.

IOS. I/O supervisor.

IPDS. Intelligent printer data stream. This is the strategic AFP printer data stream.

IPL. Initial program load.

ISO forms. Pertaining to a set of paper sizes selected from those made standard by the International Organization for Standardization for use in data processing.

J

jam recovery copy counter. A counting device in a printer that identifies how many copies of a page have successfully passed the printer-defined jam recovery point but have not been counted by the jam recovery page counter.

jam recovery page counter. A counting device in a printer that identifies how many pages have successfully passed the printer-defined jam recovery point. See also *fuser counter*.

JCL. Job control language.

JES. Job entry subsystem.

job control language (JCL). A language of control statements used to identify a computer job or describe its requirements to an operating system.

job entry subsystem (JES). An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system.

job header. A page in the printed output that indicates the beginning of a user job. A job can contain one or more data sets. See also *data set header*.

job separation. The ability to identify job boundaries by placing marks or sheets of paper between successive jobs.

job trailer. A page in the printed output that indicates the end of a user job.

JSI. Job submission interface.

K

kanji. Nonphonetic Chinese characters used in writing Japanese. In a font representing kanji characters, each character is represented by a double-byte font.

kerning. The printing of adjacent graphic characters so that their areas overlap on the left or right side.

L

| **LAN**. local area network.

landscape page presentation. The position of a printed sheet that has its long edges as the top and bottom and its short edges as the sides. Contrast with *portrait page presentation*.

LASI. Library access system interface.

LCPC. Load Code Page Control (IPDS command).

LF. Load Font (IPDS command).

LFC. Load Font Control (IPDS command).

LFCSC. Load Font Char Set Control (IPDS command).

LFE. Load Font Equivalence (IPDS command).

library. A file or a set of related files—for example, a page definition library containing one or more page definition files. A library often exists as a partitioned data set in MVS, and as a minidisk in VM.

library access system interface (LASI). The PSF subcomponent that gets resources from the libraries and stores and obtains records in a message data set.

library member. Synonym for *resource object*.

LID. Local identifier.

ligature. A single character representing two or more input characters—for example, *ff*, *ffi*, and *ffl* are characters that may be presented as ligatures.

line data. Data prepared for printing on a line printer such as a 3800 Model 1 Printing Subsystem. Line data is usually characterized by carriage-control characters and table reference characters. Contrast with MO:DCA-P data.

line descriptor. Specifications that describe how input data records are formatted into individual print lines. Line descriptors are interpreted by PSF when formatting printed output.

line merging. Printing two or more records of line data at the same location on the page. Line merging is used with line data to mix different fonts on the same line, to underscore or overstrike, and, on impact printers, to create darker print.

line-mode printing. Every 3800 printer can operate in “line mode” as a non-AFP printer. Print jobs that run in “line mode” do not use PSF or AFP resources such as page definitions or form definitions. Instead, they use 3800 line-mode resources such as FCBs and GRAPHMODs.

line printer. A device that prints a line of characters as a unit. Contrast with *page printer*.

lines per inch (lpi). (1) A unit of measurement for the specification of baseline placement. (2) A measure of the number of lines per vertical inch of paper.

| **local area network (LAN)**. (1) A computer network
| located on a user's premises within a limited
| geographical area. Communication within a local area
| network is not subject to external regulations; however,
| communication across the LAN boundary may be
| subject to some form of regulation. (T) (2) A network in
| which a set of devices are connected to one another for
| communication and that can be connected to a larger
| network.

local-attached. In PSF, an SNA-attached device that does not have a communications controller in its configuration. For example, a 3812 printer connected to a channel-attached 3174 control unit that is defined to the host system through VTAM is considered a local-attached printer. Contrast with *communication-attached*.

local identifier. A 1-byte identifier assigned to parts of the data stream to facilitate PSF processing. For example, the Map Coded Font structured field assigns each coded font a local identifier. When a coded font is required for processing, this identifier is specified in the Set Coded Font Local text control.

logical page. (1) The area on a surface of a form that is formatted for printing. (2) A collection of data that can be printed on one side of a sheet of paper. (3) A presentation space. One or more object areas or data blocks can be mapped to a logical page. A logical page has specifiable characteristics, such as size, shape, orientation, and offset, and is rectangular. Orientation and offset are specified relative to a medium coordinate system. See also *form* and *page*.

logical page origin. (1) The point on the logical page from which positions of images, graphics, page overlays, and text with 0-degree inline direction are measured. (2) The point on the logical page represented by Xp=0, Yp=0 in the Xp coordinate system.

logical unit (LU; L-unit). L-unit) (1) In SNA, a port through which an end user accesses the SNA network to communicate with another end user, and through which the end user accesses the functions provided by system services control points (SSCPs). A logical unit can support at least two sessions, one with an SSCP and one with another logical unit, and may be able to support many sessions with other logical units. (2) A unit of linear measurement. For example, in Mixed Object Document Content Architecture (MO:DCA) and AFP data streams, the following measurements are used:

1 L-unit = 1/1440 inch
1 L-unit = 1/240 inch

logon mode. In VTAM, a subset of session parameters specified in a logon-mode table for communication with a logical unit.

logon-mode table. In VTAM, a table of macro-generated constants that associate a logon-mode name with a set of session parameters. Each logon mode is identified by a logon-mode name.

lpi. Lines per inch.

LU (logical unit) base. The linear measurement base. The value defines, for any object, the meaning of the L-unit values. See also *logical unit*.

L-unit. See *logical unit*.

LU type 1. An SNA logical unit type that provides a communication protocol among host application programs and terminals. Some printers also use this protocol to communicate with host application programs.

LU type 6.2. An SNA logical unit type that converges functions from existing LU types to provide a single, interchangeable communication protocol. See also *Page Printer Communication Component*.

M

macro. Synonym for *macroinstruction*.

macroinstruction. An instruction that causes the execution of a predefined sequence of instructions.

magnetic ink character recognition (MICR). Recognition of characters printed with ink that contains particles of a magnetic material. (I) (A)

mandatory print labeling (MPL). A class defined to the resource access control facility (RACF) that causes PSF to automatically label separator pages and data pages and to enforce the user-printable area.

mapping. See *font mapping*.

marking. A method of updating certain structured fields to identify a resource as printer-resident.

maximum transmission unit. In LANs, the largest possible unit of data that can be sent on a given physical medium in a single frame.

media origin. The first hardware addressable point on the physical page. The point from which the logical page origin is positioned by the medium map. This point is represented by Xm=0, Ym=0 in the Xm, Ym coordinate system. The media origin is defined relative to the top edge of the medium.

medium. (1) A physical carrier of electrical energy. (2) A physical material (for example, paper) in or on which data is represented (for example, printed). See also *form*.

medium map. An internal object in a form definition that controls modifications to a form, page placement, and overlays. Synonymous with *copy group*.

medium overlay. An electronic overlay that is invoked by the medium map of a form definition for printing at a fixed position on the form. See *page overlay*.

member ID. The member name of a resource object, minus the 2-character prefix. For example, BITR is the member ID of the font whose member name is X1BITR.

member name. The name under which a file is stored in a library. For example, X1BITR is the member name of a font in the font library.

message data set. (1) A data set on disk storage that contains queues of messages awaiting transmission to particular terminal operators or to the host system. (2) In PSF, a virtual data set built by the Library access system interface (LASI) subcomponent in memory to store error messages for printing at the end of the document.

metrics. Synonym for *font metrics*.

MICR. Magnetic ink character recognition.

microfiche. Microfilm output on a rectangle piece of film, consisting of a matrix arrangement of data frames, index frames, and title information.

microfilm. A film containing a photographic record of printed matter, on a reduced scale.

microfilm device. Non-IBM output device that presents a hardcopy on microfilm.

microfilm setup resource. A setup file that contains information used to present AFP data on microfilm. See also *object container*.

microfilm utility. A non-IBM utility that builds an object container for microfilm setup data and places it in the AFP library.

Mixed Object Document Content Architecture (MO:DCA). An IBM-architected, device-independent data stream for interchanging documents.

mixed-pitch font. A font that simulates a typographic font. The characters are in a limited set of pitches—for example, 10 pitch, 12 pitch, and 15 pitch.

MO:DCA data. Print data that has been composed into pages. Text-formatting programs (such as DCF) can produce composed text data consisting entirely of structured fields.

MO:DCA data definition. A resource containing a set of formatting controls for printing logical pages of data. Includes controls for the number of lines per printed sheet, font selection, and print direction, and for mapping individual fields in the data to position on the printed sheets.

MO:DCA data page. A page of print data consisting entirely of structured fields.

MO:DCA document. (1) A new term for the obsolete term *composed-text document*. (2) Data composed entirely of structured fields and containing a Begin Document structured field and an End Document structured field.

MO:DCA print data set. A print data set consisting entirely of structured fields.

MO:DCA-P. See *Mixed Object Document Content Architecture-Presentation Format.*

modified-default form definition. A form definition that was the default specified in the PRINTDEV statement of the PSF startup procedure, and that has been modified by the groupvalue parameter of the COPIES parameter or by the FLASH parameter in the JCL statement.

modified-default page definition. A page definition that was the default specified in the PRINTDEV statement of the PSF startup procedure, and that has been modified by a font list specified in any of the following:

- The CHARS parameter from the user JCL or the PRINTDEV statement
- The UCS parameter from the user JCL
- The JES default font in the current printer setup

MOF. Metric-only font.

monospaced font. A font in which the graphics characters have a uniform character increment. Synonymous with *uniformly spaced font*. Contrast with *proportionally spaced font*.

MPL. Mandatory print labeling.

| **MTU**. Maximum transmission unit.

multiple-entry font. A font with multiple entries in the Map Coded Font structured field. The only fonts that have multiple entries are double-byte fonts that are defined dynamically. (The MCF points directly to a set of code page and font character set pairs.)

multiple-up. The printing of more than one page of application data on a single surface of a sheet of paper.

Multiple Virtual Storage (MVS). An IBM operating system consisting of MVS/System Product Version 1 and the MVS/370 Data Facility Product operating on a System/390 processor.

multiplexer. (1) A device that takes several input signals and combines them into a single output signal in such a manner that each of the input signals can be recovered. (T) (2) A device capable of interleaving events of two or more activities or capable of distributing events of an interleaved sequence to the respective activities. (A)

multiplexer channel. A channel designed to operate with a number of I/O devices simultaneously. Several I/O devices can transfer records at the same time by interleaving items of data.

multiplexing. In data transmission, a function that enables two or more data sources to share a common transmission medium so that each source has its own channel. (I) (A)

multipoint network. More than two devices sharing the same transmission line at the same time.

MVS. Multiple Virtual Storage.

MVSCP. MVS Configuration Program.

MVS/ESA. Multiple Virtual Storage/Enterprise System Architecture.

N

NA. Not applicable.

NACK. See *negative acknowledgment reply*.

NCP. Network Control Program.

NCP generation. The process by which the host processor assembles and link-edits a macroinstruction program to produce a Network Control Program.

negative acknowledge reply (NACK). A reply from a printer to a host indicating that an exception has occurred.

nested resource. A resource mapped in an overlay.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

nonimpact printer. A printer in which printing is not the result of mechanical impacts—for example, thermal printers, electrostatic printers, and photographic printers. Contrast with *impact printer*.

NOP. No operation.

nonprocess runout (NPRO). An operation that moves paper through the paper path without printing.

nonswitched line. A telecommunication line on which communications do not have to be established by dialing. Contrast with *switched line*.

normal duplex printing. Printing on both sides of the paper so that the sheets can be bound on the long edge of the paper. Contrast with *simplex printing*. See also *tumble duplex printing*.

notify subtask (NST). (1) A PSF subcomponent that returns processed data sets to JES or to the Direct Printer Services Subsystem and performs checkpoint processing on data sets as they are printed. (2) An external trace that contains information pertaining to the releasing and checkpointing of data sets by PSF.

NPRO. Nonprocess runout.

NST. Notify subtask.

null name. A token name with X'FFFF' in the first 2 bytes.

null value. A parameter for which no value is specified.

numeric data. (1) Data represented by numerals. (2) Data in the form of numerals and special characters—for example, a date represented as 81/01/01. See also *character data*.

N_UP printing. In basic N_UP printing, the dividing of a side of a sheet into a fixed number of equal-size partitions. For example, N_UP 4 divides each side of the sheet into four equal partitions. In enhanced N_UP printing, the sheet can be divided into 8 partitions, each of which can be anywhere on a single side of the sheet.

O

object. (1) A resource or a sequence of structured fields within a larger entity, such as a page segment or a composed page. (2) A collection of structured fields, of which the first provides a begin-object function and the last provides an end-of-object function. One example of an object is a form definition.

object container. (1) A set of structured fields used to carry object data for a variety of objects. (2) A MO:DCA-architected envelope that contains data to be processed by COM; transparent to PSF. This data is classified as “non-presentation” data. A MO:DCA “Registered Object Identification” (ROID) field is defined for the object container to denote the data type. For a microfilm device, this ROID denotes the contained data as either setup data or tape label data.

OC. See *Object container*.

offset stacking. A function that offsets printed output pages for easy separation of print jobs.

OGL/370. Overlay Generation Language/370.

OPC. Obtain printer characteristics.

Operating System/2 (OS/2). IBM's operating system for the IBM Personal System/2 or a compatible.

Operating System/400 (OS/400). IBM's operating system for Application System/400.

option. (1) A specification that can be included in a statement to specify details about the execution of a program. (2) A choice offered from a list of possibilities.

orientation. The number of degrees an object is rotated relative to a reference—for example, the orientation of an overlay relative to the logical page origin. This usually applies to blocks of information. Character rotation applies to individual characters. See also *text orientation*.

origin. A pel position from which the placement and orientation of text, images, and page segments are specified. For example, pages, overlays, and page segments have origins.

OS/2. Operating System/2.

OS/4. Operating System/400.

outline font. A font-shape technology in which the graphic character shapes are represented in digital form by a series of mathematical expressions that define the outer edges of the strokes. The resulting graphic character shapes can be either solid or hollow. Outline fonts can be scaled (sized) to any size. The name of an IBM outline font character set has the prefix *CZ*. Contrast with *raster font*.

overlay. See *electronic overlay*.

Overlay Generation Language/370 (OGL/370). An IBM licensed program for use in creating graphics by designing objects for electronic overlays, such as lines, boxes, shadings, and irregular shapes.

OVP. Operator viewing page.

P

| **packet**. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

page. (1) A collection of data that can be printed on a one side of a physical sheet of paper or form. (2) The boundary for determining the limits of printing. See also *logical page* and *physical page*.

PAGEDEF. A JCL parameter that specifies a page definition. See *page definition*.

page definition. A resource used by PSF that defines the rules for transforming line data into MO:DCA-P data and text controls.

page environment. The size of the page or overlay control data to be used in MO:DCA-P data text blocks, and the page segments or fonts to be used.

page format. A subset of a page definition, containing controls that govern the arrangement of text on a page. See *data map*.

page mode. The mode of operation in which a page printer can accept a page of data from a host processor to be printed on an all-points-addressable output medium. Data may consist of pages composed of text, images, overlays, or page segments.

page origin. See *logical page origin*.

page overlay. An electronic overlay that can be invoked for printing and positioned at any point on the page by an Invoke Page Overlay structured field in the print data. See *medium overlay*.

page position. A control in the copy group to assign the top-left boundary point of the logical page on a sheet for a data set. The page position is determined from the media origin.

page printer. Any of a class of printers that accepts MO:DCA pages, constructed of page data and images, among other things. A page printer is a device that prints one page as a unit. (I) (A) Contrast with *line printer*.

Page Printer communication component (PPCC). The access method that provides the SNA communication interface between printers and PSF.

Page Printer Formatting Aid/370 (PPFA/370). An IBM licensed program for use in creating and storing form definitions and page definitions, which are resource objects for print-job management. These stored objects can then be used to format printed output.

page segment. A resource containing MO:DCA data and images, prepared before formatting and included during printing. A page segment assumes the environment of an object in which it is included.

paging. (1) The transfer of pages between real storage and auxiliary storage. (I) (A) (2) In System 390 virtual storage systems, the process of transferring pages between real storage and external page storage.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted.
(3) Data passed between programs or procedures.

partial page. A page that does not contain all the intended data. Partial pages can be printed after an error is sensed.

partition. In basic N_UP printing, the division of the medium presentation space into a specified number of equal-sized areas in a manner determined by the current physical medium.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Contrast with *sequential data set*.

pass-through. To gain access to another network.

pattern storage (PST). An area of storage that holds the raster patterns for fonts and images.

PCC. Page Counter Control (IPDS command).

PDEF. Page definition.

PCL. Printer control language.

PDS. Partitioned data set.

pel. Picture element. Synonymous with *pixel*.

pending page queue (PPQ). In PSF, a list of pages that has been processed by PSF but has not been released from JES.

physical page. A single surface (front or back) of a form. See also *form* and *page*.

physical medium. A physical entity on which information is presented. Examples of physical media are display screens, paper, foils, microfilm, and labels.

physical unit (PU). The part of a communication controller or cluster controller that manages the resources attached to the node.

picture element (pel). (1) An element of a raster pattern about which a toned area on the photoconductor might appear. See *raster pattern*. (2) The smallest unit that can be addressed by a page printer. See *addressable point*.

PID. Process ID.

PING command. The command that sends an Internet Control Message Protocol (ICMP) echo request packet to a gateway, router, or host with the expectation of receiving a reply.

pipeline. (1) A serial arrangement of processors or a serial arrangement of registers within a processor. Each processor or register performs part of a task and passes results to the next processor; several parts of different tasks can be performed at the same time. (2) The hardware path between the channel station or received-page station and the stacker.

pitch. (1) A unit of measurement for the width of a printed character, reflecting the number of times a graphic character can be set in one linear inch. (2) The character size represented by the number of characters that can be printed horizontally in an inch; for example, 10 pitch has 10 graphic characters per inch. Uniformly spaced fonts are measured in pitch. Contrast with *point*.

pixel. Synonym for *pel*.

PLR. Priority level resource.

PMF. Print Management Facility

point. A unit of about 1/72 inch used in measuring type. Contrast with *pitch*.

point size. The height of a font, in points.

point-to-point network. An arrangement in which two devices share the same transmission line at the same time.

POR. Power-on reset.

portrait page presentation. The position of a printed sheet that has its short edges as the top and bottom and its long edges as the sides. Contrast with *landscape page presentation*.

Positive Acknowledge Reply (ACK). In the IPDS architecture, a reply to an IPDS command which has its ARQ flag on and in which no exception is reported.

power-on reset (POR). See *initial microprogram load*.

PPCC. Page Printer Communication Component.

PPFA/370. Page Printer Formatting Aid/370.

PPP. Primary print process.

PPQ. Pending page queue.

PPWTR. Page printing writer.

prefix mapping. The process that PSF uses to change font names at the coded font level to a name appropriate to the font technology that the printer supports. For example, XZ is the prefix for an outline font. When the printer supports only bounded-box raster fonts, PSF changes the prefix to X0. X0 is the prefix for the same font in the raster font library.

preprinted form. A sheet of paper containing a preprinted design of constant data into which variable data can be printed. See also *electronic overlay* and *forms flash*.

presentation device. A device that produces character shapes, graphics pictures, images, or bar code symbols on a physical medium. Examples of physical media are display screens, paper, foils, microfilm, and labels.

presentation text. Synonym for MO:DCA data.

printable area. The area on a sheet of the paper where print can be placed.

print data set. A data set created by an application program that contains the actual information to be printed and, optionally, some of the data that controls the format of the printing. The three kinds of print data set are composed text, line format, and mixed format.

print data stream. The data stream that is created by PSF and transmitted to the printer.

print direction. (1) The direction in which characters are added to a line. (2) In PSF, the specification of inline direction for the printing of text.

print file. A file created by an application program that contains the actual information to be printed and some of the data that controls the format of the printing. Print files can contain MO:DCA data, line data, or a combination of MO:DCA and line data.

print job. The data that the user submits to PSF to be printed. The user can request that a print job be printed as though it were multiple data sets.

print labeling. A controlled method of placing identification labels on each page of PSF printed output.

print position. Any location on a medium where a character can be printed.

print quality. (1) The measure of printed output against existing standards and in comparison with jobs printed previously. (2) The ability of some page printers to print data at more than one level of print quality, such as “draft” and “near-letter” quality.

Print Services Facility (PSF). A licensed program that manages and controls the input data stream and output data stream required by supported IBM page printers. PSF combines print data with other resources and printing controls to produce AFP output.

priority level resource (PLR). A marked resource known to PSF and deleted according to a modified deletion algorithm. When resources according to reasonable resource loaded value, PSF deletes all non-PLR resources before deleting any PLR resources. A frequently used resource can be marked as PLR with the Resource Exit; this improves its chances of being retained at the data set end.

processor. In a computer, a functional unit that interprets and executes instructions. (I) (A)

programming interface for customers. Any product method that enables a customer-written program to obtain the services of the product—for example, CSECT names, data areas or control blocks, data sets or files, exits, macros, parameter lists, and programming languages. Not all products have programming interfaces for customers; some products provide their services through end-user interfaces, and others provide their services only to other products. Contrast with *end-user interface*.

programming request for price quotation (PRPQ). A customer request for a price quotation on alterations or additions to the functional capabilities of system control programming or licensed programs. The RPQ may be used in conjunction with computing system RPQs to solve unique problems in data processing.

program status word (PSW). An area in storage used to indicate the order in which instructions are executed, and to hold and indicate the status of the computer system.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of the program.

proportionally spaced font. A typographic font, or in some usages a mixed-pitch font. See *typographic fonts* and *mixed-pitch fonts*.

protocol. The meanings of and the sequencing rules for requests and responses by which network addressable units (PU, LU, SSCP, and VTAM programs) in a communication network coordinate and control data transfer operations and other operations.

PSAF. Print Services Access Facility.

PSC. PostScript.

PSEG. Page segment.

PSF. Print Services Facility.

PSF Direct. A function of PSF/2 or PSF for AIX that enables another PSF program—PSF/VM, PSF/MVS, PSF/VSE, or PSF/400—to print remotely, using the SNA LU 6.2 protocol, on PSF/2 or PSF for AIX printers. The PSF program sends the print data stream directly to the PSF/2 or PSF for AIX printer, bypassing the OS/2 or AIX spool. The operator of the originating system controls printing on the PSF/2 or PSF for AIX printers as if the printers were attached to the originating system.

PST. Pattern storage.

PSW. Program status word.

PTF. Program temporary fix.

PTOCA. Presentation test object content architecture.

PU. Physical unit.

R

RACF. Resource Access Control Facility.

raster font. A font technology in which the graphic characters are defined directly by the raster bit map. Contrast with *outline font*.

raster overlay. See *raster pattern overlay*.

raster pattern. (1) A series of picture elements (pels) in scan lines to form an image. See also *page image*. (2) A pattern of bits with 0 (off) and 1 (on) that define the pels in an image. A 1-bit is a toned pel.

raster pattern overlay. An overlay loaded in the printer as a raster pattern rather than as a sequence of printer commands. Contrast with *coded overlay*.

raster pattern storage (RPS). An area of storage that holds raster patterns for fonts and images.

RCB. Resource control block.

RDW. Record descriptor word.

real storage. The main storage in a virtual storage system. Physically, real storage and main storage are identical. Conceptually, however, real storage represents only part of the range of addresses available to the user of a virtual storage system. Conventionally, the total range of addresses available to the user has been provided by the main storage. (I) (A)

reasonable resource loaded value (RRLV). A value used by PSF to keep the number of resources loaded at a manageable level. At data set end, PSF deletes resources for a particular resource type until this value is reached.

received page counter. A counting device in a printer that identifies how many pages have been successfully received and their syntax checked. Synonymous with *channel counter*.

record descriptor word (RDW). Data preceding a variable record that specifies the length of the entire record, including the RDW.

record interface. Coordinates the transmission of the printer resources needed to print a document.

recovery termination manager (RTM). A program that handles all normal and abnormal termination of tasks by passing control to a recovery routine associated with the terminated function.

region size. The amount of main storage available for a program to run in.

relative metrics. Measurement information that is defined in relation to some other units. Relative values are expressed as fractional parts of a unit square design space (Em-square), whose sides correspond to the vertical size of the font. This makes it possible to use the same value for different point sizes and different raster-pattern resolutions. The use of relative values necessitates the definition of the units of measure for the Em-square, the point of the font, and perhaps the pel resolution of the raster. Contrast with *fixed metrics*. See also *font metrics* and *bounded-box relative metrics*.

reload function. The Resource Exit can request that a resource be “reloaded..” PSF will not use an existing version of the resource, but will load the resource from a host library.

repositioning. A process in which Print Services Facility, following an indication from the printer or from JES of a potentially recoverable error, locates the correct spool record for recomposing one or more pages for printing.

Request for Comments (RFC). In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

request for price quote (RPQ). See *computing system RPQ*.

request unit (RU). In SNA, a message unit that contains control information such as a request code, or function management (FM) headers and end-user data, or both. Synonymous with *request*.

resident resource. A resource that resides in a printer (fonts or symbol sets) or in DPF (fonts, page segments, and overlays). Resident resources are identified by the APSRMARK utility of PSF or by the resident font resource table APSRFTBL.

resolution. In computer graphics, a measure of the sharpness of an image, expressed as the number of lines and columns on the display screen or the number of pels per unit of linear measure.

resource. (1) A collection of printing instructions used by Print Services Facility in addition to the print data set to produce the printed output. PSF resources include coded fonts, font character sets, code pages, page segments, overlays, form definitions, and page definitions. (2) Any source of aid used for performing a task—for example, disk storage space, computer processing time, and communications lines.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

resource-constrained condition. Occurs when the printer does not have enough storage for the resources required to print the current page.

resource manager. Any control program function responsible for allocating a resource. In PSF, it maintains the resource control structure used to manage resources; directs the retrieval of resources from libraries; deletes resources from the printer; calls for Resource Exit functions; and includes the APSRMARK utility with which resources can be marked as printer-resident or in DPF.

resource name. The name under which a resource object is stored; the first 2 characters indicate the resource type.

X0-XG, XZ	Coded font
T1	Code page
C0-CG, CZ	Font character set
S1	Page segment
F1	Form definition
P1	Page definition
O1	Overlay

resource object. A member in a resource library. Synonym for *library member*.

RFC. Request for Comments.

RLE. Run length encoding.

ROID. Registered object ID.

root. The user name for the system user with the most authority.

rotation. Synonym for *character rotation*. See also *orientation*.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer.

routine. A program, or a sequence of instructions, that is called by a program and that may have some general or frequent use. (I) (A)

RP. Raster processor.

RPC. Remove procedure call.

RPQ. Request for price quotation. See *computing system RPQ*.

RPS. Raster pattern storage.

RRB. Record resource block.

RRLV. Reasonable resource loaded value.

RTM. Recovery termination manager.

RU. Request unit.

rule. A solid or patterned line of any weight, extending horizontally or vertically across a column, row, or page.

S

SBCS. Single-byte character set

SCSI. Small computer system interface

SDF. Structured data field.

SDLC. Synchronous Data Link Control.

SDSF. Self-defining structured field.

security definition. In MVS, a member containing the definitions for one identification label. These definitions include instructions for the overlay name, and the size and origin of paper to be used.

security definitions library. In MVS, a partitioned data set, or a series of concatenated partitioned data sets, containing the security definitions for an entire system.

security label. In a trusted computing base, a security label used to maintain multiple levels of security on a system. This label is a combination of a security class and a security level. Contrast with *identification label*.

security overlay. An overlay, such as one created by use of Overlay Generation Language, that resides in a secure library and is used to place security resources on a page.

segment. Synonym for *page segment*.

sense data. (1) Data describing an I/O error. Sense data is presented to a host system in response to a Sense I/O command. (2) In SNA and TCP/IP, the data sent with a negative response indicating the reason for the response.

sequence number. A 2-byte field in a structured field introducer that identifies the position of the structured field in the data set.

sequential data set. In MVS, a data set whose records are organized on the basis of their physical positions, such as on magnetic tape. Contrast with *partitioned data set*.

SF. Structured field.

sheet. A single piece of paper. For cut-sheet printers, a synonym for *form*. The IPDS architecture defines four types of sheets: cut-sheet forms, continuous forms, envelopes, and computer output on microfilm. Each sheet has a front and a back side. Some types of media consist of multiple sheets. For example, a roll of continuous forms can be divided at the perforations into rectangular sheets. Each sheet usually has carrier or tractor-feed strips. Microfilm is another example of a medium comprising multiple sheets, whereas an envelope comprises only one sheet.

shift-out, shift-in (SOSI). Special EBCDIC control codes that exist in the data stream to indicate the switches between double-byte fonts and single-byte fonts.

simplex printing. Printing on only one side of the paper. Contrast with *duplex printing*.

single-byte coded font. A font in which the characters are defined by a 1-byte code point. A single-byte coded font has only one coded font section. Contrast with *double-byte coded font*.

single-entry font. A font with a single entry in the Map Coded Font structured field. This includes all single-byte fonts and any double-byte fonts being defined as coded fonts.

skip. (1) To ignore one or more instructions in a sequence of instructions. (2) A move of the current print position to another location.

SMF. System management facilities.

SMIT. System management interface tool.

SMP/E. System Modification Program/Extended.

SNA. Systems Network Architecture.

SNA-attached. In PSF, pertaining to a device that is linked to the host system through VTAM and uses an SNA protocol to transfer data. It does not need to be physically connected to the host; some printers are attached to a control unit, a communication controller, or both, and they can transfer data over telecommunication lines. For example, a 3820 attached to a communication network that uses the LU 6.2 communication protocol to receive data from a communication controller is considered an SNA-attached printer. Contrast with *channel-attached*.

soft page segment. A resource that is not declared in the Map Page Segment structured field but is sent to the printer inline with data. Contrast with *hard page segment*.

SOSI. Shift-out, shift-in.

spot carbon. Paper from which carbon is omitted in certain areas to suppress the printing of data on specific copies.

SRM. System Resource Manager.

SSCP. System services control point.

SSI. Subsystem Interface.

stacked-page counter. A counting device that identifies how many pages have been successfully stacked.

startup procedure. A procedure used to start PSF and to specify PSF initialization parameters and libraries that contain system resources.

station ID. For the 3800 printer, a 2-byte pipeline counter that is incremented when the last copy of a page reaches the point in the hardware represented by that counter.

STM. Sense type and model.

storage. (1) A unit into which recorded text can be entered, in which it can be retained and processed, and from which it can be retrieved. (T) (2) The action of placing data into a storage device. (I) (3) A storage device. (A)

structured field. A self-identifying string of bytes and its data or parameters.

structured field introducer. The first 8 bytes of a structured field, which indicate its length, type, and number.

subgroup. A set of modifications within a copy group that applies to a certain number of copies of a form. A copy group can contain more than one subgroup.

subnet mask. For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address.

subpage. A part of a logical page on which line data can be placed. In the page definition, multiple subpages can be placed on a physical page as specified in the print data.

subsystem interface (SSI). See *functional subsystem interface*.

supervisor call (SVC). An instruction that interrupts a program being run and passes control to the supervisor to perform a service specified by the instruction.

suppression. See *text suppression*.

suppression local ID. A value assigned in the Map Suppression coded field to a suppression named in a Line Descriptor structured field. This value is contained in the Begin and End Suppression text controls.

suppression number. An identification number from 1 to 127 generated by the Map Suppression structured field and assigned to text designated for suppression in the data map transmission subcase.

swapping. A process that interchanges the contents of an area of real storage with the contents of an area in auxiliary storage.

switched line. A telecommunication line on which communications are established by dialing. Contrast with *nonswitched line*.

switched major node. In VTAM, a major node whose minor nodes are physical and logical units attached by switched SDLC links.

SVC. Supervisor call.

symbol set. A type of font that resides in a printer but has fewer attributes than can be specified for resident coded fonts.

synchronous. (1) Pertaining to two or more processes that depend on the occurrences of a specific event such as common timing signals. (I) (A) (2) Occurring with a regular or predictable time relationship.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous information transfer over a data link connection.

SYSGEN. System generation. (A)

SYSIN. System input stream.

SYSOUT. System output stream.

system input stream (SYSIN). A data definition (DD) statement used to begin an in-stream data set.

system library. A resource library specified in the PRINTDEV statement, as distinguished from *user library*.

system management facilities (SMF). An optional control program that provides the means for gathering and recording information that can be used to evaluate system use.

System Modification Program (SMP). An IBM licensed program used to install software and software changes.

system output stream (SYSOUT). A data definition (DD) statement used to identify a data set as a system output data set.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through networks, and controlling the configuration and operation of those networks. See also *Page Printer Communication Component*.

systems services control point (SSCP). In SNA, a focal point with an SNA network for managing the configuration, coordinating network-operator and problem-determination requests, and providing directory support or other session services for end users of the network.

T

table reference character (TRC). An optional control character in an input record that specifies the font to be used to print the record. The table reference character corresponds to a font number defined in a page definition font list or to the order of font names listed in the job control CHARS parameter.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

TCP/IP-attached. In PSF, a device that is linked to the MVS system through a TCP/IP network and receives data from the MVS system using the application-layer IBM protocol for IPDS printers. Some TCP/IP-attached printers require the i-data 7913 IPDS Printer LAN Attachment. Contrast with *SNA-attached* and *channel-attached*.

telecommunication line. The part of a data circuit, external to the equipment, that connects to a data-switching exchange. Contrast with *data link*.

terminate. (1) In SNA products, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure for ending one or more designated LU-LU sessions. (2) To stop the operation of a system or device. (3) To stop the execution of a program.

text. A graphic representation of information on an output medium. Text can consist of alphanumeric characters and symbols arranged in paragraphs, tables, and columns.

text control. Structured field data that controls the format, placement, and appearance of text.

text control sequence. A text control and its associated data.

text orientation. A description of the appearance of text as a combination of print direction and character rotation.

text suppression. The intentional omission of sections of text in copy groups as specified in a parameter in the form definition.

throughput. (1) A measure of the amount of work performed by a printer over a period of time—for example, the number of impressions per minute. (2) A measure of the amount of work performed by a computer system over a period of time—for example, number of jobs per day. (I) (A) (3) In data communication, the total traffic between stations per unit of time.

title. Eye-readable strip made up of one and sometimes two adjacent frame rows, usually located at the top of each microfiche. Titles can also be used with 16mm microfilm.

TLE. Tag logical element.

token name. An 8-byte name that can be given to all internal objects and resources.

token ring. A network configuration in which tokens are passed in a circuit from node to node. A node that is ready to send data can capture the token and insert data for transmission.

trace. A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A)

transaction ID. An identifier assigned by Document Processor, associated with an I/O buffer, and representing a form of resource.

transfer counter. A counting device that identifies how many pages have been committed to printing and have been removed from the page buffer. Synonym for *committed page counter*.

transmission. (1) The sending of data from one place for reception elsewhere. (A) (2) The sending of a print data set to a printer.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet protocol as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

TRC. Table reference character.

tumble duplex printing. Duplex printing for sheets that are to be bound on the short edge of the paper regardless of whether the presentation is portrait or landscape. Contrast with *normal duplex printing*.

two-channel switch. A hardware feature with which an I/O device can be attached to two channels. A dynamic switch can be added, which makes it possible for both interfaces to be enabled at the same time with channel selection determined by programming.

typeface. A collection of fonts all having the same style, weight, and width. Each font differs from the others by point size or type family.

type size. (1) A measurement in pitch or points of the height and width of a graphic character in a font. (2) One of the many attributes of a font; other attributes, for example, are weight and width.

typographic font. A font in which the distance between characters varies. The distance from one character to another is adjusted to improve the visual flow of text by eliminating excess space.

U

UCS. Universal character set.

unbounded-box font. A font designed to use unbounded character boxes. Contrast with *bounded-box font*.

unbounded character box. A character box design that requires the character position information to be included (as untoned pels) within the box. Contrast with *bounded character box*.

unformatted print records. Line data made up of fields of data that have not been formatted into print lines. PSF uses a page definition to format these records for printing on page printers.

uniformly spaced font. A font in which all the characters have the same character increment. Contrast with *proportionally spaced font*.

units of measure. Counting methods that are either specific or relative. See also *font metrics* and *bounded-box relative metrics*.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays. Synonymous with *font*.

UPA. User-printable area.

user library. A private print-resource library owned by an individual user, accessed only when the name is specified by the owner in the JCL USERLIB parameter.

user printable area (UPA). The area within the valid printable area (VPA) in which user-generated data can print without causing an exception condition. See also *valid printable area*.

utility program. (1) A computer program in general support of computer processes—for example, a diagnostic program, a trace program, or a sort program. (T) Synonymous with *service program*. (2) A program designed to perform an everyday task such as copying data from one storage device to another. (A)

V

valid printable area (VPA). The intersection of the physical page with the current logical page or the current overlay. See also *user-printable area*.

value. A quantity assigned to a constant, a variable, a parameter, or a symbol in a command.

virtual storage. (1) The storage space which may be regarded as addressable main storage by the user of a computer system and in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations. (I) (A) (2) Addressable space that is apparent to the user as the processor storage space, from which the instructions and the data are mapped into the processor storage locations. Synonymous with *virtual memory*.

Virtual Telecommunications Access Method (VTAM). A set of programs that maintains control of the communication between terminals and application programs running under DOS/VS, OS/VS1, and OS/VS2 operating systems.

VPA. Valid printable area.

VTAM. Virtual Telecommunications Access Method.

W

WCS. Writable control storage.

WOC. Write object container.

WOCC. Write object container control.

writable control storage (WCS). Printer storage in which data can be entered, held, and retrieved. Writable control storage contains licensed internal-code instructions and other control information, such as the print buffer.

write-to-operator (WTO). An system service that enables a program to write a message notifying the system console operator of conditions that may need correcting.

WTO. Write-to-operator.

X

XA. Extended Architecture.

X-axis. In printing, an axis perpendicular to the direction in which the paper moves through the printer. See also *Y-axis*.

X-extent. A measurement along the X-axis.

XFP. X-Series Future Platform.

Xm, Ym coordinate system. The media coordinate system.

Xp, Yp coordinate system. The logical page coordinate system.

Y

Y-axis. In printing, an axis parallel to the direction in which the paper moves through the printer. See also *X-axis*.

Y-extent. A measurement along the Y-axis.

Numerical Entries

16mm. Microfilm with frames that are arranged in a 16mm-high strip, usually 100 feet long, to fit in a viewer cartridge. Microfilm uses sequential retrieval marks for view positioning.

Index

A

abends, restartable
- controlling in APSUX07 198, 213
- description 249

ACQINTV parameter 53
active environment group (AEG) 197
adding font mapping
- outline to raster 148
- raster to outline 147

additional IBM publications 345
Address Resolution Protocol tables 240
AEG (active environment group) 197
AFP performance 218
AFP/COM
- attaching channel-attached printers 253
- attaching ESCON-attached microfilm device 253
- channel-attached 13
- checkpointing JCL parameters 253, 262
- COM-Recorder, SNA-attached 42
- COMSETUP keyword 255
- defining a microfilm device to PSF 254
- defining microfilm device to MVS 13
- defining microfilm device to PSF 255
- defining to JES2 253
- defining to JES3 253
- ESCON-attached 13
- FONTDD keyword 255
- hardware requirements 251
- libraries, PSF 258
- library data set 259
- NPRO JCL parameter 253
- OBJCONDD keyword 255
- object containers 307
- operating a microfilm device 262
- performance considerations 262
- PRINTDEV parameters, PSF 254
- PSF Software Requirements 251
- PSF system programmer tasks 252
- redirecting messages 258
- resources 259
- security considerations 262
- suppressing (redirecting) messages 258
- switching setup procedures 262
- system overview 252
- tasks, PSF system programmer 252
- utility 259

algorithm, resource-deletion 211
anamorphic scaling 300
APPL statement 24
APPLID parameter 53
APSLCRL parameter area 291
APSRCF30 font conversion program 309
- description 309
- error handling 310

APSRFTBL 130
- font resource table exit, modifying 130

APSRGTNM grid parts to font member names 322
APSRMARK utility
- control statement syntax for resident fonts 133
- examples of control statements for DPF 156
- examples of control statements for resident fonts 136
- general description for resident fonts 131
- page segments used with DPF 160
- run APSW4028 first 132, 140

APSRNTGM font member names to grid parts 325
APSUBTBL
- block letter tables 180

APSURLST
- parameter area 294

APSUX01
- exit for job header separator pages 175

APSUX02
- exit for job trailer separator pages 175

APSUX03
- exit for data set header separator pages. 175

APSUX04
- exit for record processing 183

APSUX05
- exit for SMF type 6 record processing 188

APSUX06
- exit for message processing 190

APSUX07
- begin-data-set call (BDSC) 205, 214
- definition 193
- dependant overlays in, 195
- double-byte font 208
- exit for controlling resources 204
- field descriptions
  - CRLAVOLS 294
  - CRLBASE 293
  - CRLCATN 294
  - CRLCONC 293
  - CRLDATA 294
  - CRLDDNM 293
  - CRLDELN 294
  - CRLDSNM 294
  - CRLEID 293
  - CRLEND 294
  - CRLHIGCN 294
  - CRLID 293
  - CRLLEN 293
  - CRLLOWCN 294

APSUX07 *(continued)*
field descriptions *(continued)*
CRLNVOL 294
CRLPFIX 293
CRLPTF 293
CRLREL 293
CRLSPN 293
CRLTRVD 293
CRLVOLS 294
RLSTAFLG 299
RLSTATTR 297
RLSTAUX 298
RLSTCFUS 302
RLSTCID 296
RLSTCPDS 301
RLSTCPFN 301
RLSTCPID 297
RLSTCPIL 302
RLSTCPNM 302
RLSTCPUS 302
RLSTDEP 298
RLSTDFT 297
RLSTDPS 301
RLSTEXTC 301
RLSTEXTP 296
RLSTFGID 297
RLSTFLG 299
RLSTFLGS 299
RLSTFNDS 301
RLSTFNIL 302
RLSTFNNM 302
RLSTFW 297
RLSTGCID 297
RLSTGRID 299
RLSTHDR 301
RLSTHFS 300
RLSTHSCF 300
RLSTINL 297
RLSTKEEP 299
RLSTLCF 299
RLSTLCP 299
RLSTLENG 296
RLSTLFLG 299
RLSTLFN 299
RLSTLOAD 299
RLSTMEF 298
RLSTMISC 299
RLSTMSGL 298
RLSTMSGP 296
RLSTNAME 297
RLSTNEST 298
RLSTNEXT 296
RLSTOFONT 299
RLSTPAIR 302
RLSTPDS 297
RLSTPFLG 302
RLSTPLR 298

APSUX07 *(continued)*
field descriptions *(continued)*
RLSTPRVS 296
RLSTPTRS 296
RLSTRESC 298
RLSTRID 297
RLSTRRI 299
RLSTSEC 298
RLSTSID 301
RLSTSMSG 299
RLSTSPS 297
RLSTSTAT 298
RLSTUSER 298
RLSTVFS 300
XTP738MO 288
XTP7ABRC 291
XTP7ACSE 285
XTP7ATCH 289
XTP7AUX 285
XTP7C2SI 288
XTP7CID 284
XTP7CRF2 290
XTP7CRLF 290
XTP7CRLP 290
XTP7DHMO 290
XTP7DSAT 285
XTP7EBDS 286
XTP7EFFH 286
XTP7ETFF 286
XTP7ETRM 286
XTP7ETYP 284
XTP7FLG 289
XTP7FMLY 289
XTP7HCA 288
XTP7HDF 286
XTP7HQE 287
XTP7INF 289
XTP7IPCC 285
XTP7IPMF 287
XTP7IPOS 287
XTP7IPSU 288
XTP7JHMO 290
XTP7JTMO 290
XTP7LENG 284
XTP7LSTC 289
XTP7LSTP 289
XTP7MDMO 291
XTP7MDSD 288
XTP7MFLG 290
XTP7MISC 286
XTP7MSGL 290
XTP7MSGP 290
XTP7MTOF 287
XTP7NACC 286
XTP7NDEL 286
XTP7NLD 286
XTP7NOTY 285

APSUX07 *(continued)*
field descriptions *(continued)*
XTP7OVLY 289
XTP7PDFT 285
XTP7PDS 285
XTP7PDSH 285
XTP7PJHD 285
XTP7PJTR 285
XTP7PLDS 285
XTP7PMDS 285
XTP7PNAM 290
XTP7PRND 287
XTP7PRNS 287
XTP7PRT 289
XTP7PRTM 289
XTP7PRTT 289
XTP7RCF 289
XTP7RESC 289
XTP7RFD 289
XTP7RFLG 286
XTP7RFNF 287
XTP7RLST 289
XTP7RMO 289
XTP7ROC 289
XTP7RPD 289
XTP7RPS 289
XTP7RRLV 288
XTP7RSTR 291
XTP7RTYPE 285
XTP7SEC 290
XTP7TDS 286
fonts 208
form definitions 209
general considerations 195, 216
initialization call (INIT) 203
overlays in, 207
page definitions 209
page segments 208
resource deletion at DSE (RDC) 211
resource load call (RLC) 210
resource-access call (RAC) 205
termination call (TERM) 213
APSUX07 processing indicators
RLSTAFLG 304
RLSTATTR 304
RLSTAUX 304
RLSTDEP 304
RLSTDFLG 304
RLSTEXTP 303
RLSTFLGS 304
RLSTMISC 304
RLSTMSGL 304
RLSTMSGP 303
RLSTNAME 304
RLSTNEST 304
RLSTNEXT 303
APSUX07 processing indicators *(continued)*
RLSTPDS 304
RLSTPLR 304
RLSTPRVS 303
RLSTPTRS 303
RLSTRESC 304
RLSTRID 303
RLSTSTAT 304
XTP7 304
XTP7ABRC 303
XTP7ACSE 303
XTP7AUX 302
XTP7C2SI 303
XTP7CRF2 303
XTP7CRLF 303
XTP7CRLP 303
XTP7DHMO 303
XTP7DSAT 302
XTP7DSTY 302
XTP7EFFH 303
XTP7ETFF 303
XTP7ETYP 302
XTP7HCA 303
XTP7HDF 303
XTP7HQUE 303
XTP7IPMF 303
XTP7IPOS 303
XTP7IPPC 303
XTP7IPSU 303
XTP7JHMO 303
XTP7JTMO 303
XTP7MDMO 303
XTP7MDSD 303
XTP7MFLG 303
XTP7MSGL 303
XTP7MSGP 303
XTP7MTOF 303
XTP7NOTY 303
XTP7OVLY 303
XTP7PDFT 302
XTP7PDS 302
XTP7PDSH 302
XTP7PINF 303
XTP7PJHD 302
XTP7PJTR 302
XTP7PLDS 303
XTP7PMDS 302
XTP7PRND 303
XTP7PRNS 303
XTP7RFLG 303
XTP7RFNF 303
XTP7RLST 303
XTP7RRLV 303
XTP7RSTR 303
XTP7RTYP 302
XTP7TDS 303

APSUX15 213
APSW4028
  run before APSRMARK 132
APSWPROT procedure, for TCP/IP-attached printers 240
APSXUCOM macro 179
ARP tables 240
ARPAGE statement, in TCP/IP profile 240
auxiliary, default page and form definitions 209

B

Basic N_UP printing 122
BDSC (begin-data-set call) 205, 214
blank pages after separator pages 88
block letter tables, APSUBTBL 180
block size 232
block-letter routine 181
bootp
  configuration worksheet 332
bootp daemon 331
  setting up on AIX 335
  setting up on OS/2 336
bootptab
  editing the file 332
buffer sizes, specifying in TCP/IP profile 237, 238
buffer, pool 114
BUFNO parameter 53
BURST option 66

C

capture, font 141
carrier-strip marking 90
carrier-strip, mark form type 90
channel paths and switches 8
CHARS parameter 53
checkpoint 5
  data 5, 248
  data set 66, 72
  deferred-printing mode function 97
  interval 66, 72, 113
  JES 248
  JES3 72
  microfilm 253
  microfilm performance 262
  performance 72
  PSF 5, 113
  seconds between 66, 72
  starting point 66
  value 66
coding the APSXUCOM macro 179
communication network, establishing 17
communication-attached devices (token ring) 22
communication-attached printers 18
complex image 221
complex MO:DCA-P data 221
COMPRESS parameter 54
COMSETUP parameter 54
concatenated
  libraries 47
  numbers 211
  retrieval lists (CRLs) 204, 211
configuration considerations for 3130 printer 45
configuration considerations for 3160 printer 45
configuration considerations for 3935 printer 45
CONNINTV parameter 54
control block listings, PSF 279
control statements 136
Conversion Utility Program, VSE 306
converting Julian dates 179
copies of separator pages 87
copy grouping 119
copy marking 91
CRLs (concatenation retrieval lists) 204, 211
cross-reference, program indicator 302

D

data integrity 124
data protection
  See protecting data
data separation 232
data set checkpoint 66
data set, termination time 205, 210, 211
DATABUFFERPOOLSIZE statement, in TCP/IP profile 237
DATACK parameter 55
dates, converting Julian 179
default form definition in the PRINTDEV statement 117
default page definition in the PRINTDEV statement 117
default, auxiliary page and form definitions 209
deferred-printing mode
  PSF startup procedures 52, 74
  What to do first 63
defining communication-attached printers 17
defining local-attached printers 17
Defining Printers to MVS 7
definition considerations 43
deleting font mapping
  raster to outline 150
deleting resources
  full control 212
  partial control 212
  using PLRs 212
DEQ or ENQ 194, 204, 210
DEVICE statement 71
diagnostic aids
  trace reports 249

direct-printing mode
PRINTDEV statement 52
PSF startup JCL 100
disabled mechanisms 94
DISCINTV parameter 56
Distributed Print Function (DPF) 17
Distributed Print Function of PSF/2 87, 153
marking resources for 153
distribution information, separator page 83
documentation 345
double-byte font 208
DPAGELBL parameter 56
DRM see dynamic reconfiguration management 13
DUMP parameter 56
duplex printing 220
dynamic allocation
of printers 63, 98
dynamic reconfiguration management (DRM)
3820 channel-attached printer 5, 14
AFP1 printers 5, 14
channel-attached printer installation 52
HCD with DRM capabilities 13
HCD without DRM capabilities 13
how to add a 3820 channel-attached printer 14
how to add an AFP1 printer 14
MVS SP 4.2.0 and above 52
MVSCP 5

E

ECAMFCS 91
carrier-strip field 91
example code 169
for separator page exits 91
mark form carrier strip field 169
mark form carrier-strip field 91
to disable marking 91
to initiate 91
where to set this field 91
which exits 91
enhanced N_UP printing 123
ENQ or DEQ 194, 204, 210
Enterprise Systems Connection
See ESCON
error messages, PSF 247
error-handling for font conversion 310
ESCON
channel 11
defining channel-attached printers with MVSCP 11
using channel-attached printers 8
establishing a communication network 17
establishing network nodes 17
estimates, minimum storage 114
exit processing 164
exit table, APSUCOM 179
exit-put routine 182

F

FAILURE parameter 56
FDEFDD parameter 56
field descriptions for APSUX07 279
FLU
microfilm utility 259
font capture 141, 223
single-byte raster font 141
font conversion program (APSRCF30) 309
description 309
error handling 310
Font Library Update utility
See FLU
font mapping
adding new 147, 148
considerations, outline 202
considerations, recovering from font-not-found 203
creating new 144
disabling 150
outline to raster, adding 148
raster to outline, adding 147
raster to outline, disabling 150
tables 146
font mapping tables 146
font member names to grid parts (APSRNTGM) 325
font pruning 222
deactivate 222
double-byte fonts 287
non-3800 double-byte font fonts 287
performance considerations 222
processing time 222
single-byte fonts 287
when useful 222
font resource table 130
for 4224 and 4234 printers 130
FONTDD parameter 57
fonts
adding outline to raster mapping 148
adding raster to outline mapping 147
APSRMARK 131
calling out 208
considerations, mapping to outline 202
considerations, recovering from font-not-found 203
converting from 240-pel to 300-pel 309
creating new 144
disabling mapping, raster to outline 150
double-byte 208
FONTDD parameter on PRINTDEV statement 57
in resource access call 208
in resource list control block (RLST) 208
libraries 113, 129
library 47
mapping considerations, outline 202

fonts *(continued)*
mapping tables 146
MICR 124
outline, performance 223
printer-resident 133
recovery considerations, font-not-found 203
resident font utility 131
resource management 113
resource selection table, deferred environment 121
resource selection table, direct printing environment 122
restriction when substituting 209
section ID 211
specifying in startup procedures 240
FORCE parameter on MODIFY command 244, 248
form definitions
default and auxiliary 209, 210
in the resource access call 209
in the resource load call 210, 211
inline 210
modified-default 199
resource management 113
resource selection table, deferred environment 119
resource selection table, direct-printing environment 119
user-specified 210
with RLST chain 209
formatted line data 221
FORMDEF parameter 57
forms flash 118
forms marking 88
FSI above the line 6
FSI below the line 6
FSSDEF statement 70
full control of resource deletion 212
Functional Subsystem (FSS)
numbers of in operating system 63
Functional Subsystem Application (FSA)
numbers of in operating system 63
functional subsystem interface (FSI) 99, 232
deferred-printing mode 232
How PSF is marked 232
performance 232

G

GATEWAY statement, in TCP/IP profile 238, 239
gateway, token-ring 22
grid parts to font member names (APSRGTNM) 322
GROUP statement 35, 37, 42

H

hard page segment 208
hardware configuration definition
hardware configuration definition (HCD) 7, 236
For channel-attached 3800 page printers 13
For channel-attached 3820 page printers 13
microfilm device 13
microfilm printer 13
hardware problem, recovering from 248
HCD
See hardware configuration definition (HCD)
header pages 92
hierarchy for resource selection 117
fonts, deferred environment 121
fonts, direct-printing environment 122
form definitions, deferred environment 119
form definitions, direct-printing environment 119
interface attachment feature, direct-printing environment 124
introduction to 117
MICR, direct-printing environment 124
page definitions, deferred environment 120
page definitions, direct-printing environment 121
searching for resource specified for print job 122
Huffman compression 203, 224, 229

I

i-data 7913 IPDS Printer LAN Attachment
attaching 338
checking 330
configuration 329
configuring 331
installing 342
operating 338
pinging 341
setting the controls 336
Setting up the controls for the Ethernet LAN model 337
shutting down 339
starting 338
testing 340
troubleshooting 342
I/O subsystem capacity 228
indicator cross-reference, program 302
INIT (initialization call) 203
initialization call (INIT) 203
inline resources
copy grouping 117
integrity for 124
Input, PAGEDEFs and FORMDEFs 231
installation exits
for formatting separator pages (APSUX01, APSUX02, APSUX03) 175
for message processing (APSUX06) 190
for processing logical records, APSUX04 183
for resource control (APSUX07) 193
for SMF type 6 record processing, APSUX05 188
put routine interface, APSUPUT 182

installation tasks 7
installation verification program (IVP) 52, 317
  240-pel resolution printer sample 317
  example of output 317
installation-exit processing 164
intermediate device 6
INTERVAL parameter, for keep-alive transmissions 238
intervention timer, local 56, 248
IPADDR parameter 57, 240
IPDS 61, 329
  installation 329
issue an optional message 205
IVP
  See installation verification program (IVP)

J

JCL for startup procedures 80, 242
JCL parameters
  COPIES 119
  description of 113
  examples of 242
  FLASH 118
  for starting PSF 52—62
  on the PRINTDEV statement 52—62
  PIMSG 116
  used to generate a report 137
  used to mark a MVS member 136
  used with the APSRMARK utility 133
JES (Job Entry System)
  initialization 65
  JES2 initialization 69
  JES2 PRINTDEF 67, 69
  JES3 initialization 70
JES3 initialization statement and the PRINTDEV statement 209
job separator pages 82
job-header page 82
job-trailer page 82
JOBHDR parameter 57
JOBTRLR parameter 57
Julian dates, converting 179

K

keep-alive transmissions 238
KEEPALIVEOPTIONS statement, in TCP/IP profile 238

L

least-recently-used algorithm 211
libraries
  containing AFP resources 47
  fonts 47
libraries *(continued)*
  form definitions 47
  format of 47
  overlays 47
  page definitions 47
  page segments 47
  working with 47
library, resources 153, 154
line mode
  APSUX04 108
  enable 108
  IBM 3800 printers 107
  limitations, conversion function 109
  overview 107
  programming considerations 109
  programming considerations, Exit 4 108
  requesting 185
LINE statement 36, 37, 40, 43
listings, PSF control block 279
lists (CRLs), concatenation retrieval 204, 211
local intervention timer 56
local major nodes 28
local-attached printers 17
local-intervention timer 248
LOGMODE parameter 57
logon-mode table 24
LU statement 33, 38, 41
LUNAME parameter 57

M

macros (ENQ or DEQ) 194, 204, 210
macros for programming interfaces 319
magnetic ink character recognition printing 124
major nodes, local 28
mapping (fonts)
  adding new 147, 148
  considerations, outline 202
  considerations, recovering from font-not-found 203
  creating new 144
  disabling 150
  outline to raster, adding 148
  raster to outline, adding 147
  raster to outline, disabling 150
  tables 146
mapping tables 146
mark form carrier-strip (MFCS) 90
marking a font 131
marking a resource 131, 154
maximum transmission unit, in TCP/IP profile 238
message data set redirection 115
MESSAGE parameter 58
message, optionally issued 205
messages, PSF 247
MFCS, mark form carrier-strip 90

MGMTMODE parameter 58
MICR Postprocessor 124
  direct-printing environment 124
  interface attachment feature 124
microfiche
  See microfilm
microfilm
  microfilm 249
microfilm device
  ESCON-attached 9
MODEENT statement 25
MTU, in TCP/IP profile 238, 239
multipoint telecommunication line 34
MVS configuration program 7, 236
  For 3800 printers 9
  For channel-attached 3820 printers 10
  microfilm device 7
MVS/370
  numbers of in operating system 63
MVS/ESA
  numbers of FSS and FSA 63

N
N_UP printing 122
NCP
  major node 33
  MAXOUT value 227
  performance 227
nested resource 197, 212, 213
network, communication 17
nodes, local major 28
NOTIFY JCL keyword
  controlling print complete notification 128
  creating profiles 128
  giving access authority 128
NPRO 127
NPRO parameter 58
number, concatenation 211

O
OBJCONDD parameter 58
objects, resource
  See resource
operating a TCP/IP-attached printer 244
optionally issue a message 205
outline fonts
  adding mapping of raster 147
  adding mapping to raster 148
  considerations 202
  considerations, recovering from font-not-found 203
  creating new 144
  disabling mapping of raster 150
  mapping considerations 202
  mapping tables 146
outline fonts *(continued)*
  performance 223
  recovering from font-not-found considerations 203
overlays
  considerations with APSUX07 197
  in the resource access exit 207
  limits in RRLVs 211
  maximum supported in direct-printing mode 98
OVLYDD parameter 58

P
packet-size, in TCP/IP profile 238
page definitions
  default and auxiliary 209
  in the resource access call 209
  in the resource load call 210, 211
  inline 210
  modified-default 199
  resource management 113
  resource selection table, deferred environment 120
  resource selection table, direct-printing environment 121
  user-specified 210
  with RLST chain 209
page segments
  as a nested resource 212
  defining in an overlay 197
  deleting auxiliary 212
  hard 208
  maximum supported in direct-printing mode 98
  retaining 195
  soft 208
  when RRLV equals 0 212
  with reload option 208
  with selective deletion 212
parameter areas
  APSLCRL 291
  APSURLST 294
  XTP7 279
parameters
  ACQINTV 53
  APPLID 53
  BUFNO 53
  CHARS 53
  COMPRESS 54
  COMSETUP 54
  CONNINTV 54
  DATACK 55
  DISCINTV 56
  DPAGELBL 56
  DPF 159
  DSHDR 56
  DUMP 56
  FAILURE 56
  FDEFDD 56

parameters *(continued)*
FONTDD 57
FORMDEF 57
IPADDR 57
JES2 PRINTDEF 67, 69
JOBHDR 57
JOBTRLR 57
LOGMODE 57
LUNAME 57
MESSAGE 58
MGMTMODE 58
NPRO 58, 253
OBJCONDD 58
OVLYDD 58
PAGEDEF 58
PDEFDD 59
PIMSG 59
PORTNO 59
PRINTDEV 52
PSEGDD 59
RELINTV 59
RELMODE 59
SDEFDD 60
SETUP 60
SFONTDD 60
SOVLYDD 60
SPAGELBL 60
SPSEGDD 60
SYSAREA 60
TIMEOUT 60
TRACE 60
partial control of resource deletion 211
PATH statement 32
paths, channel 8
PDEFDD parameter 59
performance factors
attachment hardware 225
blank compression 229
block size 232
buffers 231
BUFNO parameter 231
channel-attached printer 225
checkpointing 232
checkpointing, microfilm 262
coat-tailing 227
communication-attached printer 225
complex image 221
complex MO:DCA-P data 221
data checks 231
data compression 223
data density and complexity 220
data rates 226
data separation 232
data transmission 223
Distributed Print Function (DPF) of PSF/2 226
DPF 226

performance factors *(continued)*
DPF Resource Saving 226
duplex printing 220
features and RPQs 220
formatted line data 221
FSI 232
host system 228
I/O capacity 228
impressions per minute (ipm) 219
internal processing speed 219
local-attached printer 225
MAXOUT value, NCP 227
mechanical speed 219
microfilm 262
NCP MAXOUT value 227
options 220
outline fonts 223
paper size 220
print job 220
print job characteristics 221
print-job size 223
printer data checks 231
printer throughput 219
printer-data requirements 230
processing capability 228
pruning of raster fonts 222
PSF Direct 226
raster storage 220
real storage 228
region sizes 114
resident fonts 222
resource buffer pool 114
resources used 222
RU size 226
simple image 221
simple MO:DCA-P data 221
SPBUFFER value 232
storage estimates 114
system environment 228
system load 228
system requirements 230
system tuning 229
tracing 233
TRKCELL parameter 232
understanding 218
unformatted line data 221
VTAM DELAY parameter 227
VTAM PSNDPAC parameter 227
VTAM request unit (RU) size 226
PIMSG parameter 59
PLR (priority level resource) 212
port number, specifying for TCP/IP-attachment 59
port switching 61
PORTNO parameter 59
Postprocessor 124
direct-printing environment 124

Postprocessor *(continued)*
  interface attachment feature 124
  MICR 124
prefix mapping 146, 202, 287
print-job size 223
PRINTDEF, parameter in JES2 67, 69
PRINTDEV statement 52—62
printer sharing in MVS 60
printers
  disabled mechanisms 94
  dynamic allocation 63, 98
  restriction for selective deletions 212
printing error messages 116
priority level resource (PLR) 212
procedures, PSF startup 74, 240
profile, TCP/IP 237
Program Directory, PSF 7, 75
program indicator cross-reference 302
programming interfaces 319
protecting data 127—128
pruning of raster fonts 222
PSEGDD parameter 59
PSF
  internal processing speed 219
  mechanical speed 219
  performance 219
  rate of printing 219
  understanding syntax 64
PSF control block listings 279
PSF Direct function of PSF for AIX 17
PSF Direct function of PSF/2 17
PSF errors
  responding 247
PSF exits
  for formatting separator pages (APSUX01, APSUX02, APSUX03) 175
  for message processing (APSUX06) 190
  for processing logical records, APSUX04 183
  for resource control (APSUX07) 193
  for SMF type 6 record processing, APSUX05 188
  put routine interface, APSUPUT 182
PSF messages 247
PSF startup procedures 74, 240
PSF support for resident fonts 129
PSNDPAC
  coat-tailing 227
  pacing value 227
  VTAM parameter 227
PU statement 41
publications 345
publications, related 345
put routine interface, APSUPUT 182

R

RAC (resource-access call) 205
  double-byte font in, 208
  fonts in, 208
  form definitions in, 209
  overlays in, 207
  page definitions in, 209
  page segments in, 208
  restrictions with, 209
RACF 127, 128
raster font, single-byte, capture 141
raster fonts
  disabling mapping to outline 150
  mapping of outline 148
  mapping tables 146
  mapping to outline 147
RDC (resource deletion at DSE) 211
re-size 309
real storage 228
reasonable resource loading values (RRLV) 211, 212
  coded fonts 307
  form definitions 307
  overlays 307
  page definitions 307
  page segments 307
redirecting message data set 115
region sizes 114
related publications 345
RELMODE parameter 59
Remote PrintManager
  managing PSF resources 153
  using APSRMARK for resource identification 153
resident fonts
  APSRMARK 131
  examples of control statements 136
  font pruning 222, 287
  general parameters 135
  performance considerations 222
  PSF supported 129
  specific parameters 133
  system environment example 129
  Utility (APSRMARK) 131
resident fonts for 4224 and 4234 printers 130
  APSRFTBL exit 130
resource 113
  access call (RAC) 205
  buffers 114
  definition of 113
  delete at DSE 211
  deletion at DSE (RDC) 211
  exit (APSUX07) 193
  font pruning 222, 287
  full control of deletion 212
  hard page segment 208
  hierarchy of 117

resource *(continued)*
integrity considerations 124
library for DPF 153
library for PSF 154
list control block (RLST) 207, 208
load call (RLC) 210
management 113
nested 197, 212, 213
partial control of deletion 211
performance considerations 222
PLR (priority level resource) 212
resident fonts 222
selection tables and hierarchy 117
soft page segment 208
types managed by PSF 113
Resource access control facility (RACF) 127, 128
Resource Conversion Utility Program 306
responding to PSF errors 247
restartable abends
controlling in APSUX07 198, 213
description 249
restrictions
exception to resource list control block 209
for APSUX07 294
for printers not supporting selective deletion 212
for system macros ENQ or DEQ 211
with RAC 209
retrieval lists (CRLs), concatenation 204, 211
return codes for font conversion 310
RLC (resource load call) 210
RLST (resource list control block) 207, 208
routines, service 181
RRLV (reasonable resource loading values) 211, 212, 307

S

sample jobs, APSRMARK 265
SDEFDD parameter 60
search for specified resource 122
section ID 211
security
See protecting data
Security Labeling
how enabled 128
protect printed output 128
selective deletion restriction 212
SENDGARBAGE parameter, for keep-alive transmissions 238
separating jobs or copies of jobs 81
carrier-strip marking 90
copy marking 91
edge marking 91
header pages 92
separating users' jobs 81
blank pages after separator pages 88

separating users' jobs *(continued)*
forms marking 88
separator pages 82
using MFCS 90
separator pages
DPF considerations 87
for job header and trailer 82
printing multiple copies of 87
service routines 181
SERVICE statement 38
SETROPTS RACLIST processing 128
simple image 221
simple MO:DCA-P data 221
sizes, region 114
SMALLDATABUFFERPOOLSIZE statement, in TCP/IP profile 238
SMESSAGE class 128
SMF type 6 records 93, 188
soft page segment 208
SOVLYD parameter 60
SPAGELBL parameter 60
SPSEGDD parameter 60
startup procedures, PSF 74, 240
storage estimates, minimum 114
switched major node 30
switches, channel 8
syntax 23
syntax for APSXUCOM macro 179
SYSAREA parameter 60
SYSPARM 84
omitted 84
value 84
system environment
storage considerations 114
system environment example 129
system macros 194, 204, 210
System Management Facility type 6 records 93
system tuning 229

T

tables for resource selection 117
tables, font mapping 146
TCP/IP profile, modifying 237
TCP/IP-attached printers
connecting 235
defining MVS communications control unit 236
defining to JES 240
defining to PSF 240
operating 244
pinging 239
port number for 59
software requirements for 235
starting 244
startup procedure for 240
stopping 244

TCP/IP-attached printers *(continued)*
  TCP/IP profile, modifying 237
TERM (termination call) 213
terminating a printer FSA with the MODIFY command 248
termination call (TERM) 213
termination time for data set 205
termination times 194—211
TIMEOUT parameter 60
timer, local intervention 56
timer, local-intervention 248
TINYDATABUFFERPOOLSIZE statement, in TCP/IP profile 238
Token ring, communication-attached devices 22
token-ring subsystem connection 42
trace description 249
TRACE parameter 60
tracing 233
tuning, system 229

U

unformatted line data 221
user job control language (JCL) parameters
  See JCL parameters
user libraries 48
Utilities
  FLU 259
  Font Library Update, microfilm 259
  microfilm FLU 259
  microfilm utility 259
  utility 259
utility
  microfilm utility 259
Utility Program, VSE 306

V

VBUILD statement 31
VSE Resource Conversion 306
VTAM (Virtual Telecommunications Access Method)
  37X5 communication controller 52
  application-program node name 53
  coat-tailing algorithm 227
  communication controller 28, 30
  communication lines 34
  communication-attached devices 20
  defining bits in the station ID 32
  defining local major nodes 28
  defining NCP major nodes 28
  definitions for PSF network nodes 20
  DELAY parameter 227
  dialing the phone 36
  disconnecting the physical unit 31
  DLOGMODE 43
  establishing a communication network 17

VTAM (Virtual Telecommunications Access Method) *(continued)*
  establishing connections 42
  IDBLK value 32
  IDNUM value 32
  IOBUF buffer pool 29
  local-attached devices 28
  logical unit 42
  logical unit (LU) 23
  LOGMODE 43
  logon-mode table entry 24
  logon-mode table entry name 29, 33, 41, 57
  logon-mode table name 39
  MODEENT statement 24
  NCP generation 20
  NCP major node definition 34
  network configurations 34
  network considerations 43
  network definition 57
  OPNDST command 44
  pacing the flow of data 30, 33
  pacing value 227
  performance 227
  PSNDPAC parameter 227
  PSNDPAC value 26
  RU size 29, 226
  RUSIZES value 26
  SIMLOGON command 44
  SNA 20
  SNA set up for ACF/VTAM 52
  SNA set up for VTAM/NCP 52
  SRCVPAC value 26
  syntax rules 23
  when line is active 35
  XID exchange 32

X

XTP7 parameter area 279
XTP7nnnn field descriptions
  See APSUX07

Readers' Comments — We'd Like to Hear from You

Print Services Facility/MVS
System Programming Guide
Version 2, Release 2, Modification 0

Publication No. S544-3672-03

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

Note: IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

	Yes	No
• Does the publication meet your needs?	____	____
• Did you find the information:		
Accurate?	____	____
Easy to read and understand?	____	____
Easy to retrieve?	____	____
Organized for convenient use?	____	____
Legible?	____	____
Complete?	____	____
Well illustrated?	____	____
Written for your technical level?	____	____
• Do you use this publication:		
As an introduction to the subject?	____	____
As a reference manual?	____	____
As an instructor in class?	____	____
As a student in class?	____	____

• What is your occupation?

____

Thank you for your input and cooperation.

Note: You may either send your comments by fax to 1-800-524-1519, or mail your comments. If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

Comments:

____
Name

____
Company or Organization

____
Phone No.

____
Address

____

____

Readers' Comments — We'd Like to Hear from You
S544-3672-03

Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

Information Development
The IBM Printing Systems Company
Department H7FE Building 003G
P O Box 1900
BOULDER CO 80301-9817

Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

IBM®

File Number: S370-40
Program Number: 5695-040

Printed in U.S.A.

S544-3672-03