INTERCOMM

ASMF
USERS GUIDE

ISOGON
CORPORATION
330 Seventh Avenue, New York, New York 10001

LICENSE: INTERCOMM TELEPROCESSING MONITOR

ASMF Users Guide

Publishing History

Publication	Date	Remarks
First Edition	September 1979	This manual corresponds to Intercomm Release 8.0.
Second Edition	April 1980	Revisions.
Third Edition	May 1981	Updates.
SPR 198	August 1981	Updates including addition of Appendix F. "Cross-Reference Programs."
2nd Printing	August 1981	Incorporating SPR 198.
SPR 223	January, 1984	Revisions corresponding to Intercomm Release 9.0, and addition of an index.
SPR 239	July, 1988	Revisions for Release 8, 9, and 10.

PREFACE

Intercomm is a state-of-the-art teleprocessing monitor system, executing on the IBM System/370 family of computers and operating under the control of IBM Operating Systems (MVS/370, MVS/XA). Intercomm monitors the transmission of messages to and from terminals, concurrent message processing, centralized access to I/O files, and the routine utility operations of editing input messages and formatting output messages, as required.

This manual documents the Automated SM (System Modification) Management Facility, a set of programs to aid in the application of SMs to Intercomm. ASMF is available at no charge to all Product Maintenance Agreement subscribers. SMs beginning with 1498 may be applied using ASMF.

The following AMSF topics are discussed in this manual:

- Commands
- Execution
- Application of SMs
- Installation
- Cross-reference listing utilities

There are six appendixes to this manual. Appendix A gives the detailed syntax of SMPROF and SMLEVEL, the two macros used in installing ASMF. Appendix B describes the INTASMF procedure used to apply SMs. Appendix C provides the structure of the SM control records, which are supplied with standard SMs, and which the user must provide for experimental and user SMs. Appendix D gives examples of the printed output reports of ASMF. Appendix E lists messages produced by the facility. Appendix F documents the Intercomm cross-reference modules.

INTERCOMM PUBLICATIONS

GENERAL INFORMATION MANUALS

Concepts and Facilities

Planning Guide

APPLICATION PROGRAMMERS MANUALS

Assembler Language Programmers Guide

COBOL Programmers Guide

PL/1 Programmers Guide

SYSTEM PROGRAMMERS MANUALS

Basic System Macros

BTAM Terminal Support Guide

Installation Guide

Messages and Codes

Operating Reference Manual

System Control Commands

CUSTOMER INFORMATION MANUALS

Customer Education Course Catalog

Technical Information Bulletins

User Contributed Program Description

FEATURE IMPLEMENTATION MANUALS

Autogen Facility

ASMF Users Guide

DBMS Users Guide

Data Entry Installation Guide

Data Entry Terminal Operators Guide

Dynamic Data Queuing Facility

Dynamic File Allocation

Extended Security System

File Recovery Users Guide

Generalized Front End Facility

Message Mapping Utilities

Model System Generator

Multiregion Support Facility

Page Facility

Store/Fetch Facility

SNA Terminal Support Guide

TCAM Support Users Guide

Utilities Users Guide

EXTERNAL FEATURES MANUALS

SNA LU6.2 Support Guide

TABLE OF CONTENTS

Page

Chapter 1

INTRODUCTION

The Automatic System Modification Facility (ASMF) is a system for the automatic control, maintenance and application of program changes (SMs) to Intercomm modules. ASMF processes official System Modifications (SMs) and Experimental Modifications (XMs) provided for the user by the vendor, and User-coded Modifications (UMs). ASMF also permits removal of SMs already applied. Successors and prerequisites are automatically tracked, and a log is maintained. SMs may be selected for application or omitted, as appropriate.

Only one of the above processes can be executed during one ASMF run. The processing is controlled by command control cards, macros and procedural parameters, as described in this manual.

The ASMF system consists of the following:

- Executable ASMF modules

- Modules, generated via ASMF macros, containing control information related to a specific user installation

- Libraries containing the ASMF modules and Intercomm modules

- Other data sets, such as the log, and the modifications themselves

- INTASMF, a JCL procedure, which is controlled by execution parameters and a set of commands

As illustrated in Figure 1, ASMF places the source and object code of the modified modules on intermediate libraries which are separate from the installation's production libraries. This enables an installation to test modified modules without disturbing the existing versions. After testing, ASMF may be used to replace production library modules with modified modules from the intermediate library.

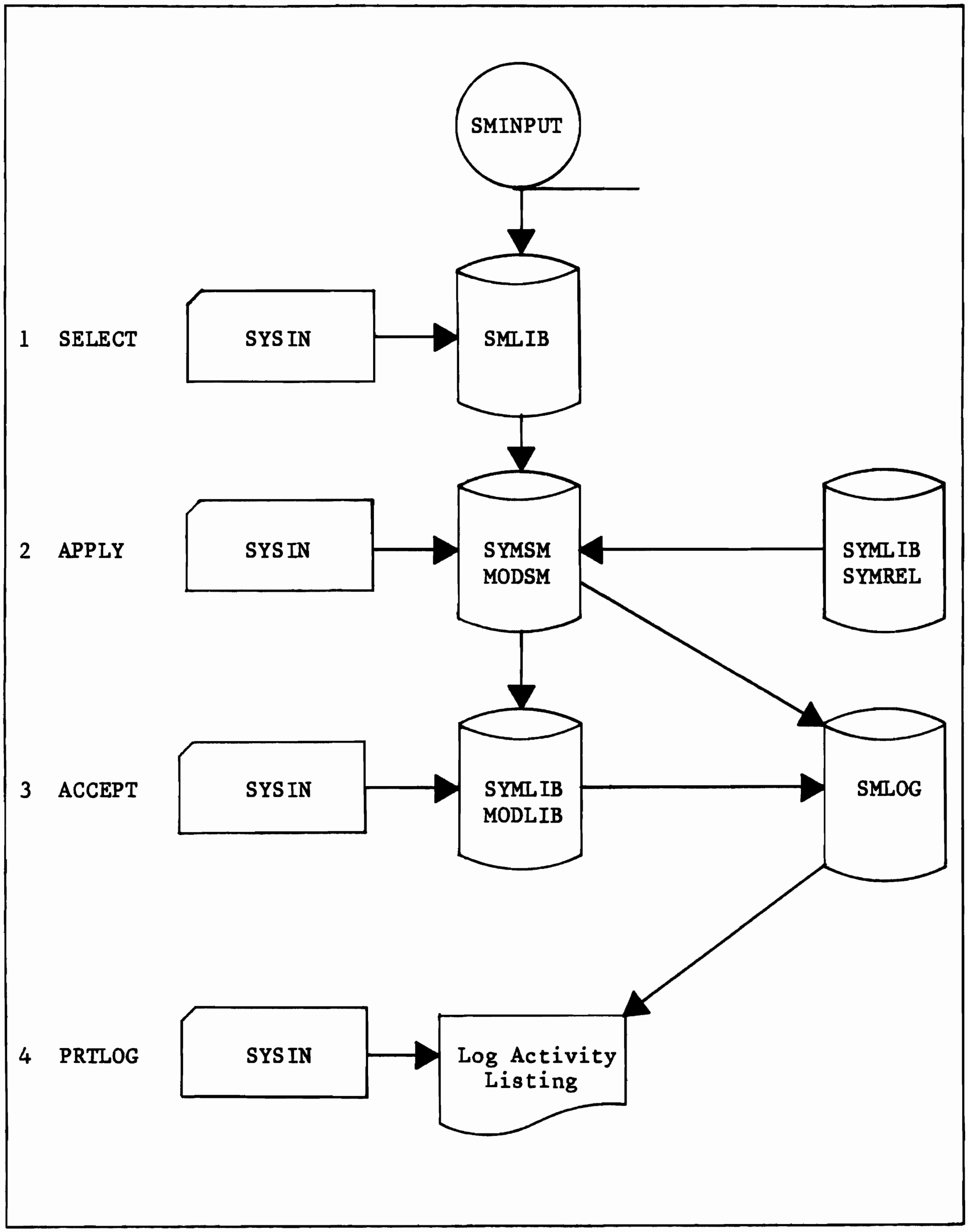

Figure 1. ASMF Process Overview

The ASMF process may be broken down into several discrete steps, as follows:

An SM tape containing Assembler Language source fixes for Intercomm, IEBUPDTE utility control statements and ASMF SM control records (SMINPUT DD), along with user-provided ASMF commands from the input stream (SYSIN DD) are read by the ASMF module in a SELECT operation (1) to produce partitioned data set SMLIB, whose members are of the following forms:

- mmmmmmmm (module name), which contains references to one or more SMs (SMnnnnxx, where nnnn is the SM number and xx is a numeric suffix for member within SM).

- SMnnnn, which may contain prerequisite SM number(s) and will always contain references to one or more Intercomm modules (mmmmmmmm), depending upon the number of modules affected by the SM.

- SMnnnnxx, which contains the IEBUPDTE utility control statements and Assembler Language source code of an SM for changing one, and only one, module. SMnnnn00 will always be produced for an SM. SMnnnn01-99 may be produced, if the particular SM affects two or more modules.

The SMLIB data set, along with ASMF commands from the input stream (SYSIN DD) are then input to an APPLY operation (2) which performs updates to Intercomm modules (copied from SYMLIB, or, if not found, from SYMREL) onto SYMSM and performs assembly and linkedits of Intercomm modules to MODSM. Note that if the APPLY step fails and it is necessary to start over, the steps listed in Chapter 5 to reuse ASMF must first be executed, then rerun the SELECT step. To test the new SMs, the order of DD statement concatenation for SYSLIB in the Intercomm linkedit must specify MODSM before MODLIB and MODREL. After thorough testing of the Intercomm system, as modified by the current batch of SMs, data sets SYMSM and MODSM, along with ASMF commands from the input stream (SYSIN DD) are input to an ACCEPT operation (3). Unlike SELECT and APPLY, for which ASMF provides the reversal operations REJECT and DELETE, there is no ASMF- supplied reversal operation for an ACCEPT. The ACCEPT operation copies (via IEBCOPY) the modified modules from SYMSM and MODSM to the production libraries SYMLIB and MODLIB. A record of activity is kept on SMLOG. SMLOG may be printed at any time by using the PRTLOG command (4).

SYMLIB and MODLIB may not contain either user mods (UMs) or experimental SMs (XMs). After APPLYing a new set of SMs (output to SYMSM), they are to be examined to determine if user mods to the affected modules are still viable (needed, can be applied), and if an official SM has replaced code provided previously by an Experimental SM. If not, the UM or XM may be reapplied with this output directed to SYMUSR. Then assemble and linkedit the module from SYMUSR to MODUSR before testing the new SMs. The order of load library concatenation for SYSLIB in the Intercomm linkedit before testing then becomes: MODUSR, MODSM, MODLIB, MODREL. Remove MODSM after ACCEPTing the new SMs.

Block sizes for SYMREL, SYMLIB, SYMUSR and SYMSM must all be the same, likewise for MODREL, MODLIB, MODUSR and MODSM.

Standard SMs are distributed to Product Maintenance Agreement subscribers on a 1600 BPI standard label tape. File 1 contains the SMs in a printable format. File 2 contains the SMs along with ASMF SM control records (as described in Appendix C) for use with the INTASMF procedure. The procedure is listed in Appendix B; note that the DD statement SMINPUT refers to File 2 of the SM tape (or the user's XMs or UMs file).

The following JCL may be used to print the first file of the SM tape:

```
//stepname   EXEC   PGM=IEBGENER
//SYSPRINT     DD   SYSOUT=A
//SYSIN        DD   DUMMY
//SYSUT2       DD   SYSOUT=A
//SYSUT1       DD   DSN=INT.SM.PRINT.mmm,DISP=SHR,DCB=DEN=3,
//                  UNIT=TAPE,VOL=(PRIVATE,SER=volser),
//                  LABEL=(1,SL)
```

where mmm is the three-character month name on the external SM tape label; for example, JAN=January.

This file contains four parts as follows:

- Cumulative SM index by member name, describing all SMs issued to date for the current Intercomm release
- Monthly index by SM, listing the new SMs (and applicable modules) in ascending numerical order
- Detailed description and change decks for each SM.
- Listing of 80-character records on File 2 (ASMF control records and change decks for each member within each SM).

If you do not have a bundled (base Intercomm plus all special features) system, the SM printout should be examined to determine whether some SMs affect modules not provided on SYMREL (such as those related to the TOTAL data base, or VTAM Front End, for example). In this case, the select step can be used to specify a list of applicable SMs, rather than trying to apply all the SMs. Also note that on the detailed description of each SM, the experimental XM which this SM replaces is given in the upper right corner. Ensure that modules to which those experimental XMs may have been applied are not on any of the libraries used for the ASMF process or for subsequent assemblies and linkedits of Intercomm modules or user tables.

Chapter 2

COMMANDS

2.1 INTRODUCTION

There are seven ASMF commands:

- SELECT--load selected SMs from the data set with the ddname SMINPUT (File 2 on the SM tape as distributed, or the user or experimental SM data set) onto the ASMF control data set (SMLIB) and verify if they are applicable to the user's system.
- APPLY--apply selected SMs to the test libraries (SYMSM/MODSM).
- REJECT--remove selected SMs which were applied to the test (SM) libraries without removing them from the ASMF control data set.
- DELETE--REJECT selected SMs, and then remove them from the ASMF control data set.
- ACCEPT--copy updated modules from the test libraries (SYMSM/MODSM) to the production libraries (SYMLIB/MODLIB).
- PRINTTP--print selected SMs. (See Appendix D.1.)
- PRTLOG--print the log data set. (See Appendix D.2.)

ASMF command cards are input via SYSIN, and may be intermixed with control parameter cards overriding predefined installation parameters, as explained in Section 2.4.

In a given run, all commands must apply to SMs, XMs, or UMs. The three types may not be mixed.

2.2 FORMAT

Commands are free-form in card columns 1 through 71, except for continuations, which must begin in column 1. A nonblank character in column 72 indicates that the next card is a continuation. Commands and parameters are separated by one or more blanks.

2.3 SYNTAX

The following are the syntax specifications of the commands. Explanations of the parameters are given in Section 2.3.6.

2.3.1 SELECT and PRINTTP

```
{SELECT } [{XMS}]  {for-month }  [{{EXCEPT}sm-list}] [FORCE]
{PRINTTP} [{UMS}]  {(month-no)}  [{{[ONLY]}       }]
          [{SMS}]                [{ALL            }]
```

2.3.2 APPLY

```
APPLY [{XMS}] [{{EXCEPT}sm-list}] [{UPDONLY}] [FORCE] ['comment']
      [{UMS}] [{{[ONLY]}       }] [{ASMONLY}]
      [{SMS}] [{ALL            }] [{UPDASM }]
```

2.3.3 REJECT and DELETE

```
{REJECT} [{XMS}] [{{EXCEPT}sm-list}] [FORCE] ['comment']
{DELETE} [{UMS}] [{{[ONLY]}       }]
         [{SMS}] [{ALL            }]
```

2.3.4 ACCEPT

```
ACCEPT ['comment']
```

2.3.5 PRTLOG

```
PRTLOG
```

2.3.6 Notes

ALL
specifies that all SMs are to be processed. This is the default.

ASMONLY
specifies that only assemblies and linkedits are to be done.

comment
is a comment, up to 40 characters long, enclosed in apostrophes, to be added to the log entry.

EXCEPT
specifies that all SMs, except those specified in the accompanying sm-list, are to be processed.

FORCE

specifies that the SMs indicated are to be processed, even if an SM does not meet all requirements for processing (such as that all predecessors have been processed). FORCE must be used to reapply an SM which had been rejected.

for-month

specifies the name (first three characters) of the month associated with the SMs (JAN for January, etc.). This is used to check against the data set name used for SMINPUT.

month-no

specifies the sequential number of the month coded in parentheses (1=January, 12=December) associated with the SMs. This is used to check against the data set name used for SMINPUT.

ONLY

specifies that only the SMs given in the sm-list are to be processed. This is the default if an sm-list is specified.

sm-list

is a sequence of SM numbers in ascending order of the form

```
{number       }[,...,{number       }]
{number-number}[      {number-number}]
```

for example:

```
870
870,872-874,915
900-902,904,910-912
```

There can be no spaces in the sm-list.

SMS

specifies Standard SMs, and is the default.

UMS

specifies User-coded SMs.

UPDASM

specifies that updates, assemblies and linkedits are to be done. This is the default.

UPDONLY

specifies that only updates are to be done.

XMS

specifies Experimental SMs.

2.4 CONTROL PARAMETERS

Certain parameters which are defined for the installation in the Profile Table (load module ASMFPROF, created by the SMPROF macro) may be overridden via input control commands submitted in the job stream with the ASMF command cards.

The format for these control commands is the same as for the ASMF command cards: free-form in columns 1-71, with the parameter name separated from the parameter value by one or more spaces. (The equals sign should not be coded.)

The control parameters specified apply to all the ASMF command cards following in the same job, unless they are themselves overridden by succeeding control parameters.

Parameters of the SMPROF macro beginning with '#' are overridden at execution time via the PARM subparameter of the EXEC JCL statement. (See Appendix A.)

All parameters of the SMPROF macro except those beginning with '#' may be overridden through the input job stream. These are:

ASM
ASMRC
COPY
LKED
LKOP
UPD
ZAP

For example, the following may be used to specify a different Assembler module:

```
ASM IXFOXOO
```

If LKOP is specified, its values must be coded without being enclosed in quotes, and must be separated by commas, with no embedded blanks.

For example:

```
LKOP  LIST,LET,NCAL,REUS
```

Chapter 3

EXECUTION

3.1 PARAMETERS

ASMF is executed via a procedure with the name INTASMF. The procedure as released is given in Appendix B. There are nine procedure symbolic parameters and four program execution parameters that may be coded.

3.1.1 Procedure Symbolic Parameters

The nine procedure symbolic parameters are:

P
: specifies the Intercomm library prefix. The default is INT. Code in single quotes if more than one high level qualifier is used, for example: P-'INT.REL9'.

Q
: specifies the SM library qualifier/suffix. The default is SM.

L
: specifies the production library suffix. The default is LIB.

V
: specifies the volume serial number of the SMINPUT data set.

MON
: specifies the SM input data set's DSN suffix (first three characters of the month name).

REG
: specifies execution region size. The default is 300K.

SYSDA
: specifies the type of system direct access working storage. Default is SYSDA.

TAPE
: specifies the type of system tape unit. Default is 9TRK.

WKSPAC
: specifies the disk work file space allocated for each work data set. The default is '(CYL,(3,1)'.

3.1.2 Program Execution Parameters

Certain ASMF control values specified on the SMPROF macro may be overridden via the PARM field of the EXEC statement for stepname ASMF as keyword parameters.

The SMPROF parameters which may be overridden in this way are: #CT (the maximum number of ASMF SM control cards per SM), #IN (the number of control commands per run), #MD (the maximum number of modules per run), and #SM (the maximum number of SMs per run).

For example, to respecify the maximum number of SMs and modules for a particular run, the following must be coded:

```
// EXEC INTASMF,PARM.ASMF='#SM=300,#MD=600'
```

For a complete description of the meaning of those values, see the SMPROF macro description in Appendix A.

3.2 DATA SETS

The following data sets are required for ASMF processing (assuming P=INT and Q=SM):

- INT.SMLIB--ASMF control data set. (If data set blocksize greater than 6400 (Release 10) or 3200 (Release 9), see SMPROF macro, parameter #MD in Appendix A).
- INT.SMLOG--ASMF activity log (print) data set.
- INT.SYMSM--intermediate library to hold the source code of modified modules.
- INT.MODSM--intermediate library containing the modified load modules.
- INT.SYMLIB and INT.SYMREL--libraries containing current Intercomm source modules
- INT.MODLIB and INT.MODREL--libraries containing current load module versions and ASMF load modules
- SMINPUT--the distributed SM minireel or card-image input for experimental or user SMs.

It is assumed that the installation's Assembler resides on SYS1.LINKLIB. If this is not the case, the load library containing the Assembler must be concatenated to the STEPLIB DD statement for step ASMF.

Chapter 4

APPLYING SYSTEM MODIFICATIONS

4.1 EXPERIMENTAL OR USER SMs

The following steps should be followed to apply experimental or user SMs using ASMF. The procedure symbolic parameter Q=XM should be specified for each step.

1. Load the SMs from SMINPUT into INT.XMLIB using the SELECT command. SMINPUT must contain control records, as explained in Appendix C. The DD statement for SMINPUT must be changed as appropriate to specify the source of SMINPUT.

2. Apply the SMs using the APPLY command.

3. Relink the Intercomm execution load module into a test library and test the SMs by using the modified version of Intercomm.

4. If changes to an SM are needed, back off and delete the SM from INT.XMLIB using the DELETE command. Make the necessary changes to the SM and repeat steps 1 through 3 until the SM is correct.

5. When an experimental SM becomes a standard SM, apply the corresponding standard SM following the steps outlined in Section 4.2.

The ACCEPT command should not be used with experimental or user SMs.

4.2 STANDARD SMs

The following steps should be followed to apply official SMs using ASMF:

1. Request loading of the SMs from the SM tape defined for SMINPUT into INT.SMLIB by using the SELECT command.

2. Apply the SMs using the APPLY command.

3. Relink the Intercomm execution load module into a test library and test the SMs by using the modified version of Intercomm.

4. If necessary, reject SMs causing problems by using the REJECT command.

5. Accept SMs into the user production libraries (SYM/MODLIB) by using the ACCEPT command.

6. Relink Intercomm to create the production version.

7. Scratch and reallocate SMLIB, MODSM and SYMSM to prepare them for the next new batch of SMs.

NOTE: TSO/SPF users updating Intercomm modules via SPF should ensure that the TSO PROFILE parameters AUTONUM, NUMBER, and STATS are set off, as they cause directory entry modification of the Intercomm modules source library.

ASMF modifies directory entries, and expects to find certain information intact.

4.3 SAMPLE JCL

For a SELECT operation for all SMs on an SM tape, use:

```
//       EXEC     INTASMF,MON=month,V=tape-volser
//ASMF.SYSIN       DD  *
SELECT  month
/*
```

Where month is the 3-character month name from the SM tape DSN. See Chapter 2 for SELECT statement syntax when only selected SMs are desired (because some SMs update modules not issued on your SYMREL).

For an APPLY operation to apply the selected SMs (from SMLIB), use:

```
//       EXEC     INTASMF,MON=month,V=tape-volser
//ASMF.SYSIN       DD  *
APPLY
/*
```

All selected SMs (those put on SMLIB data set) will be applied and the updated modules will be assembled and linked (SYMSM/MODSM data sets). At this time, determine (from list of updated modules/macros) whether user system tables on SYM/MODUSR need be reassembled due to macro changes. Use:

```
//       EXEC     ASMPCL,Q=USR,U=SM,NAME=table-name,LMOD=table-name
```

so that the modified macro version on SYMSM is used. Add a

```
//ASM.SYSIN       DD  DSN=INT.SYMREL(table-name),DISP=SHR
```

statement for tables that COPY user entries such as BTVRBTB. If an updated version of the base Intercomm table is on SYMSM or SYMLIB, change the SYMREL data set name accordingly.

Chapter 5

ASMF INSTALLATION PROCEDURES

A SM tape supplied for ASMF installation contains one data set following the two SM files: INT.MODASMF.

INT.SYMREL provides the following source members:

INTASMF--the ASMF JCL procedure
SMLEVEL--an ASMF macro (define system release/SM level)
SMPROF--an ASMF macro (define user installation profile)
PROFILE--sample source member, showing use of SMPROF macro
SMS--sample source member, showing use of SMLEVEL macro
XMS--sample source member, showing use of SMLEVEL macro

INT.MODASMF contains the following load modules:

SMDELET
SMCOPY
SMSTOW
SMSTAT
SMREJECT
SMMAIN
SMAPPLY
SMACCEPT
ASMFPROF
GETMEM
INSORT
INTASMF (executed ASMF module)
LOGIT
PRTLOG
PUTSSI
READD
IAIM.... (cross-reference listing utilities--see Appendix F)

To install ASMF, the following steps are recommended:

1. If not done at Release 9 installation time, use IEBCOPY, as illustrated below, to copy the INTASMF procedure from SYMREL into the installation's procedure library (usually SYS1.PROCLIB)

```
//  EXEC     PGM-IEBCOPY
//SYSPRINT  DD      SYSOUT-A
//IN        DD      DSN-INT.SYMREL,DISP-SHR
//OUT       DD      DSN-installation-proclib-name,
//                  other dd statement parms as required
//SYSUT3    DD      UNIT-SYSDA,DISP-(,DELETE),SPACE-(TRK,(1,1))
//SYSIN     DD      *
    COPY I-IN,O-OUT
    SELECT MEMBER-INTASMF
/*
```

2. Use IEBCOPY, as illustrated below, to copy INT.MODASMF into MODREL. ASMF modules are supplied on the SM tape (file 3) in load module form. No further linkedits are required.

```
//            EXEC     PGM=IEBCOPY
//SYSPRINT    DD       SYSOUT=A
//SYSUT3      DD       UNIT=SYSDA,SPACE=(TRK,(5,5))
//SYSUT4      DD       UNIT=SYSDA,SPACE=(TRK,(5,5))
//IN          DD       DSN=INT.MODASMF,LABEL=(3,SL),DISP=OLD,
//                     UNIT=unit-name,DCB=DEN=3,
//                     VOL=SER=volser-of-minireel
//OUT         DD       DSN=INT.MODREL,DISP=OLD
//SYSIN       DD       *
    COPY I=((IN,R)),O=OUT
/*
```

3. Under MVS, the library containing the ASMF load modules must be authorized. The modules INTASMF, SMAPPLY, SMREJECT and SMCOPY dynamically invoke system programs such as IEBUPDTE, the Assembler, Linkage Editor and IEBCOPY. Therefore, INT.MODASMF must be authorized. This is accomplished by placing the name of the MODASMF library in the IBM system member IEAAPF00 on SYS1.PARMLIB. Use IEBCOPY (as illustrated above) to unload and create a new PDS called INT.MODASMF. Change the OUT DD statement to allocate and specify INT.MODASMF, for example:

```
//OUT         DD       DSN=INT.MODASMF,DISP=(NEW,CATLG,DELETE),
//                     UNIT=SYSDA,SPACE=(TRK,(15,1,5)),
//                     VOL=SER=volser,
//                     DCB=(DSORG=PO,RECFM=U,BLKSIZE=6160)
```

Note that ASMF modules are all linked with the REUS attribute, not RENT. Use only the authorized MODASMF data set for STEPLIB in the INTASMF procedure. IAIM... modules (and GETMEM) do not have to be executed from an authorized library and should be on MODREL.

In addition, the main load module INTASMF must be assigned the authorization code via a relink to MODASMF as follows:

```
//            EXEC     LKEDP,Q=ASMF,
//                     PARM.LKED='LIST,LET,REUS,SIZE=(256K,100K)'
//LKED.SYSIN DD *
    INCLUDE            SYSLMOD(INTASMF)
    SETCODE            AC(1)
    NAME               INTASMF(R)
/*
```

Further information on authorized libraries may be found in IBM's OS/VS2 System Programming Library: Supervisor or MVS/XA SPL: System Macros and Facilities, Vol.1.

The following additional steps should be performed to execute (reuse) ASMF:

1. Using CREATEGF, allocate and initialize INT.SMLOG, the ASMF Standard SM log data set and INT.XMLOG, the ASMF Experimental/ User SM log data set. BLKSIZE=3152 is recommended for 3330 disks. One 3330 track should hold about 200 log entries. SPACE=(CYL,(1)) is a reasonable allocation.

2. Assemble and linkedit the ASMFPROF module using the SMPROF macro, (see Appendix A), and place the load module on MODREL, or MODASMF if used.

3. Assemble and linkedit the SMS module (reflecting the current level of standard SMs of the installed Intercomm) using the SMLEVEL macro (see Appendix A). A current member is supplied on MODREL of an Intercomm release tape and automatically updated by INTASMF execution.

4. Assemble and linkedit the XMS module (reflecting the current level of experimental SMs of the installed Intercomm using the SMLEVEL macro. (See Appendix A.)

5. Change STEPLIB and other DD statements, as required, for the installed INTASMF procedure (see Sections 3.1 and 3.2 and Appendix B). If an authorized MODASMF is used for STEPLIB, do not concatenate MODLIB or MODREL with it.

6. The current XMS and SMS load modules must reside in one of the libraries defined for STEPLIB for step ASMF in the INTASMF procedure. For the first execution of ASMF after installing an Intercomm release tape, copy SMS from MODREL to MODASMF, if necessary. For subsequent executions, copy SMS from MODLIB (where the updated member is stored by ACCEPT step of ASMF execution - see Chapter 1).

7. Allocate INT.SMLIB, the ASMF Standard SM control data set and INT.XMLIB, the ASMF Experimental/User SM control data set. Specify DCB=(RECFM=FB,LRECL=80,BLKSIZE=1280,DSORG=PO). A reasonable allocation for 300 SMs on a 3330 disk is SPACE=(CYL,(3,1,50)). If #CT (specified via SMPROF or via PARM.ASMF at execution) is greater than 16, BLKSIZE should be respecified as at least #CT x 80. The space allocation for INT.XMLIB can be appropriately smaller.

8. Allocate four ASMF intermediate test library data sets as follows:

```
//EXEC  PGM-IEFBR14

//SYMSM  DD  DSN-INT.SYMSM,SPACE-(CYL,(10,2,25)),
//           DCB-INT.SYMLIB...

//MODSM  DD  DSN-INT.MODSM,SPACE-(CYL,(5,1,25)),
//           DCB-INT.MODLIB...

//SYMXM  DD  DSN-INT.SYMXM,SPACE-(CYL,(1,1,25)),
//           DCB-INT.SYMLIB...

//MODXM  DD  DSN-INT.MODXM,SPACE-(CYL,(1,1,25)),
//           DCB-INT.MODLIB...
```

Appendix A

ASMF MACROS

The following macros are documented in this appendix:

- SMLEVEL--create the SMS, XMS, and/or UMS modules, specifying SM installation levels

- **SMPROF--create the Profile Table (load module ASMFPROF), specifying installation-dependent systemwide parameters and constants.**

SMLEVEL--Create the SMS or XMS Module

The SMLEVEL macro is used to create the SMS, XMS and UMS modules. Each must be done separately and need only be done for the installation of ASMF. Each will be automatically updated by ASMF as maintenance is performed. The syntax is identical for each.

The form of the SMLEVEL macro is as follows:

```
symbol     SMLEVEL      [{APPLY }=(sm-number[,...,sm-number])]
                        [{NAPPLY}                            ]

                        [,BASE={lowest-sm-number}]
                        [      {1497            }]

                        [,LAST={highest-sm-number}]
                        [      {1497             }]

                        [,GEN={NO }]
                        [     {YES}]

                        [,RELEASE={release-number}]
                        [         {0900          }]
```

symbol must be SMS or XMS or UMS.

APPLY

specifies a list of SM (XM/UM) numbers (between BASE and LAST) which have been applied. If the list exceeds assembler macro limits, use more than one SMLEVEL macro. If more than one SMLEVEL macro is used to create either the XMS, UMS or the SMS module, they must either all use APPLY or all use NAPPLY. APPLY should be used if more SMs have not been applied than have been applied; NAPPLY should be used if more SMs have been applied than have not been applied.

BASE

specifies the lowest SM (XM/UM) number applied at this installation or as specified on the latest Intercomm release tape. For SMS, this number must be no lower than 1497. The default is 1497 for Release 9.0, 1730 for Release 10.0. It is assumed that all SMs below the BASE level have been applied to the SYMREL and MODREL libraries. All new SMs <u>accepted</u> between BASE and LAST are assumed to reside on SYMLIB and MODLIB.

GEN

specifies whether or not this is the only or last SMLEVEL macro to be coded for this module. NO indicates that another SMLEVEL macro follows. The default is YES.

LAST

specifies the highest SM (XM/UM) number already applied at this installation. It must be greater than or equal to the BASE and must not be greater than the lowest number of the SM that you are about to apply. The default is 1497 for Release 9.0, 1730 for Release 10.0.

NAPPLY

specifies a list of SM numbers (between BASE and LAST) which have not been applied. If more than one SMLEVEL macro is used to create either the XMS, UMS or the SMS module, they must either all use APPLY or all use NAPPLY. APPLY should be used if more SMs have not been applied than have been applied; NAPPLY should be used if more SMs have been applied than have not been applied.

RELEASE

specifies the Intercomm release, as a four-digit number. The default is 0900 for Release 9.0, 1000 for Release 10.

SMPROF--Create the PROFILE Table

The SMPROF macro is used to create the Profile Table (load module ASMFPROF), which specifies installation constants and parameters.

The four parameters beginning with # may be overridden at INTASMF execution time via the PARM subparameter of the EXEC JCL statement. (See Section 3.1.2.) The others can be overridden as control parameters submitted in the SYSIN input stream along with ASMF command cards. (See Section 2.4.)

The form of the SMPROF macro is as follows:

```
 [symbol]   SMPROF    [#CT={max-num-command-cards-per-sm}]
                      [    {16                          }]

                      [,#IN={max-num-control-commands-per-run}]
                      [     {16                              }]

                      [,#MD={max-num-modules-per-run}]
                      [     {400                    }]

                      [,#SM={max-num-sms-per-run}]
                      [     {150                }]

                      [,ASM={assembler-module-name}]
                      [     {ASMBLR               }]

                      [,ASMRC={max-successful-return-code}]
                      [       {0                         }]

                      [,COPY={copy-utility-module-name}]
                      [      {IEBCOPY                 }]

                      [,LKED={linkage-editor-module-name}]
                      [      {LINKEDIT                  }]

                      [,LKOP={'link-edit-options'}]
                      [      {'LIST,XREF,LET,NCAL'}]

                      [,UPD={source-update-utility-module-name}]
                      [     {IEBUPDTE                         }]

                      [,ZAP={superzap-utility-module-name}]
                      [     {IMASPZAP                    }]
```

#CT

specifies the maximum number of ASMF command cards per SM. The default is 16. This value may be overridden at execution.

#IN

specifies the maximum number of control parameter commands per run. The default is 16. This value may be overridden at execution.

#MD

specifies the maximum number of modules (Csects/Macros/Dsects) to be updated per run; also specifies the blocksize of SMLIB divided by 16. If SMLIB blocksize/16 is greater than the default, then this parameter must be overridden at execution time (see example in section 3.1.2). The default is 400 for Release 10 (200 for Release 9). This value may be overridden at execution.

#SM

specifies the maximum number of SMs (SMs/XMs/UMs) per run. The default is 150. This value may be overridden at execution.

ASM

specifies the name of the Assembler module (IEV90, IXFOX00, etc.). The default is ASMBLR.

ASMRC

specifies the highest assembly return code to be considered successful. The default is 0.

COPY

specifies the name of the copy utility module. The default is IEBCOPY.

LKED

specifies the name of the Linkage Editor module. The default is LINKEDIT.

LKOP

specifies linkedit options, enclosed in apostrophes. The default is 'LIST,XREF,LET,NCAL'. MVS users should then relinkedit all LPA- eligible modules as RENT (or place control parameter overrides in the input stream for the APPLY process).

UPD

specifies the name of the source update utility module. The default is IEBUPDTE.

ZAP

specifies the name of the Superzap utility module. The default is IMASPZAP.

Appendix B

INTASMF PROCEDURE

The following is the INTASMF cataloged procedure as supplied. The user may modify it to fit installation requirements as necessary.

```
//**********************************************************************
//*                                                                    *
//***INTERCOMM****AUTOMATED-SM-MANAGEMENT-FACILITY PROCEDURE*************
//*                                                                    *
//**********************************************************************
//INTASMF  PROC P=INT,          *FILES STANDARD HIGH LEVEL INDEX       *
//              Q=SM,           *WORK LIBRARIES QUALIFIER              *
//              L=LIB,          *PRODUCTION LIBRARIES QUALIFIER        *
//              SYSDA=SYSDA,    *SYSTEM DIRECT ACCESS WORKING STORAGE  *
//              WKSPAC='(CYL,(3,1))',    *WORK FILES SPACE ALLOCATION  *
//              TAPE=9TRK,      *9 TRACK TAPE UNIT STANDARD NAME       *
//              V=VOLSER,       *SM INPUT TAPE VOLUME SERIAL NUMBER    *
//              MON=MON,        *SM BATCH MONTH (3 CHARACTERS)         *
//              REG=300K        *REGION SIZE                           *
//*                                                                    *
//*                                                                    *
//*                                                                    *
//**********************************************************************
//*   STEP 1 - TEMPORARY DATA SET ALLOCATION                           *
//**********************************************************************
//*                                                                    *
//S1       EXEC PGM=IEFBR14
//ASMPUNCH DD DSN=&&SMPUN,SPACE=&WKSPAC,UNIT=&SYSDA,DISP=(,PASS)
//SSIOUT   DD DSN=&&SMSSI,SPACE=(TRK,(1)),UNIT=&SYSDA,DISP=(,PASS)
//*                                                                    *
//*                                                                    *
//*                                                                    *
//**********************************************************************
//*   STEP 2 - ASMF EXECUTION                                          *
//**********************************************************************
//*                                                                    *
//ASMF     EXEC PGM=INTASMF,REGION=&REG
//STEPLIB  DD DSN=&P..MODEL,DISP=SHR => MODASMF .
//         ~~DD DSN=&P..MODREL,DISP=SHR~~
//*
//*
//*
//SMLIB    DD DSN=&P..&Q.LIB,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=960,LRECL=96)
//SMSYSOT  DD SYSOUT=A,DCB=(RECFM=FBA,BLKSIZE=810,LRECL=81)
//LISTOUT  DD SYSOUT=A,DCB=BLKSIZE=121          ***USER MAY OVERRIDE***
//LOGPRINT DD SYSOUT=A
```

(continued)

```
//SMLOG     DD  DSN=&P..&Q.LOG,DISP=SHR
//COPYUT3   DD  UNIT=&SYSDA,SPACE=&WKSPAC
//COPYUT4   DD  UNIT=&SYSDA,SPACE=&WKSPAC
//UPDSYSIN  DD  DSN=&P..&Q.LIB,DISP=SHR
//LIBUT1    DD  DSN=&P..SYM&Q,DISP=SHR
//          DD  DSN=&P..SYM&L,DISP=SHR
//          DD  DSN=&P..SYMREL,DISP=SHR
//LIBUT2    DD  DSN=&P..SYM&Q,DISP=SHR
//ASMLIB    DD  DSN=&P..SYM&Q,DISP=SHR
//          DD  DSN=&P..SYM&L,DISP=SHR
//          DD  DSN=&P..SYMREL,DISP=SHR
//          DD  DSN=SYS1.MACLIB,DISP=SHR
//ASMUT1    DD  UNIT=&SYSDA,SPACE=&WKSPAC                      }
//ASMUT2    DD  UNIT=(&SYSDA,SEP=(ASMUT1)),SPACE=&WKSPAC        } See note
//ASMUT3    DD  UNIT=(&SYSDA,SEP=(ASMUT1,ASMUT2)),SPACE=&WKSPAC }
//SSIOUT    DD  DSN=*.S1.SSIOUT,VOL=REF=*.S1.SSIOUT,DISP=(OLD,DELETE)
//ASMPUNCH  DD  DSN=*.S1.ASMPUNCH,VOL=REF=*.S1.ASMPUNCH,DISP=(OLD,DELETE)
//LNKSYSN   DD  DSN=*.S1.SSIOUT,VOL=REF=*.S1.SSIOUT,DISP=(OLD,DELETE)
//          DD  DSN=*.S1.ASMPUNCH,VOL=REF=*.S1.ASMPUNCH,DISP=(OLD,DELETE)
//LNKSLIB   DD  DSN=&P..MOD&Q,DISP=SHR
//          DD  DSN=&P..MOD&L,DISP=SHR
//          DD  DSN=&P..MODREL,DISP=SHR
//LNKSLIB0  DD  DSN=&P..MOD&Q,DISP=SHR
//LNKSLIB1  DD  DSN=&P..MOD&L,DISP=SHR
//LNKSLIB2  DD  DSN=&P..MODREL,DISP=SHR
//LNKSYSLM  DD  DSN=&P..MOD&Q,DISP=SHR
//LNKUT1    DD  UNIT=&SYSDA,SPACE=&WKSPAC                      } See note
//ASMSYSN   DD  DSN=&P..SYM&Q,DISP=SHR
//          DD  DSN=&P..SYM&L,DISP=SHR
//          DD  DSN=&P..SYMREL,DISP=SHR
//ASMSYSN0  DD  DSN=&P..SYM&Q,DISP=SHR
//ASMSYSN1  DD  DSN=&P..SYM&L,DISP=SHR
//ASMSYSN2  DD  DSN=&P..SYMREL,DISP=SHR
//SYSUDUMP  DD  SYSOUT=A
//SMINPUT   DD  DSN=INT.SM.PUNCH.&MON,DISP=(OLD,PASS),
//              VOL=(PRIVATE,RETAIN,SER=&V),
//              LABEL=(2,SL),UNIT=(&TAPE,,DEFER)
```

<u>NOTE</u>: ASMUT1, ASMUT2, ASMUT3 and LNKUT1 are eligible for VIO, which offers possible performance improvements.

For execution of this procedure under MVS, the STEPLIB library must be authorized (see Chapter 5, installation step 3), and

```
//          DD  DSN=SYS1.AMODGEN,DISP=SHR
```

must be added to the concatenation for ASMLIB (after DD statement for SYS1.MACLIB).

Data set names for ddnames ASMSYSN0-ASMSYSN2 must be of the format x.SYMyyy, where 'x' may be any number of high-level qualifiers (defaults to INT), and 'yyy' is the suffix of the dsname lowest-level qualifier. The data set name must not exceed 44 characters, including periods. LIBUT1 SYSN0-1, -2 must [illegible] YSN3 etc

Appendix C

SM CONTROL RECORDS

The data set SMINPUT must contain SM control information required for ASMF. This is automatically supplied (on File 2 of the SM tape) for Standard SMs but must be coded by the user for XMs and UMs. This is in the form of 80-byte control records, as follows:

Bytes	Contents	Description
1-2	++	Identifies SM control record (See NOTE)
3-5	ttt	Type of SM control record:
	SM	SM Declaration
	UM	User SM
	XM	Experimental SM
	PRE	Prerequisite SM list
	ASM	Reassembly list
	MOD	Module Header
6-7	blank	
8-80	operands	Operands vary depending on the type of SM control record. Operands are fixed-length and separated by one blank.

<u>NOTE</u>: SM control records starting with "++" should not be put on a PANVALET data set, where "++" has a special significance.

Each SM begins with the SM Declaration, as follows:

```
123... 8     13    18     23

++SM   nnnn rrrr [ssss] [xxxx]
```

where:

- nnnn is the four-digit SM number
- rrrr is the applicable four-digit Intercomm release number
- ssss is the four-digit Intercomm lower release number (if this SM applies to more than one release level)
- xxxx is the four-digit number of the experimental SM being replaced by this SM, if any.

For example:

```
123... 8    13
++SM   0823 0701
```

denotes SM 823 for release 7.01.

One or more optional Prerequisite SM lists may follow the SM Declaration, as follows:

```
123... 8    13
++PRE  nnnn nnnn ....
```

where nnnn is a four-digit SM number which is a prerequisite for the SM being processed.

One or more optional Reassembly lists may follow:

```
123... 8        17
++ASM  aaaaaaaa aaaaaaaa ...
```

where aaaaaaaa is an eight-character name of one or more modules which must be reassembled after application of the current SM. Each module name is left-justified and padded with blanks.

Each IEBUPDTE deck for a module is immediately preceded by a Module Header, formatted as follows:

```
123... 8        17
++MOD  mmmmmmmm t
```

where:

- mmmmmmmm is the name of the module affected by the subsequent IEBUPDTE deck. The module name is left-justified and padded with blanks.

- t indicates the type of module, as follows:

  C--CSECT

  D--DSECT or COPY code

  M--Macro

  Z--Superzap

NOTE: For users who have changed the source member names of Intercomm modules or tables, and for those using the Multiregion Facility (where multiple versions of the SPALIST, for example, are required), File 2 of the SM tape may be off-loaded to a data set where the user can add/change member names on the Reassembly control card (++ASM), or add more Reassembly control cards, as needed. This new file then becomes the SMINPUT data set (see the dicussion of INTASMF execution in Chapter 3)

Appendix D

PRINTED REPORTS

This appendix gives samples of the two reports which are output by ASMF, the listing of SMs via the PRINTTP command, and the listing of the SM Log via the PRTLOG command.

D.1 SM LISTING

The following is an example of an SM listing issued by ASMF:

```
       SYSTEM MODIFICATION 1 1 7 8

++SM   1178 0800 0700
++MOD  VTLUCMD   C
./     CHANGE NAME=VTLUCMD,SSI=08001178                T=C
         TM     LUBFLAG2,LUBDTOK       DATA TRAFFIC O  K   SM1178 08921000
         BZ     4(R7)                  NO SEND ALLOWED     SM1178 08921100
         DC     C'SM '                                            99999997
         DC     X'08001178'                                       99999998
```

D.2 SM LOG

The following is an example of the SM Log listing issued by ASMF.

```
0001 91800132 IJKPRINT SM REL #   91800134 U X 0700 C
0002 91800132 IJKPRINT SM SM  #   91800134 U X 0701 C
0001 91800132 IJKPRINT SM     #   91800134 A X 0701 C
0001 91800134 IJKPRINT SM                  S X 0701 M
0003 91800134 IJKPRINT SM REL LIB 91800147 U X 0700 C
0004 91800145 ATTRIB   SM REL LIB 91800147 U S 1181 M
0005 91800145 COMMAND  SM REL LIB 91800147 U S 1181 M
0006 91800145 CNTLCHR  SM REL LIB 91800147 U S 1181 M
0007 91800145 MMUED003 SM REL LIB 91800147 U S 1182 C
0002 91800145 MMUED003 SM     LIB 91800147 A S 1182 C
0002 91800145          SM     LIB 91800147 C             NOW ACCEPT SMS
```

The report contains the following:

Field	Contents
1	Sequence number within function
2	Time stamp in the form YDDDHHMM
3	Module name
4	Name of library updated. This is extracted from ASMSYSNO ddcard DSN.
5	Name of library used as input to IEBUPDTE. This is extracted from ASMSYSNO, ASMSYSN1, or ASMSYSN2.
6	Name of library into which the module was ACCEPTed or an # indicating that the module was scratched. Extracted from ASMSYSNO, ASMSYSN1, or ASMSYSN2.
7	Time stamp of when module was accepted or scratched in the form YDDDHHMM.
8	Function code, as follows: U--Update A--Assembly S--Scratched C--Copy Z--Zap
9	Class code, as follows: S--Standard SM X--Experimental SM (XM) U--User-Coded SM (UM)
10	SM, XM or UM number
11	Type code, as follows: C--CSECT D--DSECT M--Macro Z--Superzap
12	Comments from APPLY, REJECT, DELETE, or ACCEPT command card

Appendix E

MESSAGES AND CODES

NOTE: In certain cases messages having different text and different meaning are issued with the same control number (such as SMACCEPT-001). Those messages are indicated in this listing by an asterisk (*) in the left margin which is not actually part of the message.

Messages which are purely informative and do not indicate an error condition are indicated by a bullet (●) in the left margin which is not actually part of the message.

Unless otherwise indicated, current execution continues, and remaining SMs are processed as possible after an error is detected.

BDAM RTN ** BLKSIZE MISSING

Cause: Internal program error. LOG BLKSIZE=0. Abend 65 follows.

Action: Submit MSR with dump.

BDAM RTN ** NEG NO. BYTES PASSED

Cause: Internal program error. READ called with invalid parameter. Abend 65 follows.

Action: Submit MSR with dump.

BDAM RTN ** READ ERR, DCBADDR=xxxxxx, RELBYTE=yyyyyyyy

Cause: Read error: EOF reached on READ operation. Internal program error. Abend 65 follows.

Action: Submit MSR with dump.

BDAM RTN ** REL BYTE 0 INVLD

Cause: Internal program error. READ called with invalid parameter. Abend 65 follows.

Action: Submit MSR with dump.

BDAM RTN ** WRITE ERR, DCBADDR=xxxxxx, RELBYTE=yyyyyyyy

Cause: WRITE ERROR: SMLOG out of space. Abend 65 follows.

Action: Print SMLOG, scratch it, then reallocate and intialize it using CREATEGF. Rerun job.

---

* SMACCEPT -001- ACCEPT PROCESSING COMPLETE WITH ERRORS

Cause: Successful processing of some SMs. IEBCOPY errors prevented processing all SMs. Copy of successful modules from PMI.SYMSM and PMI.MODSM to production libraries completed.

Action: Examine IEBCOPY error messages for correction procedures.

•* SMACCEPT -001- ACCEPT PROCESSING COMPLETE WITH NO ERRORS

Cause: Successful processing of ACCEPT command.

SMAPPLY -001- ERROR IN SMSTATUS ROUTINE - ABORTING

Cause: Error in processing. Other messages listed.

Action: Examine other error messages for correction procedures.

SMAPPLY -002- SMnnnn A PREREQ FOR xxxxxxxx HAS NOT BEEN APPLIED

Cause: Prerequisite SM not applied to system.

Action: If rejected SM is required apply prerequisite SM, or specify FORCE in APPLY command, and run again. Only the rejected SMs need be reapplied.

• SMAPPLY -003- SMnnnnnn WILL BE FORCE APPLIED

Cause: Prerequisite SM not applied to system, but FORCE was specified on APPLY command. The SM is FORCE applied. This message is always preceded by SMAPPLY-002.

SMAPPLY -004- xxxxxxxx NOT FOUND IN SMLIB CANNOT APPLY

Cause: Error in ASMF control data set. SMINPUT missing control records.

Action: If SMINPUT was card-image, check for missing cards. If SMINPUT was tape, error is in SM release tape.

---

* SMAPPLY -005- SMnnnnnn xxxxxxxx NOT FOUND CANNOT APPLY

Cause: Module to which specified SM applies cannot be found. Possibly because the module was to be added by an earlier SM that was not applied, or module is part of a special feature not purchased by the user.

Action: Correct error condition and run again for rejected SM or SMs, or ignore.

----

* SMAPPLY -005- SMnnnnnn xxxxxxxx WAS FOUND CANNOT ADD

Cause: A module that was to be added via SMnnnn was found to already exist on SYMSM. Possibly because the SM was applied twice, or the module was added by an XM or UM.

Action: Determine why the module already was on the library. Correct condition and rerun job for rejected SM or SMs as necessary.

SMAPPLY -006- SMnnnnnn xxxxxxxx HAS USER/XM MODS MUST FORCE APPLY

Cause: Module xxxxxxxx was found to have user modifications or experimental SMs. This condition is detected only if UMs or XMs were applied via ASMF. The SM is not applied unless FORCE is specified on APPLY command card.

Action: Rerun job to FORCE apply rejected SM or SMs if necessary, or apply rejected SM to a version of the module that does not contain experimental SMs.

● SMAPPLY -007- SMnnnnnn WILL BE FORCE APPLIED

Cause: A module, to which SMnnnn is to be applied, contains user or experimental SMs but FORCE was specified on APPLY command card. This message is always preceded by message SMAPPLY-006. The SM is FORCE applied.

● SMAPPLY -008- SMnnnnnn xxxxxxxx UPDATE SUCCESSFUL

Cause: The SM specified has been successfully applied to the module specified.

SMAPPLY -009- SMnnnnnn STOW ERROR FOR XM/USER MOD

Cause: 1. No space left in directory of PMI.SYMSM data set.

2. Member name to be replaced not found on PMI.SYMSM.

3. Permanent I/O error on PMI.SYMSM.

4. Insufficient virtual storage.

Action: 1. Reallocate PMI.SYMSM and increase the number of directory blocks, rerun the job.

2. Determine why the member name is missing, correct and rerun the job.

3. Determine cause of I/O error from IBM data management messages.

4. Rerun job specifying larger region size.

SMAPPLY -010- SMnnnnnn-xxxxxxxx - MODULE NOT FOUND

Cause: Module to be zapped by the specified SM was not found on PMI.MODSM, PMI.MODREL or PMI.MODLIB.

Action: Correct error condition and run again to apply rejected SM or SMs.

• SMAPPLY -011- SMnnnnnn xxxxxxxx SUPERZAP SUCESSFUL

Cause: Superzap SMnnnn has been successfully applied to module xxxxxxxx.

SMAPPLY -012- MODULE TABLE FULL- ABORTING

Cause: #MD too small.

Action: Increase #MD via PARM.INTASMF and run again.

SMAPPLY -013- xxxxxxxx NOT FOUND IN LIBRARY CANNOT ASSEMBLE

Cause: Module xxxxxxxx not found on DDNAME LIBUT1.

Action: Correct error condition and run again for rejected SMs.

•* SMAPPLY -016- ASSEMBLY OF xxxxxxxx SUCCESSFUL

Cause: The assembly of the specified module was successful.

●* SMAPPLY -016- LINKEDIT OF xxxxxxxx SUCCESSFUL

Cause: The linkedit of the specified module was successful.

SMAPPLY -017- SM nnnn NOT FOUND

Cause: The specified SM was not found. May not have been SELECTed first.

Action: Correct error condition and run again for rejected SMs.

SMAPPLY -018- SM nnnn: RC xx FROM mmmmmmmm MOD=xxxxxxxx

Cause: A return code of xx was received from the assembler or linkage editor named mmmmmmmm for module xxxxxxxx. Check the assembly or linkedit listing to determine the nature of the errors.

Action: Correct the errors or override ASMRC in the command input stream, then rerun the job to reapply the SM.

● SMAPPLY -019- SMnnnn [SMnnnn...SMnnnn]

Cause: Follows SMAPPLY-016: Assembly of xxxxxxxx Successful. Gives list of SMs that were applied to the module.

SMCOPY -001- MODULE xxxxxxxx DDNAME=dddddddd NOT FOUND

Cause: Module not found in data set specified by DDNAME.

Action: Correct error condition and run again for rejected SM.

SMCOPY -002- ERROR FROM IEBCOPY

Cause: Errors occured during a copy operation.

Action: Check IEBCOPY output to determine the nature of the errors; correct and run again.

SMDELET -001- SMnnnnn NOT FOUND CANNOT DELETE

Cause: SM specified not found on SMLIB control data set.

Action: Correct error condition and run again.

SMDELET -002- SMnnnnnn ERROR- INDEX RECORD FOR mmmmmmmm NOT FOUND

Cause: Part of SM specified cannot be found for deletion.

Action: If SMINPUT was card-image, check for missing or out-of-sequence cards. If SMINPUT was tape, error in SM release tape.

●* SMDELET -004- SMnnnnnn CONTROL RECORD FOR mmmmmmmm DELETED

Cause: Successful deletion.

---

* SMDELET -004- SMnnnnnn CONTROL RECORD FOR mmmmmmmm I/O ERROR

Cause: Permanent I/O error on SMLIB or insufficient virtual storage.

Action: Check IBM data management messages to determine cause, or increase region size and rerun job.

---

* SMDELET -004- SMnnnnnn CONTROL RECORD FOR mmmmmmmm NOT FOUND

Cause: Record not found. Possible user error, such as two deletes for same SM.

Action: Determine cause of missing control record.

SMMAIN -001- OPEN ERROR SYSIN DSN- RUN CANCELED

Cause: SYSIN could not be opened. Current execution is terminated.

Action: Correct error condition and run again.

SMMAIN -002- BAD INPUT PARAMETER - RUN ABORTED

Cause: Unidentifiable parameter specified via PARM.INTASMF. Current execution is terminated.

Action: Correct error condition and run again.

SMMAIN -003- MAX # CONTROL CARDS WONT FIT IN SMLIB BLKSIZE-ABORTING

Cause: #CT too large for SMLIB blocksize. #CT times logical record size cannot exceed blocksize. Current execution terminated.

Action: Increase BLKSIZE or reduce #CT appropriately and run again. (See execution Step 7 in Chapter 5.)

SMMAIN -004- INVALID OPERATION CODE-- RUN ABORTED

Cause: Unidentifiable ASMF command. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -005- UNSUPPORTED OVERRIDE OF PARMS

Cause: Can only override #CT, #IN, #MD, and #SM via PARM.INTASMF. Other SMPROF parameters can be overridden in the job stream with ASMF control commands. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -006- cccccccc- INVALID OPTION - RUN ABORTED

Cause: ASMF command parameter specified was not UPDONLY, ASMONLY, UPDASM, FORCE, or a comment. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -007- SM TABLE FULL -ABORTING

Cause: #SM too small; too many SMs specified via commands. Current execution terminated.

Action: Increase #SM via PARM.INTASMF and run again.

SMMAIN -008- RUN TERMINATED BECAUSE OF CONTROL CARD ERRORS

Cause: One or more syntax errors in ASMF commands. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -009- TOO MANY CONTINUATION CARDS- RUN ABORTED

Cause: #IN too small for number of command parameters specified. Current execution terminated.

Action: Increase #IN via PARM.INTASMF and run again.

SMMAIN -010- INVALID CONTINUATION CARD - RUN ABORTED

Cause: Syntax error in continuation of ASMF command. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -011- INVALID OPERAND - RUN ABORTED

Cause: Error in ASMF command. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -012- SM TABLE SIZE EXCEEDED - RUN ABORTED

Cause: #SM too small; too many SMs on ASMF control data set. Current execution terminated.

Action: Increase #SM via PARM.INTASMF and run again.

SMMAIN -013- MONTH NAME INVALID - RUN ABORTED

Cause: Unrecognizable for-month name in SELECT or PRINTTP command. Current execution terminated.

Action: Correct error condition and run again.

SMMAIN -014- INVALID MONTH NUMBER - RUN ABORTED

Cause: Month number not between 1 and 12 or not enclosed in parentheses in SELECT or PRINTTP command. Current execution terminated.

Action: Correct error condition and run again.

SMREJECT -001- MODULE TABLE FULL- ABORTING

Cause: #MD too small. Current execution terminated.

Action: Increase #MD via PARM.INTASMF and run again.

----

* SMREJECT -002- SMnnnnnn LOADMOD OF xxxxxxxx I/O ERROR

Cause: I/O error on LNKSYSLM due to permanent data set I/O error or insufficient region size.

Action: Determine cause of permanent I/O error from IBM data management messages, correct and rerun, or rerun with larger region size.

----

* SMREJECT -002- SMnnnnnn LOADMOD OF xxxxxxxx NOT FOUND

Cause: Load module not found on LNKSYSLM, possibly because it was already rejected, or SMnnnn was never applied.

Action: Determine cause of error. Rerun if necessary.

●* SMREJECT -002- SMnnnnnn LOADMOD OF xxxxxxxx REJECTED

Cause: Load module xxxxxxxx deleted from LNKSYSLM.

---

* SMREJECT -002- SMnnnnnn SOURCE OF xxxxxxxx I/O ERROR

Cause: I/O error on LIBUT2 due to permanent data set I/O error or insufficient region size.

Action: Determine cause of permanent I/O error from IBM data management messages, correct and rerun, or rerun with larger region size.

---

* SMREJECT -002- SMnnnnnn SOURCE OF xxxxxxxx NOT FOUND

Cause: Source module xxxxxxxx not found on LIBUT2, possibly because it was already rejected, or SMnnnnnn was never applied.

Action: Determine cause of error. Rerun if necessary.

●* SMREJECT -002- SMnnnnnn SOURCE OF xxxxxxxx REJECTED

Cause: Source module xxxxxxxx deleted from LIBUT2.

SMREJECT -003- CONTROL RECORD FOR xxxxxxxx NOT FOUND

Cause: Record missing from SMLIB data set. Current execution terminated.

Action: Determine cause of error. (If SM was incorrect when loaded, this error should have been discovered during APPLY processing.) Rerun job to REJECT the SM.

● SMREJECT -004- SMnnnnnn xxxxxxxx UPDATE SUCCESSFUL

Cause: SMnnnnnn has been reapplied to the specified module. It is reapplied because another SM to the module was rejected via ASMF command.

● SMREJECT -005- xxxxxxxx NOT FOUND IN SMLIB CANNOT APPLY

Cause: SM member xxxxxxxx not found in ASMF control data set. The SM was being reapplied because a SM was rejected via ASMF command. (This error should have been detected during APPLY processing.)

SMREJECT -006- SMnnnnnn STOW ERROR FOR XM/USER MOD

Cause: Same as conditions indicated under SMAPPLY-009 except that DDNAME is LIBUT2.

Action: Same as for SMAPPLY-009 except DDNAME=LIBUT2.

SMREJECT -007- SMnnnnnn-xxxxxxxx - MODULE NOT FOUND

Cause: Module xxxxxxxx to which specified SM applies not found on either LNKSYSLM or LNKSLIB.

Action: Correct error condition and run again. Rerun job to ensure all non-rejected SMs are reapplied.

● SMREJECT -008- SMnnnn xxxxxxxx SUPERZAP SUCCESSFUL

Cause: Successful reapplication of SMnnnn to module xxxxxxxx.

●* SMREJECT -009- ASSEMBLY OF xxxxxxxx SUCCESSFUL

Cause: Successful assembly after a reapplied SM.

●* SMREJECT -009- LINKEDIT OF xxxxxxxx SUCCESSFUL

Cause: Successful linkedit after a reapplied SM.

SMREJECT -010- SM nnnn NOT FOUND

Cause: An SM to be reapplied was not found.

Action: Correct error condition and run again if necessary, to reapply SMs not previously rejected. The SM may not have been originally selected.

SMREJECT -011- SM nnnn: RC xx FROM mmmmmmmm

Cause: An unsuccessful return code of xx from assembler or linkage editor module mmmmmmmm.

Action: Determine the nature of the errors from linkage editor or assembler output. Correct the errors or override the ASMRC value via the control command input stream. Rerun the job to reapply the specified SM.

SMSTAT -001- UNABLE TO COPY SM STATUS MODULE- RUN CANCELLED

Cause: SMS or XMS module could not be copied by IEBCOPY. Current execution terminated.

Action: Make sure SMS or XMS module is present on LNKSLIB or LNKSYSLM. Check IEBCOPY output for error messages.

SMSTAT -002- SM NUMBER nnnn EXCEEDS CAPACITY OF SM STATUS TABLE- RUN CANCELED

Cause: SMS or XMS module full and cannot be updated. Current run is terminated.

Action: Scratch and reallocate the SYMSM and MODSM data sets. Then reassemble and link SMS or XMS module specifying BASE and LEVEL on SMLEVEL macro as the number of the last SM applied prior to the current batch. Then rerun the job from the beginning.

SMSTOW -001- UNABLE TO OPEN SM PDS

Cause: SMLIB data set could not be opened. The current run is terminated.

Action: Check JCL for correct parameters. Check VTOC listing for data set; ensure that allocation was successful and DCB parameters are correct. Rerun the job.

SMSTOW -002- CANNOT HAVE MIXED TYPES IN SAME RUN- SM, XM, OR UM

Cause: ASMF commands specifying more than one type of SM were found in the same run, or command card and SM control record do not reference same SM type. Current run is terminated.

Action: Execute one run for each type of SM.

● SMSTOW -003- SMnnnn DOES NOT APPLY TO REL xxxx-FORCE SELECTED

Cause: Because FORCE was specified, the SM specified has been processed although it is not appropriate for this release.

● SMSTOW -004- SMnnnn DOES NOT APPLY TO REL xxxx - EXCLUDED

Cause: The SM specified has not been processed because it is not appropriate for this release.

SMSTOW -005- SM NUMBERS OUT OF SEQUENCE

Cause: Self-explanatory. Current run is terminated.

Action: Respecify SM numbers and control records in numerical sequence.

SMSTOW -006- ccc CARD AFTER MOD CARD ABORTING RUN

Cause: Error in SM control records in SMINPUT.

Action: If SMINPUT was card-image, check for an out-of-sequence card. If SMINPUT was tape, error in SM release tape.

SMSTOW -007- SM CONTROL CARD WORK TABLE FULL - ABORTING

Cause: SMLIB blocksize too small. Current run is terminated. Blocksize must be greater than or equal to #CT times logical record length.

Action: Reallocate SMLIB specifying larger blocksize and rerun, (see Chapter 5).

SMSTOW -008- ccc INVALID ++ OPERATION - IGNORED

Cause: Unrecognizable SM control record in SMINPUT.

Action: If SMINPUT was card-image, check for mispunched card, correct error and rerun job. If SMINPUT was tape, error in SM release tape.

SMSTOW -009- BLDL- PERM I/O ERROR IN SOURCE LIBRARY

Cause: Permanent I/O error on LIBUT1, or insufficient virtual storage. Current run is terminated.

Action: Check IBM data management messages to determine cause of error. Also, check JCL and VTOC listing to ensure allocation specifications are correct, or rerun job with larger region size.

SMSTOW -010- SMnnnn-xxxxxxxx NOT FOUND IN LIBRARY

Cause: Module not found on LIBUT1. Possibly the module was added via an earlier SM that was not applied, or is part of a special feature not purchased by the user.

Action: Determine cause of missing module. Rerun the job to reapply the SM if necessary.

SMSTOW -011- MODULE TABLE FULL - RUN ABORTED

Cause: #MD too small. Current run is terminated.

Action: Increase #MD via PARM.INTASMF and run again.

---

* SMSTOW -012- xxxxxxxx CONTAINS USER MODS MUST FORCE APPLY SMnnnn

Cause: Module xxxxxxxx contains user SMs. The SM will not be applied. If desired, rerun the job specifying FORCE on ASMF command card to apply this SM.

---

* SMSTOW -012- xxxxxxxx CONTAINS EXP MODS MUST FORCE APPLY SMnnnn

Cause: Module xxxxxxxx contains experimental SMs. The SM will not be applied.

Action: If desired, rerun the job specifying FORCE on ASMF command card to apply this SM.

SMSTOW -013- BLDL- PERM I/O ERROR ON SMLIB

Cause: I/O error on SMLIB, or insufficient virtual storage. Current run is terminated.

Action: Examine IBM data management messages to determine the nature of the I/O error. Also check JCL and VTOC listing to ensure valid parameter, or rerun job with larger region size.

SMSTOW -014- SM LIBRARY DIRECTORY FULL

Cause: SMLIB out of directory blocks. Current run is terminated.

Action: Reallocate SMLIB, with a larger number of directory blocks. Rerun the job.

SMSTOW -015- STOW- PERM I/O ERROR

Cause: Permanent I/O error on data set SMLIB. Current run is terminated.

Action: Check IBM data management messages to determine cause of error. Also check JCL and VTOC listing and ensure all specifications are correct.

SMSTOW -016- COULD NOT OPEN TO READ TAPE LABEL

Cause: Open on SMINPUT failed. Current run is terminated.

Action: Check JCL on SMINPUT DD statement; verify parameters are correct.

SMSTOW -017- INVALID TAPE LABEL

Cause: Self-explanatory. Current run is terminated via ABEND CC=001. The SM release tape is bad.

Action: A new copy should be obtained.

● SMSTOW -018- DSNAME MONTH NE CNTRL CARD MONTH, USE CARD

Cause: Month specified on SMINPUT dsname not the same as specified on ASMF command card. Current run continues.

Action: ASMF makes required adjustments.

● SMSTOW -019- CODED MONTH NE TAPE LABEL MONTH, USE LABEL

Cause: Month specified on SMINPUT tape label not the same as specified on ASMF command card. Current run continues. ASMF takes the month from the SM release tape label.

SMSTOW -020- SYSTEM MODIFICATION xxxxxxxx ALREADY IN DIRECTORY, IGNORED

Cause: Self-explanatory. Possibly, the same SM was applied twice.

Action: Determine why the member was already in the library. Printing the contents of the member may aid diagnosis of the problem.

SMSTOW -021- SM nnnn ALREADY SELECTED - IGNORED

Cause: Self-explanatory. Possibly, a duplicate SELECT card for the same SM was specified.

Action: Determine why the member was already in the library (SM already applied). Printing the contents of the member may aid diagnosis of the problem. If SM selected is not identical to one which already exists, use DELETE and SELECT to replace the existing one with the new one.

Appendix F

CROSS-REFERENCE PROGRAMS

F.1 Introduction

This appendix documents the Intercomm cross-reference modules, as follows:

- IAIMCOCR - Scans a library for COPY statements and prints a sorted cross-reference listing by COPY member name, showing modules which copy that member into those modules.

- IAIMGOCR - Accepts only eighty-character source code as input and produces (as input for a sort) a data set containing a list of global symbols with names of modules referring to them, and indicating at what sequence number in the module (macro) the reference is made.

- IAIMGOC2 - Prints a global symbol cross-reference listing after the output from IAIMGOCR has been sorted.

- IAIMMOCR - Scans a library for modules that refer to specific or all macros and produces a sorted cross-reference listing giving names of all modules referring to them.

- IAIMMRF1 - Lists one or more macros, or an entire macro library with a cross-reference list of macro sequence symbols and globals following each macro. Input is eighty-character source code.

- IAIMMRF2 - Produces a cross-reference list of macro sequence symbols and globals from an input of 121-character (Assembly print output) records.

- IAIMOPCD - Scans a source library and produces a sorted directory listing, followed by a sorted, detailed 'op-code' occurrence cross-reference listing.

- IAIMXRF1 - Scans load modules and analyzes them for internal CSECT size, external references and entry points. Output is then sorted (in separate steps) and passed to IAIMXRF2 for printing.

- IAIMXRF2 - Prints an ordered core usage list, followed by a list of external references and entry points with CSECTs and members containing them. This information is produced by IAIMXRF1 and sorted by SORT before being passed to this program for printing.

F.2 IAIMCOCR--Copy Member Occurrences

This program generates an alphabetic list of all copy members that are referenced and the modules that reference them. The input data set (SYSUT1) is a card-image, Assembler Language source library. The copy members themselves need not appear in the input library.

Sample JCL:

```
//IAIMCOCR JOB
//STEP1    EXEC PGM=IAIMCOCR,REGION=100K
//STEPLIB  DD   DSN=INT.MODREL,DISP=SHR          Program Load Library
//SYSPRINT DD   SYSOUT=A,DCB=BLKSIZE=1210        Message Data Set
//SYSUT1   DD   DSN=INT.SYMREL,DISP=SHR          Input Library to Cross
//*                                                Reference
//SYSUT2   DD   SYSOUT=A                         Output Cross Reference
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR        OS Sort Routines
//SORTIN   DD   UNIT=SYSDA,SPACE=(TRK,(10,10))   Temp Output & Sort Input
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)       Sort
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)       Work
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)       Areas
//SYSOUT   DD   SYSOUT=A                         SORTs System Messages
//SORTOUT  DD   UNIT=SYSDA,SPACE=(TRK,(10,10))   Sorted List to be formatted
```

Sample output is shown on the following page.

NOTE: A list of COPY statements alphabetized by module name may be obtained via the IAIMOPCD program.

```
                                   COPY OCCURRENCE CROSS-REFERENCE LISTING                 8/05/81     PAGE   1
MEMBER       COPIED BY
$LUCCALL      LCOMP     LUNIT

$LUCSB        LCOMP     VTCSB

ACCTYPES      MAPACCT   SAMSECT   TRACK

ALTREPRT      PMIOUTPT

ASMLOGCH      MPGGEN    WORKSHPM

ASYDSECT      ASYNCLDR  CLOSDWN3  LOADOVLY  PMIDLOAD  STARTUP3  SYCT400

BRODSECT      BROADRTN  FESEND

BTAMWORK      BDIAL     BLHIN     BLHOT     BSCDIAL   BSCLEASE  BTSEARCH  BUNKRAMO  CNT01MOD  ERRSTMSG
              GENVERB   GRAPHICS  PMIWILT   PMI2741   PMI3735S  PMI7770S  SIMTTY

BTMDSECT      FEMSG     TPUMSG

CHKPTDST      CHECKPT3  RESTORE3

DDQENV        DDQMOD    DDQSTART

DDQSECTS      DDQMOD    DDQSTART

DDSASECT      IXFHND01

DEVLISTC      INTBFTAI  INTSEC00  INTSEC02  ISGEN     MAPIN     MAPOUT    MMUDDM    MMUDDMT   MMUDDMU
              MMUSTART  OUT3270   PMIDVASN  PMIEXTRM  PMIOUTPT  PMIRJE    RJEROUT   RJESEND   SIMCRTA
              TALLY

DEVSETNG      DEFINE    DEVICE    ISGEN     LOADMAP   LPSPA     MAPGROUP  MAPOUT    MMUSTART  MMUVT
              STATION

DEVTABL       BDIAL     BLHIN     BLHOT     BMH000    BSCDIAL   BSCLEASE  BSTAT2    BTAMLINF  BTAMSIM
              BTAMSTAT  BTSEARCH  BTVERIFY  BUNKRAMO  CNT01MOD  COPYSS    ERRSTATS  ERRSTMSG  FECMD
              FEMSG     GFEINTFC  GRAPHICS  INTSEC02  OUTDS40   OUT3270   PMIEXTRM  PMIWILT   PMI2741
              PMI7770S  RJESEND   SIMTTY    TALLY     TCAMINTF  TPUMSG    USRECRY

DIALTABL      BDIAL     BSCDIAL   BSTAT2    BTAMLINF  BTAMSIM   BTAMSTAT  FECMD     PMI2741   PMI3735S
              TPUMSG

DLIB          CHCKPTSS  CPLUNCSS  DBCHKDSP  KEYFLIP

DSEDSECT      MRINTER

DYNDSECT      DYNLLOAD  ICOMDYNL  LOADSCT   PMIPL1    STARTUP3  TDUMP

EDITPRMC      EDIT000   EDIT001   EDIT002   EDIT003   EDIT005   EDIT008   EDIT009   PMIEDIT

ENQDSECT      MRINTER   MRPURGE
```

F.3 IAIMGOCR and IAIMGOC2--Global Occurrences

These programs generate an alphabetical list of all global symbols that are referenced, the modules that reference them and the sequence numbers of the statements at which they are referenced. The input data set (SYSLIB) is a card image, Assembler Language source library.

Sample JCL:

```
//IAIMGOCR  JOB   ...,REGION=512K
//STEP1     EXEC  PGM=IAIMGOCR
//STEPLIB   DD    DSN=INT.MODREL,DISP=SHR                Program Load Library
//SYSLIB    DD    DSN=INT.SYMREL,DISP=SHR                Input Library to
//*                                                        Cross Reference
//SYSPRINT  DD    SYSOUT=A                               Message Data Set
//SYSOUT    DD    SYSOUT=A
//GLOBAL    DD    DSN=&&STGOCR,DISP=(NEW,PASS),          Cross Reference to
//                DCB=BLKSIZE=1210,UNIT=SYSDA,             be sorted
//                SPACE=(TRK,(20,20))
//STEP2     EXEC  PGM=SORT
//SORTLIB   DD    DSN=SYS1.SORTLIB,DISP=SHR              OS Sort Routines
//SORTIN    DD    DSN=&&STGOCR,DISP=(OLD,DELETE)         Sort Input
//SORTWK01  DD    UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)    Sort
//SORTWK02  DD    UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)    Work
//SORTWK03  DD    UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)    Areas
//SYSOUT    DD    SYSOUT=A                               Sort System Messages
//SORTOUT   DD    DSN=&&STGOC2,DISP=(NEW,PASS),          Sort Output
//                SPACE=(TRK,(20,20),DCB=(RECFM=FBA,
//                LRECL=121,BLKSIZE=121),UNIT=SYSDA
//SYSIN     DD    *                                      Sort Control Fields
    SORT FIELDS=(2,9,A,11,8,A,21,8,A),FORMAT=CH
/*
//STEP3     EXEC  PGM=IAIMGOC2                           Print Program
//STEPLIB   DD    DSN=INT.MODREL,DISP=SHR                Program Load Library
//SYSUT1    DD    DSN=&&STGOC2,DISP=(OLD,PASS)           Input From Sort
//SYSUT2    DD    SYSOUT=A,DCB=BLKSIZE=1210              Output Cross Reference
//SYSPRINT  DD    SYSOUT=A                               Message Data Set
```

Sample output is shown on the following page.

```
                                            GLOBAL OCCURRENCE CROSS-REFERENCE LISTING                 8/05/81     PAGE    1
SYMBOL    MEMBER    REFERENCES
&$B       $LUCSB    00110000 00240000 00240000 00280000 00280000 00300000 00300000 00430000 00430000 00440000
          $LUCYN    00014000 00030000 00030000
          LCOMP     00260000 01150000 01190000 01190000 01200000
          VTCSB     00220000
&$BC      $LUCSB    00120000 00220000 00220000 00310000 00310000 00490000 00490000 00500000
          $LUCYN    00014000 00018000 00018000 00023000 00023000
          LCOMP     00260000 01160000 01160000
          VTCSB     00220000
&$CSBGEN  VTLSB     00240000 00620000 00640000
&$CTC     $LUCLASS  00190000 00270000 00290000 00310000 00311000 00312000 00390000 00450000 00460000
&$CTINIT  $LUCLASS  00200000 00250000 00320000
&$CTN     $LUCLASS  00180000 00260000 00280000 00300000 00310500 00311500 00370000
&$CVBGEN  VTCSB     00200000 00530000 00550000
&$LVBGEN  VTLSB     00220000 00651500 00652500
&$NODEF   $LUCSB    00230000
          $LUCYN    00015000 00019000
          LCOMP     00270000 01030000 01170000
          VTCSB     00230000 00680000
&$TCLASS  $LUCLASS  00170000 00340000 00450000 00460000
          ICOMLINK  00048100 00324063 00324090 00324210
          VTCSB     00210000 00310000 00380000 00431100 00530000 00540000 00550000 00620000
          VTCVB     00200000 00310000 00320000 00340000 00400000 00441000
          VTLSB     00230000 00310000 00350000 00560300 00615000 00651500 00652000 00652000 00652500 01770000
          VTLSB     02133000
&$TCODE   $LUCLASS  00170000 00460000
          ICOMLINK  00048100
          VTCSB     00210000 00620000
          VTCVB     00200000
          VTLSB     00230000 00562000 00615000 01770000
&$V       $LUCCV    00140000 00160000 00200000 00230000 00280000 00310000
          LCOMP     00260000
          VTCSB     00220000 00770400 00770600 00790000 00790000 00800000 00800000
&#CT      SMPRCF    00040000 00370000
&#IN      SMPRCF    00030000 00360000
&#MD      SMPRCF    00060000 00390000
&#SM      SMPRCF    00050000 00380000
&@AREA    VTLUCM2   00900000 01160000 01170000 01290000 01310000
&A        BER       00002000 00003000
          BHE       00002000 00003000
          BHER      00002000 00003000
          BHR       00002000 00003000
          BLE       00002000 00003000
          BLER      00002000 00003000
          BLR       00002000 00003000
          BMR       00002000 00003000
          BNER      00002000 00003000
          BNHR      00002000 00003000
          BNLR      00002000 00003000
          BNMR      00002000 00003000
          BNOR      00002000 00003000
          BNPR      00002000 00003000
          BNZR      00002000 00003000
          BOR       00002000 00003000
          BPR       00002000 00003000
```

F.4 IAIMMOCR--Macro Occurrences

This program produces an alphabetical list of all referenced macros and the modules that reference them. The input data set (SYSUT1) is a card-image, Assembler Language source library.

The names of the macros to cross-reference are described on control cards in the SYSIN data set. There are three types of control cards: INCLUDE, INCLUDE ONLY and EXCLUDE. These control words can appear anywhere before column 72 and are followed by a list of macro names, separated by commas. If all macro names cannot fit on one control card, the card should end with a comma before column 72 and can be continued beginning in any column on as many continuation cards as needed. If the SYSIN data set is empty, all member names of the SYSLIB data set will be used as macro names to cross-reference. The function of each control card is as follows:

- INCLUDE ONLY

  The cross-reference produced will only be for the macros that appear on this card.

- INCLUDE

  All member names in the SYSLIB data set as well as any names on this card will be used as macro names to cross-reference. This might be used when you want to scan for certain OS macros, as well as all Intercomm macros.

- EXCLUDE

  All member names of the SYSLIB data set will be used as macro names except for those listed on this control card.

The INCLUDE and EXCLUDE control cards may both be used during a run of the program. The INCLUDE card would specify additional macro names to those in the SYSLIB data set while the EXCLUDE card would limit the scan by deleting from the cross-reference specified macro names that appeared in the data set. For example, you may want to exclude some Intercomm macros, while including some OS macros.

To obtain a cross-reference of all OS/VS macros, as well as Intercomm macros, concatenate SYS1.MACLIB (and SYS1.AMODGEN if MVS) to the SYSLIB data set.

Sample JCL:

```
//IAIMMOCR JOB
//STEP1    EXEC PGM=IAIMMOCR,REGION=128K
//STEPLIB  DD   DSN=INT.MODREL,DISP=SHR                Program Load Library
//SYSIN    DD   *                                      Control Cards
    INCLUDE ATTACH,CALL,LINK,XCTL
/*
//SYSLIB   DD   DSN=INT.SYMREL,DISP=SHR                Macro Names Library
//SYSUT1   DD   DSN=INT.SYMREL,DISP=SHR                Input Library to XREF
//SYSUT2   DD   SYSOUT=A,DCB=BLKSIZE=1210              Output Cross Reference
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR              OS Sort Routines
//SORTIN   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))           Sort Input
//SORTOUT  DD   UNIT=SYSDA,SPACE=(TRK,(10,10))         Sort List to be formatted
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG) Sort
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG) Work
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG) Areas
//SYSOUT   DD   SYSOUT=A                               Sort System Messages
```

Sample output is shown on the following page.

```
                                     MACRO OCCURRENCE CROSS-REFERENCE LISTING                        8/05/81    PAGE    1

MACRO NAME    REFERENCED BY
$LUCCV        VTCSB
$LUCLASS      ICOMLINK  VTCSB     VTCVB     VTLSB
$LUCYN        $LUCSB    LCOMP
AIDDATA       BTSAMP
AIDGRP        BTSAMP
ATTACH        ABTOTEND  ATTOTRS   BTAMLINE  DYNLINK   ICOMDYNL  ICOMTASK  IJKTLOOP  MRINTER   PMISNAP1
              RJESTART  STARTUP3  STOSTART  TCAMASYN  TOTSTART
ATTRIB        LOGCHARS
BDEVICE       BTSAMP    BTVRBTB   FEINSTDK
BER           BLHSTRC   BSCLEASE  FECMD     IXFHND01  MAPIN     PDATBASE  VTEXITS   VTRECVE
BETAMOD       INTBETAI
BHR           BTSEARCH  FECMD     VTRECVE
BLINE         BTSAMP    BTVRBTB   FEINSTDK
BLOCKAD       COBOLGN
BLOCKAW       COBOLGN
BLOCKBD       COBOLGN
BLOCKBW       COBOLGN
BLOCKCD       COBOLGN
BLOCKCW       COBOLGN
BLR           LOGANAL   MRINTER
BNER          BLHIN     FECMD     IXFHND01  LOGANAL   MAPIN     VTQMOD
BNHR          BLER      BSCLEASE  FECMD     IXFFAR    TPUMSG
BNLR          BHER      FECMD
BNOR          IXFHND01
BNPR          EXMVE
BNZR          BMH000    FECMD     LOGANAL   SYCT400   VTCDM2    VTLUCMD   VTQMOD    VTRECVE
BOR           BTSEARCH  DYNLLOAD  IGCICOM   IXFHND01  LOGANAL   MAPIN     MRCSAMOD  SYCT400   TALLY
              VTEXITS   VTLUCMD   VTQMOD    VTRECVE   VTSEND
BTERM         BTSAMP    BTVRBTB   FEINSTDK
BTERR         BLHIN     BLHOT     BSCDIAL   BSCLEASE  BTSEARCH  PMI2741   PMI3735S  PMI7770S  TPUMSG
BTLCEX        LOGPROC   LOGPUT    MRIDAXS
BTMSG         FEMSG     TPUMSG
BTSPA         BDIAL     BLHIN     BLHOT     BLHSTRC   BLMSGCOL  BMH000    BSCDIAL   BSCLEASE  BSEGMOD
              BSTAT2    BTAMLINE  BTAMSIM   BTAMSTAT  BTSEARCH  BTVERIFY  BUNKRAMO  CNT01MOD  COPYSS
              ERRSTMSG  FECMD     FEMSG     GFEINTFC  GRAPHICS  LOGPUT    MRMOD     PMIEXTRM  PMIRETRV
              PMIWILT   PMI2741   PMI3735S  PMI7770S  QUEUEMOD  REQONDDQ  SIMTTY    SSPOLL    TALLY
              TCAMINTF  TPUMSG    USRECRY   USRER129
BTVERB        BTSAMP    BTVRBTB   FEINSTDK
BVBASE        FECMD     FEMSG
BZR           BDIAL     BLHIN     BLHSTRC   BSCLEASE  FECMD     FEMSG     INTSEC02  IXFHND01  LOGPROC
              MAPIN     MRCSAMOD  PMIEXTRM  TALLY     VTCDM2    VTEXITS   VTLUCMD   VTRECVE   VTSEND
              VTSQRSYN  WTOMOD
CALL          ABTOTEND  ASYNCLDR  ATTOTRS   BDIAL     BLHIN     BLHOT     BLHSTRC   BLMSGCOL  BMH000
              BROADRTN  BSCDIAL   BSCLEASE  BSEGMOD   BSTAT2    BTAMLINE  BTAMSIM   BTSEARCH  BTVERIFY
              BUNKRAMO  CALCRBN   CALLIF    CFMSINTF  CFMSMAC1  CHANGE    CHCKPTSS  CHECKPT3  CLOSDWN3
              CNT01MOD  COBPUT    CONVERSF  COPYSS    CPLUNCSS  DHADACHK  DBRSTRT   DDQINTFC  DDQMOD
              DDQSTART  DISPLAY   DYNLLOAD  ECHOMSG   EDIT006   FDPSEND   FECMD     FEMSG     FESEND
              FINTUNER  FORMGEN   GETSEG    GFEINTFC  GPSS      GRAPHICS  IJKPRINT  INTBETAI  INTCRQ
              INTDPLOK  INTDEXTR  INTPRQ    INTSEC00  INTSEC02  INTSTORF  INTSTROB  INTSTS    INTVL
              IRANGE    ISGEN     ISGOUT    IXFCHKPT  IXFCREAT  IXFCTRL   IXFDISAM  IXFFAR    IXFHND00
              IXFHND01  IXFLOG    IXFOPR01  IXFRPT01  IXFRVRSE  IXFSNAPL  IXFVERF1  LHTRACE   LOADMAP
              LOADPAGE  LOADSCT   LOGANAL   LOGHIST   LOGINPUT  LOGPRINT  LOGPROC   LOGPUT    LOGRESP
```

F.5 IAIMMRF1--Global and Sequence Symbol References (Card-image Input)

This program prints out an Assembler Language source program or macro, followed by an ordered cross-reference listing of all global symbols (&XXX) and sequence symbols (.XXX). The input (SYSLIB) is a card image, Assembler Language sequential or partitioned data set. When a partitioned data set is used as input, the program produces a cross-reference of each member in the data set that has MACRO as the op-code on the first card. To specify specific members of a PDS to be cross-referenced, rather than the entire data set, code the PARM parameter on the EXEC statement. The PARM parameter should be set to a list of member names to be cross-referenced, with a comma separating each name.

Sample JCL:

```
//IAIMMRF1   JOB
//STEP1      EXEC  PGM=IAIMMRF1,REGION=100K
//                 [,PARM='BTERM,STATION']     optional
//STEPLIB    DD    DSN=INT.MODREL,DISP=SHR     Program Load Library
//SYSLIB     DD    DSN=INT.SYMREL,DISP=SHR     Input Library
//SYSPRINT   DD    SYSOUT=A                    Message Data Set
//SYSOUT     DD    SYSOUT=A,DCB=BLKSIZE=1210   Output Cross Reference
```

Sample output is shown in the following two pages.

```
STMT      SOURCE STATEMENT                                                       PMISNAP  PAGE      1

   1                MACRO                                                               00000100
   2      &NAME     PMISNAP &DCB=,&TCB=,&ID=,&SDATA=,&PDATA=,&STORAGE=,&LIST=,         *00000200
                          &FAST=NO,             TO REQUEST A FAST SNAP               SKX00000210
                          &INDUMP=NO,                INDICATIVE DUMP                 DMK*00000215
                          &SPA=,&SPAEXT=,                                             RB*00000220
                          &MF=                                                          00000300
   3                LCLB  &B(8)                 FLAG BYTE                             SK 00000400
   4                LCLC  &DC                   DCB                                   SK 00000500
   5      &DC       SETC  '&DCB'                INITIALIZE DCB PARAMETER              SK 00000600
   6                AIF   ('&DCB' NE '').DCBOK     IF DCB SPECIFIED OR               DFK 00000700
   7                AIF   ('&MF(1)' EQ 'E').DCBOK     IF MF=(E,XXXX) THEN SKIP.      DFK 00000750
   8      &DC       SETC  '*-*'                 DEFAULT DCB IS TO BE USED.           DFK 00000800
   9      .DCBOK    ANOP                                                              SK 00000900
  10      &NAME     SNAP  DCB=&DC,TCB=&TCB,ID=&ID,SDATA=&SDATA,PDATA=&PDATA,          SKX00001100
                          STORAGE=&STORAGE,LIST=&LIST,MF=&MF                            00001200
  11                AIF   ('&MF' EQ 'L').MEXIT                                          00001300
  12                ORG   *-2                   DELETE SNAP SVC INSTRUCTION.          SK 00001305
  13                AIF   ('&SPAEXT' EQ '').NEXT2                                     RB 00001310
  14                L     15,SEXISNAP-SPAEXT(&SPAEXT) . GET ICOMSNAP ENTRY.           SK 00001320
  15                AGO   .BALR2                                                      RB 00001330
  16      .NEXT2    AIF   ('&SPA' EQ '').NSPA2                                        RB 00001340
  17                L     15,SPAEXTAD-SPALIST(&SPA) .GET SPA-EXTENSION                RB 00001350
  18                L     15,SEXISNAP-SPAEXT(15) . GET ICOMSNAP ENTRY.                SK 00001360
  19                AGO   .BALR2                                                      RB 00001370
  20      .NSPA2    ANOP                                                              RB 00001380
  21                L     15,=V(ICOMSNAP) .     GET ICOMSNAP ENTRY POINT.             SK 00001400
  22      .BALR2    ANOP                                                              RB 00001420
  23                BALR  14,15 .               GO TO INTERCOMM SNAP ROUTINE          SK 00001500
  24                DC    Y(ESN&SYSNDX.-*) .    LENGTH OF INTERCOMM PARAMETERS        SK 00001510
  25      &B(1)     SETB  ('&FAST' EQ 'YES')    SET FAST-SNAP FLAG                    SK 00001520
  26      &B(2)     SETB  ('&INDUMP' EQ 'YES')                                       DMK 00001522
  27                DC    B'&B(1)&B(2)&B(3)&B(4)&B(5)&B(6)&B(7)&B(8)' FLAGS           SK 00001530
  28                DC    B'0' .                RESERVED                              SK 00001540
  29      ESN&SYSNDX DS   0H .                                                        SK 00001550
  30      .MEXIT    MEND                                                              SK 00001700
```

```
SYMBOL    REFERENCES                                                              PMISNAP  PAGE      2

.BALR2      15    19    22
.DCBOK       6     7     9
.MEXIT      11    30
.NEXT2      13    16
.NSPA2      16    20
&B           3    25    26    27    27    27    27    27    27    27    27
&DC          4     5     8    10
&DCB         2     5     6
&FAST        2    25
&ID          2    10
&INDUMP      2    26
&LIST        2    10
&MF          2     7    10    11
&NAME        2    10
&PDATA       2    10
&SDATA       2    10
&SPA         2    16    17
&SPAEXT      2    13    14
&STORAGE     2    10
&SYSNDX     24    29
&TCB         2    10
```

F.6 IAIMMRF2--Global and Sequence Symbol References (121-column Assembler listing input)

This program produces an ordered cross-reference listing of all global symbols (&XXX) and sequence symbols (.XXX) of an Assembler Language program or macro. The input (SYSLIB) is a 121-column Assembler-listing sequential or partitioned data set. The input library is not supplied to the user; if one has not been created, the uses described below for permanent data sets are not applicable. However, the program may still be used as an Assembler postprocessor for temporary data set input (see below).

With permanent data set input, the program produces a cross-reference of each member of the input partitioned data set. To produce cross-references for specific members of a PDS, code the PARM parameter on the EXEC card for IAIMMRF2 with the list of desired members, separated by commas.

Sample JCL:

```
//IAIMMRF2  JOB ...,REGION=512K      USE WITH PERMANENT DATA SET INPUT
//*
//STEP1     EXEC  PGM=IAIMMRF2,PARM='BTERM'
//STEPLIB     DD  DSN=INT.MODREL,DISP=SHR   Program Load Library
//SYSPRINT    DD  SYSOUT=A                  Message Data Set
//SYSOUT      DD  SYSOUT=A,DCB=BLKSIZE=1210 Cross Reference Output
//SYSLIB      DD  DSN=INT.ASMREL,DISP=SHR   Input Assembler Listing
//*
```

Only the cross-reference is produced on the SYSOUT data set. If the SYSOUT data set is to include the Assembler output as well as the cross-reference, in any of the cases above, prefix the PARM parameter with the constant PRINT/, that is, code PARM='PRINT/BTERM'.

With temporary data set input, the program functions as a spot assembler when it is run immediately after an assembly with the Assembler SYSPRINT output spooled to a temporary system data set. If the temporary data set is a PDS, the PARM field on the EXEC statement for IAIMMRF2 should specify the desired PDS member name. With sequential data set input, do not code the PARM field. The output produced will have the symbol cross-reference listing appended to each Assembler listing.

For the program to function as a postassembly processor, appending the symbol cross-reference listing to the Assembler output, the following must be done:

- specify a single module name in the PARM field
- specify DISP=OLD on the input data set (SYSLIB)
- specify the last member written to SYSLIB as the module name.

Sample JCL:

```
//IAIMMRF2  JOB ...,REGION=512K        USE AS AN ASSEMBLER POSTPROCESSOR
//*
//STEP1  EXEC   ASMPC,Q=REL,NAME=BTERM           Assembly Step
//ASM.SYSPRINT  DD DSN=&&TEMP(BTERM),UNIT=SYSDA,
//              SPACE=(CYL,(2,2,1),RLSE),        Temporary Assembler
//              DISP=(,PASS)                     Listing Library
//*
//STEP2  EXEC   PGM=IAIMMRF2,PARM='BTERM'
//STEPLIB  DD   DSN=INT.MODREL,DISP=SHR          Program Load Library
//SYSPRINT DD   SYSOUT=A                         Message Data Set
//SYSOUT   DD   SYSOUT=A,DCB=BLKSIZE=1210        Cross Reference Output
//SYSLIB   DD   DSN=&&TEMP,DISP=(OLD,DELETE)     Input Assembler Listing
//*                                              Library
```

Sample symbol cross-reference output is shown on the following page.

```
SYMBOL    REFERENCES                                                                  BTERM     PAGE      1

.ADPLS     445   450
.ADPLS1    397   476
.ADPLS2    434   478
.ALT1      310   317
.AUTUP     184   468
.AUTUP1    185   471
.BADTPUP    82   539
.BSCLST    271   289
.BSCNOID   111   547
.BSDCALL   232   234
.BSDERR2    99   100   120   122   127   130   552
.BSDFONE   129   132
.BSDID     332   335
.BSDIGIT   353   357
.BSDMISS   116   548
.BSDSW     223   226
.BTRTDIL   171   430
.BTRT1     377   381
.BTRT2     330   380   398   401   421   428   436   448   463
.BTRT2A    401   403
.BTRT3     404   454
.BTRT4     405   457
.BTRT5     406   458   460
.BTRT5A    408   467   474
.BTRT6     410
.BTSKP      95   136   137   138   139   140   142   520   522
.BTSKP11   158   165
.BTSKP12   164   167
.CKBSDID   126   130
.CKGBLID    98   111
.CKRCVD    106   110   116
.CNT1050   382   461
.CONV      406   464
.CONV2     465
.CRT       405   458
.CTL       403   451
.CTLBYTE   363   365
.CTLERR    451   523
.DILEN     246   248
.DILERR1    86   525
.DILERR2    87    89    93   527
.DILERR4    88   529
.DILERR5    91   531
.DILNO10   277   280
.DILNO27   281   285
.DUMMY1    407   472
.END       416   556
.ENDWILT   302   328   339   360
.ERROR1    241   533
.ERROR2    307   535
.ERROR3    303   537
.ERR1       81   169   518
.ERR2      141   385   520
.FINLOC    257   259
.GBLID     102   106
.GENFONE   348   351
.GENGR     388   417
.GENGR2    391   419
```

F.7 IAIMOPCD--Op-Code Occurrences

This program produces an ordered source library directory listing and a detailed sorted operation code occurrence cross-reference listing including:

- Operation code or macro name
- Member name
- Concatenation number within SYSLIB (which may be a concatenation of several data sets) of the data set in which the operation code was found
- Complete source statements for each occurrence of the op-code/macro

Operation codes cross-referenced by this program are all macro names (plus COPY) found in the SYSOPLIB data set(s) directory which are also found on the SYSLIB data set(s) and specific operation codes (macro names) requested via SYSIN control cards. SYSIN control cards consist of op-codes separated by commas or blanks.

Op-codes SSK and ISK, for example, should be specified via SYSIN. If the SYSOPLIB DD is DUMMYed, output will consist only of those op-codes (macros) requested via SYSIN.

Sample JCL:

```
//IAIMOPCD JOB
//STEP1    EXEC PGM=IAIMOPCD
//STEPLIB  DD   DSN=INT.MODREL,DISP=SHR       Program Load Library
//SYSPRINT DD   SYSOUT=A                      Opcode Occurence Listing
//SYSLIB   DD   DSN=INT.SYMREL,DISP=SHR       Opcode Input Library
//SYSIN    DD   *                             Optional/Additional Opcodes
SSK,ISK
/*
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(4,1)) Sort
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(4,1)) Work
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(4,1)) Data Sets
//SYSOPLIB DD   DSN=INT.SYMREL,DISP=SHR       Op-code Directory Library
//         DD   DSN=SYS1.MACLIB,DISP=SHR      (or Libraries)
```

Sample output (SYSIN was omitted) is shown on the following two pages.

```
                                        REQUESTED 'OP-CODE' DIRECTORY                 08/06/81          PAGE       1

$LUCCALL   $LUCCV     $LUCLASS   $LUCSB     $LUCYN     AAAAAAAA   ABEND      ABTOTEND   ACB        ACBVS
ACCTYPES   ACI        ADDUSR     ADJTCBQ    AIDDATA    AIDGRP     ALTREPRT   AMDSADMP   AMDSADM2   AMGINTFC
ANALYZ     AS         ASCBCHAP   ASCTR      ASGNBFR    ASLIST     ASMLOGCH   ASMOC      ASMPC      ASMPCL
ASMPCM     ASMTRTAB   ASYDSECT   ASYNCH     ASYNCLDR   ATLAS      ATTACH     ATTEN      ATTNINQ    ATTOTRS
ATTRIB     BATCHPAK   BCGROUP    BCS        BDEVICE    BDIAL      BEINSTCK   BER        BETAMOD    BHE
BHER       BHR        BINSRCH    BLDL       BLDVRP     BLE        BLER       BLHIN      BLHOT      BLHSTRC
BLHTRACE   BLINE      BLMSGCOL   BLOCKA     BLOCKAD    BLOCKAW    BLOCKB     BLOCKBD    BLOCKBW    BLOCKC
BLOCKCD    BLOCKCW    BLR        BLS        BMH000     BMR        BNER       BNGIAL     BNGIBT     BNGIER
BNGIIALL   BNGIIBIT   BNGIIERR   BNGIILC1   BNGIILC2   BNGIIMN1   BNGIIMN2   BNGII327   BNGIL1     BNGIL2
BNGIM1     BNGIM2     BNGIOALL   BNGIOBIT   BNGIOERR   BNGIOLC1   BNGIOLC2   BNGIOMN1   BNGIOMN2   BNGIO327
BNGISALL   BNGISBIT   BNGISERR   BNGISLC1   BNGISLC2   BNGISMN1   BNGISMN2   BNGIS327   BNGI32     BNGTDEMF
BNHR       BNLR       BNMR       BNOR       BNPR       BNZR       BOR        BPR        BQBOCOMN   BQBOJOB
BROADRTN   BRODSECT   BSCDIAL    BSCLEASE   BSEGMOD    BSP        BSTAT2     BTAMLINE   BTAMSCTS   BTAMSIM
BTAMSTAT   BTAMWORK   BTERM      BTERR      BTLCEX     BTMDSECT   BTMSG      BTSAMP     BTSEARCH   BTSPA
BTVERB     BTVERIFY   BTVRBTB    BUFINQ     BUILD      BUILDRCD   BUNKRAMO   BUR        BVBASE     BZR
CA         CALCRBN    CALL       CALLDISP   CALLGFE    CALLIF     CALLOVLY   CALLRTM    CALLTSSR   CALL3886
CALL3890   CAMLST     CANCELMG   CARRIAGE   CATALOG    CATCH      CCI        CDAL       CFMSINTF   CFMSMAC1
CFMSMAC2   CHANGE     CHANGKEY   CHAP       CHCKPTSS   CHECK      CHECKPT    CHECKPT3   CHGNTRY    CHI
CHKA1      CHKA2      CHKA3      CHKA4      CHKBI      CHKDE      CHKHF      CHKPT      CHKPTDST   CHKREG
CHKRG      CHKRQ      CHKSN      CHKYN      CI         CIRB       CKFIT      CKLINK     CKOVLYNO   CKREQ
CKWORK     CKWORK1    CKWORK2    CK3270     CLOSDWN3   CLOSE      CLSDST     CNCCHECK   CNTLCHR    CNTRL
CNT01MOD   COBLOGCH   COBOLGN    COBPC      COBPCL     COBPUT     COBREENT   COBSTORF   COBUPC     COBUPCL
COBUPCLD   CODE       CODETRNS   COMMAND    COMMBUF    CONFIGUR   CONVERSE   CONVERT    COPRE      COPY
COPYSS     COUNTER    CPLUNCSS   CPUIDSND   CREATEGF   CREATSIM   CRUNCH     CST        CSTBL      CT
CTBFORM    CTBL       CTRGROUP   CTRLIST    CTRSCHED   CUTOFF     CV         CVTBL      DAR        DATBASXT
DATETIME   DBADACHK   DBCHKDSP   DBMCHECK   DBRSTRT    DCB        DCBD       DDNFIND    DDQDELFE   DDQDS
DDQDSTBL   DDQENV     DDQFH      DDQINSFE   DDQINTFC   DDQMOD     DDQSECTS   DDQSTART   DDQWTO     DDSASECT
DEFAREA    DEFAULTS   DEFCCW     DEFINE     DEFSYM     DELETE     DELOAD     DEMF       DEQ        DETACH
DEULIST    DEVICE     DEVLISTC   DEVSETNG   DEVTABL    DEVTYPE    DFR        DFTRMLST   DIALTABL   DISABLE
DISCONV    DISPATCH   DISPGUID   DISPLAY    DISPSET    DISREORG   DLETE      DLIB       DLINT      DLINTIN
DLVRP      DO         DOM        DRDISK     DRDUSUB    DRFMFIND   DRFORM     DRHELP     DRINIT     DRMOVE
DRMVCL     DRNXTF     DRPOS      DRSAVE     DRTAB      DRUSCALL   DRUSPARM   DRXMIT     DSEDSECT   DSG
DSGA       DSGNL      DSIAMH     DSIART     DSICBH     DSICBS     DSICES     DSICLB     DSICLS     DSICWB
DSIDATIM   DSIDCT     DSIDDT     DSIDFL     DSIDKS     DSIDQT     DSIDSB     DSIDSRB    DSIFRE     DSIGET
DSIIFR     DSILCS     DSILOD     DSIMBI     DSIMBS     DSIMDS     DSIMQS     DSIMVT     DSINAT     DSIOIS
DSIOIT     DSIOPN     DSIPAS     DSIPDB     DSIPOS     DSIPRS     DSIPSS     DSIRDS     DSIRET     DSISAT
DSISAV     DSISCB     DSISCE     DSISCT     DSISNT     DSISSS     DSISVL     DSISWB     DSITIB     DSITID
DSITVB     DSIUSE     DSIWAT     DSIWCS     DSIWLS     DSIWTO     DSIXMH     DSIZCSMS   DSIZVSMS   DSPLY
DS40TRDM   DVMODIFY   DVT        DWF        DXR        DYNALLOC   DYNDSECT   DYNLINK    DYNLLOAD   ECHOMSG
EDITPRMC   EDIT000    EDIT001    EDIT002    EDIT003    EDIT004    EDIT005    EDIT006    EDIT007    EDIT008
EDIT009    EDIT3270   ENABLE     ENDFLD     ENDGROUP   ENDINTAB   ENDMAP     ENDMODE    ENDREQ     ENDSEG
ENQ        ENQDSECT   ENTER      ENTR       ENVIRON    EOV        ERASE      ERRLBLKD   ERRORMSG   ERRSTATS
ERRSIMSG   ERRTBLKD   ESETL      ESTAE      EVENT      EVENTS     EXC        EXCP       EXCPVR     EXECRPL
EXLST      EXLVS      EXMVE      EXSS       EXT        EXTERM     EXTRACT    EXTRT      FAKEDISP   FDETL
FDHDR      FDITCB     FDPLOAD    FDPSEND    FDPLIST    FECMD      FECMDSEC   FECMMOD    FEERR      FEINSTDK
FEMACGBL   FEMSG      FEMSGEQU   FEOV       FESEND     FESTAE     FIELD      FIND       FINDDD     FINDQNUM
FINTUNER   FIXSECT    FLDGEN     FLDSYM     FLDVALS    FM         FORMAT     FORMGEN    FORTLINK   FORWARD
FQES       FRACHECK   FRDSECTS   FREEBUF    FREEDBUF   FREEMAIN   FREEPOOL   FREEVRE    FTBLISTC   FULLSCR
FUNCT      GAMFQES    GBFLM      GBINF      GBPOS      GBPST      GCNL       GCNOP      GCNTRL     GDBSTUP
GDCDS      GDPD       GDRD       GDUAS      GDULIST    GDUTRANS   GDV        GECF       GECP       GEN
GENCB      GENERTRN   GENESIS3   GENFTBLE   GENINDEX   GENRDT     GENSD      GENSEC     GENTABL    GENVERB
GEOS       GEPI2      GEPM       GESD       GET        GETBUF     GETDEV     GETIX      GETLINE    GETMAIN
GETPHYSC   GETPOOL    GETSCAN    GETSEG     GETSPA     GETTHRED   GETVRF     GEVI2      GEVM       GFF
GFEDSECT   GFEINTFC   GIBLC      GINIT      GMVA       GMVD       GNOP2      GNOP4      GODFL      GPDI
GPSS       GRAPHICS   GRLAD      GREADR     GROUP      GSPLC      GSRPOS     GSERV      GSRT       GSXY
[illegible]D GTND     GTRACE     GTRU       GTSIZE     GTTERM     GTXT       GUSTOR     GWRITE     HANGUP
```

```
                                   'OP-CODE' OCCURRENCE CROSS-REFERENCE LISTING       08/06/81       PAGE      1

OP-CODE      MEMBER      DSNAME    SOURCE RECORD

$LUCCV       $LUCCV       0                  $LUCCV  &SUB,&VALUE,&IDXVAL,&DEF                                  00020000

             VTCSB        0                  $LUCCV  1,&PADIN(1),,&DV(1)                                       00740000
                                             $LUCCV  2,&PADIN(2),,&DV(2)                                       00750000
                                             $LUCCV  3,&PADOUT(1),,$DV(3)                                      00760000
                                             $LUCCV  4,&PADOUT(2),,&DV(4)                                      00770000

$LUCLASS     $LUCLASS     0                  $LUCLASS &VALUE=,&MACRO=                                          00020000

             ICOMLINK     0                  $LUCLASS  VALUE=&VTAM(&I),MACRO=ICOMLINK                       JA 00324080

             VTCSB        0                  $LUCLASS  MACRO=VTCSB,VALUE=&COMPTYP                              00300000

             VTCVB        0                  $LUCLASS MACRO=VTCVE,VALUE=&COMPTYP ELSE GET FROM COMPTYP         00330000

             VTLSB        0                  $LUCLASS  MACRO=VTLSB,VALUE=&LUTYPE                               00300000

$LUCYN       $LUCSB       0                  $LUCYN  RLSERSP,&RLSERSP,1      BIT 0                             00130000
                                             $LUCYN  CRT,&CRT,0              BIT 1                             00140000
                                             $LUCYN  CONV,&CONV,0            BIT 2                             00150000
                                             $LUCYN  SEGLOCK,&SEGLOCK,1      BIT 3                             00160000
                                             $LUCYN  LOG,&LOG,1              BIT 4                             00170000
                                             $LUCYN  LSYNCH,&LSYNCH,0,COMPL=YES   BIT 5                     SM1168 00180000

             $LUCYN       0                  $LUCYN  &NAME,&VALUE,&DEF,&COMPL=NO                               00002000

             LCOMP        0                  $LUCYN  TPUP,&TPUP,1       DEF IS TPUP=YES                        01180000

ABEND        ABTOTEND     0        ABORT     ABEND 017,DUMP                                                    00033000

             ATTOTRS      0                  ABEND 1021,DUMP                                                   00007200

             BLHOT        0                  ABEND 21,DUMP                                        M.H. 1/13/72 16243000

             BMHD00       0        ERRNOTF   ABEND 777,DUMP                                                    00557000

             BTAMLINE     0                  ABEND 045,DUMP                                                    01460555
                                             ABEND 004,DUMP                                                    03100000
```

F.8 IAIMXRF1 and IAIMXRF2--CSECT Sizes, Entry Points and External Symbols

This program produces a listing (PSIZE) of the sizes of all CSECTs within member names in a load library, in K (and/or fractions thereof) and hexadecimal, as well as entry point (PNTRY) and external reference (PXTRN) cross-references. The input library (DMOD) is a load module library.

Sample JCL appears on the following page.

```
//IAIMXREF JOB  ...,REGION=512K
//STEP1    EXEC PGM=IAIMXRF1
//STEPLIB  DD   DSN=INT.MODREL,DISP=SHR              Program Load Library
//DMOD     DD   DSN=INT.MODREL,DISP=SHR              Input Library
//DNTRY    DD   DSN=&&NTRY,DISP=(,PASS),             Entry Points
//              SPACE=(2400,(100,100)),                to be sorted
//              UNIT=SYSDA,DCB=BLKSIZE=2400
//DXTRN    DD   DSN=&&XTRN,DISP=(,PASS),             External References
//              SPACE=(2400,(100,100)),                to be Sorted
//              UNIT=SYSDA,DCB=BLKSIZE=2400
//PSIZE    DD   SYSOUT=A,DCB=BLKSIZE=1210            CSECT sizes output
//SYSPRINT DD   SYSOUT=A                             Message Data Set
//*
//STEP2    EXEC PGM=SORT                             Entry Point Sort
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR            OS Sort Routines
//SYSOUT   DD   SYSOUT=A                             Sort's System Messages
//SORTIN   DD   DSN=&&NTRY,DISP=(OLD,DELETE)         Entry Points
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)   Sort
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)   Work
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)   Areas
//SORTOUT  DD   DSN=&&NTRYS,DISP=(,PASS),            Entries to be Printed
//              SPACE=(240,(300,300)),UNIT=SYSDA,
//              DCB=(RECFM=FB,LRECL=24,BLKSIZE=240)
//SYSIN    DD   *
    SORT FIELDS=(1,8,A,17,8,A,9,8,A),FORMAT=CH
/*
//STEP3    EXEC PGM=SORT                             External Reference Sort
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR            OS Sort Routines
//SYSOUT   DD   SYSOUT=A                             Sort's System Messages
//SORTIN   DD   DSN=&&XTRN,DISP=(OLD,DELETE)         External References
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)   Sort
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)   Work
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(3),,CONTIG)   Areas
//SORTOUT  DD   DSN=&&XTRNS,DISP=(,PASS),            External References
//              SPACE=(240,(300,300)),UNIT=SYSDA,      to be Printed
//              DCB=(RECFM=FB,LRECL=24,BLKSIZE=240)
//SYSIN    DD   *
    SORT FIELDS=(1,8,A,17,8,A,9,8,A),FORMAT=CH
/*
//STEP4    EXEC PGM=IAIMXRF2
//STEPLIB  DD   DSN=INT.MODREL,DISP=SHR              Program Load Library
//PNTRY    DD   SYSOUT=A,DCB=BLKSIZE=1210            Entry Points List
//PXTRN    DD   SYSOUT=A,DCB=BLKSIZE=1210            External References List
//INXTRN   DD   DSN=&&XTRNS,DISP=(OLD,DELETE)
//INNTRY   DD   DSB=&&NTRYS,DISP=(OLD,DELETE)
//SYSPRINT DD   SYSOUT=A                             Message Data Set
```

Sample output is shown on the following three pages.

```
     CORE ESTIMATES ORDERED BY MODULE AND CSECT                    81.217

MEMBER    CSECT     SIZE
NAME      NAME      (K)    (HEX)

AAAAAAAA
          AAAAAAAA  1.2    004A8



ABTOTEND
          ABTOTEND  0.4    0016E
          TOTABRTS  0.8    00328



AMGINTFC
          AMGINTFC  0.5    00204



ASYNCH
          ASYNCH    0.1    00079



ASYNCLDR
          ASYNCLDR  0.4    00198



ATTOTRS
          ATTOTRS   0.5    001D8



BATCHPAK
          BATCHPAK  1.0    003E5



PAGE    1
```

```
LISTING BY ENTRY POINT                                              81.217

ENTRY PT   CSECT      MEMBER

AAAAAAAA   AAAAAAAA   AAAAAAAA
ABNDCANC   SPIESNP2   SPIESNAP
ABTOTEND   ABTOTEND   ABTOTEND
ACCESS     IXFMON01   IXFHND01
           IXFMON01   PMIEXLD
ACCTABLE   RSMGMNT    MANAGER
ADCON#     IHCFCVTH   INTVL
ADJSWTCH   IHCEFNTH   INTVL
AEICLDWN   PMICLDWN   CLOSDWN3
AEIOUTPT   PMIOUTPT   PMIOUTPT
AEISNAP    PMIDCB     PMIDCB
           PMIDCB     PMIEXLD
AEISTUP    PMISTUP    STARTUP3
AIDCSECT   AIDCSECT   BLHIN
ALLOCATE   IXFMON01   IXFHND01
           IXFMON01   PMIEXLD
ALOG       IHCSLOG    INTVL
ALOG10     IHCSLOG    INTVL
AMGINTFC   AMGINTFC   AMGINTFC
ANSWER     CNT01MOD   CNT01MOD
ARITH#     IHCEFNTH   INTVL
ASMTDLI    SBTSK      SBTSKDLI
ASYNCECB   ASYNCECB   SYCT400
ASYNCH     ASYNCH     ASYNCH
ASYNCLDR   ASYNCLDR   ASYNCLDR
ASYNECBS   DELOAD     DELOAD
ASYNMSGC   ASYNMSGC   WTOMOD
ATTNRT     GRAPHICS   GRAPHICS
ATTCTRS    ATTOTRS    ATTOTRS
AUTOTPUP   TPUMRES    TPUMSG
BADCONUM   SUBOUTPT   PMIOUTPT
BADPRM     SUBEDIT    PMIEDIT
BADVMI     SUBOUTPT   PMIOUTPT
BATCHPAK   BATCHPAK   BATCHPAK
           BATCHPAK   PMIEXLD
BDIAL      BDIAL      BDIAL
BE001I     WTOSECU    SECURE00
BE006I     WTOCPID    CPUIDSND
BE007I     WTOCPID    CPUIDSND
BE009I     WTOCPID    CPUIDSND
BFAMSTRT   BTAMSTRT   PMIBTSTR
BINSRCH    BINSRCH    BINSRCH
           BINSRCH    IAIMMOCR
BINSRCH2   BINSRCH    BINSRCH
           BINSRCH    IAIMMOCR
BITSECT    BITSECT    PMIEXLD
           BITSECT    SPAEXT
BI001A     WTOVRFY    BTVERIFY
BI002I     WTOVRFY    BTVERIFY

PAGE    1
```

```
LISTING BY EXTERNAL REFERENCE                                  81.217

EXT-REF    CSECT      MEMBER

ABNDCANC   STAERTRY   STAERTRY
ABTOTEND   TOTABRTS   ABTOTEND
           TOTSTART   TOTSTART
ACCESS     SPAEXT     PMIEXLD
           REENTSB1   REENTSBS
           SPAEXT     SPAEXT
           LPSPA      SPALP
ACCTABLE   RMTRACE    RMTRACE
ADABAS     DBADACHK   DBADACHK
           DBRSTRT    DBRSTRT
ADCON#     IHCECOMH   INTVL
           IHCEFNTH   INTVL
           IHCETRCH   INTVL
ADDREC     SPAEXT     PMIEXLD
           SPAEXT     SPAEXT
ADJSWTCH   IHCECOMH   INTVL
AIDCSECT   BLHIN      BLHIN
AIDDATA    BLHIN      BLHIN
           VTCDM2     VTCDM2
           VTSTART    VTSTART
ALLOCATE   SPAEXT     PMIEXLD
           REENTSB1   REENTSBS
           SPAEXT     SPAEXT
           LPSPA      SPALP
ALOG       INTVL      INTVL
AMGSTART   IXFMON00   IXFHND00
           IXFMON00   PMIEXLD
ANALYZER   RMENQ      MANAGER
           RMPC       MANAGER
           RSMGMNT    MANAGER
ANSWER     STAERTRY   STAERTRY
ARITH#     IHCECOMH   INTVL
ASYNCECB   BATCHPAK   BATCHPAK
           IJKTLOOP   IJKTLOOP
           MRCSAMOD   MRCSAMOD
           BATCHPAK   PMIEXLD
           SPA        PMISPA
           STUOVLY    STARTUP3
ASYNCH     SPAEXT     PMIEXLD
           SPAEXT     SPAEXT
           STUOVLY    STARTUP3
ASYNCLDR   SPAEXT     PMIEXLD
           SPAEXT     SPAEXT
           STUOVLY    STARTUP3
ASYNECBS   PMIDLOAD   PMIDLOAD
           SPAEXT     PMIEXLD
           SPAEXT     SPAEXT
           PMISTUP    STARTUP3
           STUOVLY    STARTUP3

PAGE    1
```

F.9 IAIMDREF--DSECT OCCURRENCES

This program requires, as input, a PDS containing the spooled print output of assemblies of all members (programs) to be cross-referenced. Currently, for Intercomm, this data set occupies an entire 3330-1 (3330 MOD II) disk pack. Therefore, IAIMDREF is not supplied as a program to be executed at the user site. Instead, the output (with the name IAIMDREF) is supplied on microfiche (see the fiche index) as a separate page when an Intercomm release is issued. A new listing is also supplied with each periodic microfiche update, incorporating members which have been updated by SM.

F.10 RETURN CODES AND ABENDS

Module Name	Return Code	Meaning
IAIMCOCR	4	DCB Open Error. Check JCL.
	100+	Error in Sort Phase: Return code minus 100=sort-return-code.
IAIMGOCR	4	Unable to open DCB. Check JCL for SYSLIB, SYSOUT.
	8	Attempted to cross-reference a SYSLIB data set with DISP=NEW or DISP=MOD.
	12	SYSLIB data set organization (DSORG) not PS or PO.
	16	SYSLIB RECFM is V or U.
	20	SYSLIB LRECL is not 80.
	24	GLOBAL DD statement missing.
IAIMGOC2	4	DCB Open Error. Check JCL.
IAIMMOCR	4	Unable to open DCB. Check JCL.
	8	Invalid SYSIN control card.
	100+	Error in Sort Phase: Return code minus 100=sort-return-code.
IAIMMRF1	4	Unable to open DCB. Check JCL for SYSLIB, SYSOUT.
	8	Attempted to cross-reference a SYSLIB data set whose DISP=NEW or DISP=MOD.
	12	SYSLIB data set organization (DSORG) not PO or PS.
	16	SYSLIB RECFM is V or U.
	20	SYSLIB LRECL is not 80.
IAIMMRF2	4	Unable to Open DCB. Check JCL for SYSLIB, SYSOUT.
	8	Attempted to cross-reference a SYSLIB data set whose DISP=NEW or DISP=MOD.
	12	SYSLIB data set organization (DSORG) not PO or PS.
	16	SYSLIB RECFM is V or U.
	20	SYSLIB LRECL is not 121.

Module Name	Return Code	Meaning
IAIMOPCD	4	Error occurred (see accompanying WTO message)
IAIMXRF1	12	FIND macro error condition on input load module library.
	24	Unable to open input load module library.

INDEX

SPR NO.

239

SYSTEM PUBLICATION REVISION

Title: ASMF Users Guide

Product: Intercomm

Date: 7/88

New or Revised Pages:

Title page, ii-iv, 2.1-2.2, 3, 8, 10, 12-13, 16-19, 39, 71

Deleted Pages:

NONE

Unchanged Backup Pages:

4, 7, 9, 11, 14-15, 40

Subject of Attached Revisions:

Release 9.0 updates and Release 10.0 revisions and additions.

ISOGON CORPORATION