

GC28-1108-3  
File No. S370-37

**System Modification  
Program Extended**

**Program Product**

**Messages and Codes**

**Program Number 5668-949**

**IBM**

GC28-1108-3  
File No. S370-37

**Program Product**

**System Modification  
Program Extended**

**Messages and Codes**

**Program Number 5668-949**

**IBM**

## Notice to OS/VS1 Users

Do not use this book for your OS/VS1 system. SMP/E Release 3 does not support OS/VS1. Use the latest SMP/E Release 2 edition of this book instead.

## Fourth Edition (June 1985)

This edition applies to System Modification Program Extended (SMP/E) Release 3 for Release 3.8 of OS/VS2 (MVS) and to all subsequent releases unless otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

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

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department 52Q, Neighborhood Road, Kingston, N. Y., U.S.A., 12401. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever to you.

# Preface

## Who Should Read This Book

This book is for SMP/E users who need information about an SMP/E message or return code. This manual will help you identify an SMP/E message or code, understand the action taken by the system, and respond to the message as needed.

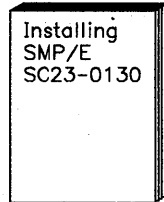
## How to Use This Book

To interpret an SMP/E message or code, see these chapters:

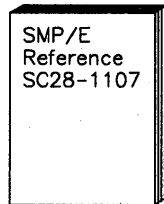
- **Chapter 1. SMP/E Return Codes** - This chapter lists return codes according to the command, interprets the return codes, and lists possible error conditions. In addition, except for the LOG command, each section outlines instructions for error recovery.
- **Chapter 2. SMP/E Messages** - This chapter lists the SMP/E diagnostic messages, along with the system action and your required response. Messages appear alphanumerically.

## Related SMP/E Books

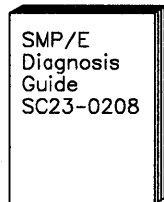
This manual will help you interpret return codes and messages that you will encounter when you run SMP/E. Use the following books for further information.



Explains how to plan for SMP/E Release 3 and install it using SMP of SMP/E.



Explains SMP/E commands and processing in detail.



Explains how to handle suspected SMP/E problems. This is new for SMP/E Release 3.

Other books in the SMP/E Release 3 library are:

- *SMP/E General Information*, GC28-1106
- *SMP/E User's Guide*, SC28-1302
- *SMP/E Reference Summary*, SX22-0006
- *SMP/E Program Directory*

## Other Related Books

Product	Books
ISPF Version 1	<p><i>ISPF General Information</i>, GC34-2078</p> <p><i>ISPF Dialog Management Services</i>, SC34-2088</p> <p><i>ISPF Dialog Management Services Examples</i>, SC34-2089</p> <p><i>ISPF/PDF for MVS: Program Reference</i>, SC34-2089</p> <p><i>ISPF for MVS Installation and Customization</i>, SC34-2084</p>
ISPF Version 2	<p><i>ISPF and PDF General Information</i>, GC34-2136</p> <p><i>ISPF Dialog Management Services</i>, SC34-2137</p> <p><i>ISPF Dialog Management Services Examples</i>, SC34-2138</p> <p><i>PDF Reference</i>, SC34-2139</p> <p><i>PDF Edit Macros</i>, SC34-2133</p> <p><i>PDF Services</i>, SC34-2134</p> <p><i>PDF Library Management</i>, SC34-2135</p> <p><i>ISPF and PDF Installation and Customization</i>, SC34-2143</p> <p><i>ISPF and PDF Diagnosis</i>, SC34-2140</p>
MVS Release 3.8	<p><i>OS/VS Linkage Editor and Loader</i>, GC26-3813</p> <p><i>OS/VS and DOS/VS Assembler Language</i>, GC33-4010</p> <p><i>OS/VS MVS Utilities</i>, GC26-3902</p> <p><i>OS/VS2 MVS JCL</i>, GC28-0692</p> <p><i>OS/VS2 System Programming Library: Service Aids</i>, GC28-0674</p> <p><i>OS/VS2 System Programming Library: Data Management</i>, GC26-3830</p>

Product	Books
MVS Release 3.8 (Cont.)	<p data-bbox="675 235 1110 262"><i>OS/VS2 DADSM Logic</i>, SY26-3858</p> <p data-bbox="675 289 1195 346"><i>OS/VS2 MVS Data Management Services Guide</i>, GC26-3875</p> <p data-bbox="675 373 1156 430"><i>OS/VS Virtual Storage Access Method Programmer's Guide</i>, GC26-3838</p> <p data-bbox="675 457 1218 485"><i>OS/VS2 Access Method Services</i>, GC26-3841</p> <p data-bbox="675 512 1110 539"><i>OS/VS2 DADSM Logic</i>, SY26-3828</p>
MVS/XA	<p data-bbox="675 554 1224 611"><i>MVS/Extended Architecture Linkage Editor and Loader User's Guide</i>, GC26-4011</p> <p data-bbox="675 638 1123 695"><i>MVS/Extended Architecture Data Administration: Utilities</i>, GC26-4018</p> <p data-bbox="675 722 1188 779"><i>MVS/Extended Architecture System-Data Administration</i>, GC26-4010</p> <p data-bbox="675 806 1091 863"><i>MVS/Extended Architecture Data Administration Guide</i>, GC26-4013</p> <p data-bbox="675 890 1237 976"><i>MVS/Extended Architecture Data Administration: Macro Instruction Reference</i>, GC26-4014</p> <p data-bbox="675 1003 1123 1060"><i>MVS/Extended Architecture VSAM Administration Guide</i>, GC26-4015</p> <p data-bbox="675 1087 1237 1173"><i>MVS/Extended Architecture VSAM Administration: Macro Instruction Reference</i>, GC26-4016</p> <p data-bbox="675 1201 1227 1287"><i>MVS/Extended Architecture VSAM Catalog Administration: Access Method Services Reference</i>, GC26-4075</p> <p data-bbox="675 1314 1175 1400"><i>MVS/Extended Architecture Integrated Catalog Administration: Access Method Services Reference</i>, GC26-4019</p> <p data-bbox="675 1428 1175 1514"><i>MVS/Extended Architecture Integrated Catalog Administration: Access Method Services Reference Summary</i>, GX26-3739</p> <p data-bbox="675 1541 1201 1627"><i>MVS/Extended Architecture DADSM and Common VTOC Access Facility Diagnosis Guide</i>, SY26-3986</p> <p data-bbox="675 1654 1149 1711"><i>MVS/Extended Architecture DADSM Diagnosis Reference</i>, SY26-3904</p> <p data-bbox="675 1711 1149 1768"><i>MVS/Extended Architecture DADSM Diagnosis Reference</i>, SY26-3904</p> <p data-bbox="675 1795 1237 1852"><i>Assembler H Version 2: General Information</i>, GC26-4035</p>

Product	Books
MVS/XA (Cont.)	<p data-bbox="846 233 1279 289"><i>Assembler H Version 2 Application Programming: Guide</i>, SC26-4036</p> <p data-bbox="846 317 1284 401"><i>Assembler H Version 2 Application Programming: Language Reference</i>, GC26-4037</p> <p data-bbox="846 428 1295 485"><i>Assembler H Version 2: Installation</i>, SC26-4030</p> <p data-bbox="846 512 1406 543"><i>MVS/Extended Architecture JCL</i>, GC28-1148</p> <p data-bbox="846 571 1370 627"><i>MVS/Extended Architecture Service Aids</i>, GC28-1159</p>
MVS/370	<p data-bbox="846 642 1344 699"><i>MVS/370 DADSM Diagnosis Reference</i>, SY26-3919</p> <p data-bbox="846 726 1312 783"><i>MVS/370 Linkage Editor and Loader</i>, GC26-4061</p> <p data-bbox="846 810 1211 842"><i>MVS/370 Utilities</i>, GC26-4065</p> <p data-bbox="846 869 1344 926"><i>MVS/370 System Programming Library: Data Management</i>, GC26-4056</p> <p data-bbox="846 953 1312 1010"><i>MVS/370 Data Management Services</i>, GC26-4058</p> <p data-bbox="846 1037 1284 1094"><i>MVS/370 Data Management Macro Instructions</i>, GC26-4057</p> <p data-bbox="846 1121 1370 1152"><i>MVS/370 VSAM User's Guide</i>, GC26-4066</p> <p data-bbox="846 1180 1328 1211"><i>MVS/370 VSAM Reference</i>, GC26-4074</p> <p data-bbox="846 1239 1386 1295"><i>MVS/370 Access Method Services Reference for VSAM Catalogs</i>, GC26-4059</p> <p data-bbox="846 1323 1386 1400"><i>MVS/370 Access Method Services Reference for the Integrated Catalog Facility</i>, GC26-4051</p> <p data-bbox="846 1428 1403 1514"><i>MVS/370 Access Method Services Reference Summary for the Integrated Catalog Facility</i>, GX26-3745</p> <p data-bbox="846 1541 1386 1598"><i>MVS/370 DADSM and Common VTOC Access Facility Diagnosis Guide</i>, SY26-3918</p> <p data-bbox="846 1625 1344 1682"><i>MVS/370 DADSM Diagnosis Reference</i>, SY26-3919</p> <p data-bbox="846 1709 1289 1766"><i>OS/VS-DOS/VS-VM/370 Assembler Language</i>, GC33-4010</p> <p data-bbox="846 1793 1117 1824"><i>MVS JCL</i>, GC28-1300</p> <p data-bbox="846 1852 1403 1908"><i>OS/VS2 MVS System Programming Library: Service Aids</i>, GC28-0674</p>



## Notations Used in This Publication

- Abbreviations used in this publication are:

**DLIB**     Distribution library

**PTF**        Program temporary fix

**SMP/E**     System Modification Program Extended

**SMP4**     System Modification Program Release 4

- Messages appear in all capital letters. Symbols are listed below:

'            Apostrophe

\*            Asterisk

Blank      (Not coded)

,

=            Equal sign

( )         Parentheses

.

- Italic lowercase words and symbols represent variables. For example, “*member*” is a variable name. However, the message that SMP/E generates will have the specific name (for example, ALPHA) where *member* appears.
- Hyphens that join lowercase words and symbols represent one variable name. For example, “*member-name*” is the variable name. However, SMP/E will substitute a specific value for the variable (for example, BETA).
- Braces enclose a group of items of which one must appear in the message. For example, “UPDATE {FAILED|SUCCESSFUL}” means that you will see either “FAILED” or “SUCCESSFUL” in the message.
- Brackets enclose an item that *may* appear in the message. For example, “[. RETURN CODE = *rc*]” means that this information does not always appear in the message.
- A vertical line separates several possible items in a message. For example, “UPDATE {FAILED|SUCCESSFUL}” means that either “FAILED” or “SUCCESSFUL” will appear in the message, but not both at once.

# Contents

<b>Chapter 1. SMP/E Return Codes</b>	<b>1</b>
ACCEPT Return Codes	2
ACCEPT Error Recovery	4
APPLY Return Codes	5
APPLY Error Recovery	7
CLEANUP Return Codes	8
CLEANUP Error Recovery	9
CONVERT Return Codes	9
CONVERT Error Recovery	9
GENERATE Return Codes	9
GENERATE Error Recovery	10
JCLIN Return Codes	10
JCLIN Error Recovery	11
LIST Return Codes	11
LIST Error Recovery	12
LOG Return Codes	12
RECEIVE Return Codes	12
RECEIVE Error Recovery	14
REJECT Return Codes	14
REJECT Error Recovery	15
REPORT Return Codes	15
REPORT Error Recovery	16
RESTORE Return Codes	16
RESTORE Error Recovery	18
SET Return Codes	18
SET Error Recovery	18
UCLIN Return Codes	19
UCLIN Error Recovery	19
ZONECOPY Return Codes	19
ZONECOPY Error Recovery	20
ZONEDELETE Return Codes	20
ZONEDELETE Error Recovery	20
ZONEEDIT Return Codes	20
ZONEEDIT Error Recovery	21
ZONEEXPORT Return Codes	21
ZONEEXPORT Error Recovery	22
ZONEIMPORT Return Codes	22
ZONEIMPORT Error Recovery	23
ZONEMERGE Return Codes	23
ZONEMERGE Error Recovery	23
ZONERENAME Return Codes	23
ZONERENAME Error Recovery	24
<b>Chapter 2. SMP/E Messages</b>	<b>25</b>
Message Format	25
Diagnostic Messages	26

# Summary of Changes

## Highlights of SMP/E Release 3 (June 1985)

### Changes for This Book

**Chapter 1. SMP/E Return Codes** - The return codes for the following new commands have been added:

CLEANUP  
REPORT  
ZONEEDIT

**Chapter 2. SMP/E Messages** - The following messages have been added to SMP/E Release 3:

GIM515	GIM625	GIM656	GIM668	GIM680
GIM610	GIM626	GIM657	GIM669	GIM681
GIM611	GIM627	GIM658	GIM670	
GIM612	GIM629	GIM659	GIM671	
GIM617	GIM630	GIM660	GIM672	
GIM618	GIM632	GIM661	GIM673	
GIM619	GIM633	GIM662	GIM674	
GIM620	GIM634	GIM663	GIM675	
GIM621	GIM650	GIM665	GIM676	
GIM622	GIM653	GIM666	GIM677	
GIM623	GIM654	GIM667	GIM678	
GIM624	GIM655		GIM679	

The following messages have been changed for SMP/E Release 3:

GIM203	GIM258	GIM448	GIM525
GIM239	GIM359	GIM451	GIM564
GIM240	GIM363	GIM505	GIM573
GIM246	GIM419	GIM508	

### Changes for SMP/E Release 3

SMP/E Release 3 does not support VS1. However, to help VS1 customers migrate to MVS, SMP/E Release 2 and its library will continue to support VS1. See the SMP/E Release 3 announcement letter for ordering information.

SMP/E performs the following new functions:

- Automatic Re-Receive of Reworked SYSMODs

SMP/E automatically re-receives a SYSMOD that is more current than the version already received on the user's system. This eliminates the need for user intervention to delete the previous version of the SYSMOD and helps ensure that the latest level of a SYSMOD is available for installation.

- CLEANUP Command

The new CLEANUP command deletes entries from the SMPMTS, SMPSTS, and SMPSCDS data sets.

- Control Block and Storage Dumps

The DEBUG command can now be used to request a dump of certain SMP/E control blocks when a specified dump point is reached. It may also be used to request a dump of SMP/E and its work areas whenever a specified message is issued.

- Cross-Zone Requisite Reporting

SMP/E provides a new REPORT command to help synchronize the installation of service for products in different zones. The REPORT command lists SYSMODs that should be installed in a zone because of conditional requisites that are installed in other zones.

- Dynamic Allocation of Temporary Data Sets

Temporary data sets can now be dynamically allocated using DDDEF entries in the GLOBAL, target, or distribution zone.

- Exception SYSMOD Processing

In addition to the existing HOLDs for exception SYSMODs, a ++HOLD modification control statement may specify a CLASS. This CLASS identifies an alternative way to release an exception SYSMOD for processing.

- Inline SOURCEID

SOURCEIDs may now be specified inline on the ++ASSIGN modification control statement, as well as on the RECEIVE command.

- JCLIN Processing

SMP/E can now save inline JCLIN during ACCEPT processing, as well as during APPLY processing. JCLIN saved in the distribution zone can be copied into the target zone and used by the GENERATE command to provide a SYSGEN-like installation process for products without SYSGEN support. Also, if SMP/E finds a program or procedure name it does not recognize, it will issue message GIM650 to warn you of the problem.

- **Moving, Deleting, and Renaming System Elements**

SYSMODs may now contain modification control statements that move, delete, or rename system elements. The ++MOVE, ++DELETE, and ++RENAME statements eliminate the need to call utilities and use UCLIN to make those changes.

- **RECEIVE by FMID**

The FORFMID operand may be specified on the RECEIVE command to limit the SYSMODs and HOLDDATA being received to a specific product or set of related products.

- **SMPLOG Improvements**

The SMPLOG is now a variable-blocked data set, which allows the user to improve performance during SMPLOG processing and to make more efficient use of DASD space. In addition, a second SMPLOG data set may be defined to hold log information when the first log data set is filled. This prevents the loss of log data.

- **SMPTLIB Cataloging**

SMP/E automatically catalogs SMPTLIB data sets when it allocates them and uncatalogs them when it deletes them.

- **User-Written Zone Descriptions**

With the UCLIN command, the user can add a zone description to GLOBAL, distribution, or target zones. This description is for customer use; SMP/E does not refer to it.

- **ZONEEDIT Command**

The new ZONEEDIT command provides a way to quickly change information in DDDEF and UTILITY entries.

# Chapter 1. SMP/E Return Codes

SMP/E sets return codes according to conditions that occur during the processes you invoke. Since SMP/E can meet many different conditions during an SMP/E invocation, the final return code that SMP/E lists is the return code associated with the most severe condition SMP/E met.

Messages report various situations and unusual conditions to the SMPOUT file. These messages fall into one of six categories that give the severity of the condition that SMP/E detected:

- Informational messages show stages of SMP/E processing or accompany other messages to further explain unusual conditions.
- Warning messages (Return code 4) - show that SMP/E detected a situation that may be invalid. Look at these messages to find if SMP/E's action was appropriate.
- Error messages (Return code 8) - show that some SMP/E processing did not complete properly. For example, when APPLY processing of a SYSMOD stops, it is always an error condition and results in a return code of at least 8.
- Severe messages (Return code 12) - show that an entire SMP/E function failed. For example, when APPLY processing stops due to insufficient storage, SMP/E returns a code of 12.
- Terminating messages (Return code 16) - show that a situation occurred that forces SMP/E to stop processing.
- Terminating messages (Return code 20) - show that an internal SMP/E error caused SMP/E to stop processing. For example, there may be an interface problem between SMP/E modules or a data error on the SMPCSI data set.

## ACCEPT Return Codes

- 00—ACCEPT processing completed successfully.
- 04—ACCEPT processing completed with *possible* error conditions (warnings). No SYSMODs had their processing terminated.
  - ACCEPT invoked a system program to perform some work and the system program returned a nonzero, but still acceptable, return code. One of the following system programs could generate this return code:
    - Assembler (ASMBLR)
    - IEBCOPY - invoked to copy one or more modules, macros, or source modules, or to compress a data set
    - IEBUPDTE - invoked to update or replace source modules or macros
    - IMASPZAP - invoked to perform a ZAP operation
    - Linkage editor (IEWL).
- 08—ACCEPT processing completed, but processing errors occurred. At least one SYSMOD had its processing terminated. Possible error conditions are:
  - ACCEPT invoked a system program to perform some work and the system program returned a nonzero and unacceptable return code. One of the following system programs could generate this return code:
    - Assembler (ASMBLR)
    - IEBCOPY - invoked to copy one or more modules, macros, or source modules
    - IEBUPDTE - invoked to update or replace source modules or macros
    - IMASPZAP - invoked to perform a ZAP operation
    - Linkage editor (IEWL).
  - The affected SYSMOD entries have the ACCEPT and ERROR status indicators set in the distribution zone.

- SMP/E encountered an error while scanning IMASPZAP control statements. Check SMPOUT output for error messages to determine the cause of the problem. The affected SYSMOD entry has the ACCEPT and ERROR indicators set, although no update has been made to the module unless an EXPAND linkage editor control statement was included in the modification. In this case, the module has been link-edited to expand its size in the distribution library.
- An IMASPZAP VERIFY REJECT was encountered by the IMASPZAP program. Check SYSPRINT output for error messages to determine the cause of the problem. The affected SYSMOD entry has the ACCEPT and ERROR indicators set, although no update has been made to the module unless an EXPAND linkage editor control statement was included in the modification. In this case, the module has been link-edited to expand its size in the distribution library.
- A DD statement was missing. ACCEPT did not process any SYSMOD that required the missing DD statement.
- A SYSMOD specified in the SELECT operand list has an entry in the distribution zone that indicates that it has been superseded by another SYSMOD.
- A TXLIB or LKLIB member cannot be found. The affected SYSMOD entry has the ACCEPT and ERROR status indicators set in the distribution zone.
- A SYSMOD specified in the SELECT operand list was not found on the PTS, or in the GLOBAL zone of the SMPCSI.
- PEMAX was too small to process one or more SYSMOD entries and/or selected element entries being modified. If the latter situation is true, the affected SYSMOD entries might have the ACCEPT and ERROR status indicators set in the distribution zone. Check SMPOUT output for error messages to determine which SYSMODs and/or elements were affected.
- An error occurred while attempting to open a target system or distribution library. The affected SYSMOD entry might have the ACCEPT and ERROR status indicators set in the distribution zone.
- 12-ACCEPT processing terminated. Possible error conditions are:
  - A function SYSMOD was selected for processing and subsequently terminated before any updating of distribution libraries.
  - No SYSMODs met ACCEPT specifications.
  - A GETMAIN failure occurred during ACCEPT processing.
  - An error occurred while opening or closing an SMP/E data set.



- A syntax error was detected in the ACCEPT command.
- The ACCEPT command was not processed because a previous command returned a nonacceptable return code.
- A DD statement was missing.
- 16—A severe error occurred and SMP/E processing was terminated. Possible error conditions are:
  - IEBCOPY, invoked to compress a data set, returned a nonacceptable code. ACCEPT did not run, but the elements within the subject SYSMODs that were candidates for replacement may have been deleted from the appropriate distribution libraries.
 

*Note: The distribution libraries might be unusable. Examine the IEBCOPY output to determine the status of the data set when IEBCOPY failed.*
  - A severe error occurred while accessing an SMP/E data set.
  - An error occurred while writing a message.

## ACCEPT Error Recovery

After completion or abnormal termination of the ACCEPT function, examine SMPOUT and SYSPRINT to determine the relative success of the function. Note that partially applied SYSMODs have the ACCEPT and ERROR status indicators set in the SYSMOD entries on the distribution zone. Examine the reports if they have been produced.

You must rerun ACCEPT for a SYSMOD that failed during a previous ACCEPT. After an ACCEPT fails, SMP/E does not allow any function other than ACCEPT to be performed on that PTF. If you remove the ERROR status indicator in the distribution zone SYSMOD entry and attempt a subsequent RESTORE that will use some or all of the copies of the elements in the distribution libraries supposedly updated or replaced by that SYSMOD, unpredictable results should be expected. The following processing takes place:

- All linkage editor processing is repeated.
- All IEBCOPY processing is repeated.
- All macro and source module updating is repeated.
- All assemblies are repeated.

- All IMASPZAP processes are repeated. However, if any IMASPZAP process completed through the IMASPZAP REPLACE stage, or if any IMASPZAP process produced an IMASPZAP VERIFY REJECT in the previous ACCEPT, this rerun of ACCEPT will also fail. To correct this problem:
  1. For the modules involved in the SYSMOD, use the utility IEBTPCH to obtain the IMASPZAP control cards from the PTS by punching the SYSMOD.
  2. REJECT the SYSMOD from the PTS.
  3. Correct any IMASPZAP modification processed that caused a VERIFY REJECT.
  4. RECEIVE and ACCEPT the SYSMOD as corrected.

## APPLY Return Codes

- 00—APPLY processing completed successfully.
- 04—APPLY processing completed with *possible* error conditions. Possible error conditions are:
  - APPLY invoked a system program to perform some work and the system program returned a nonzero, but still acceptable, return code. One of the following system programs could generate this return code:
    - Assembler (ASMBLR)
    - IEBCOPY - invoked to copy modules, macros, or source modules, or to compress a data set
    - IEBUPDTE - invoked to update or replace source modules or macros
    - IMASPZAP - invoked to perform a ZAP operation
    - Linkage editor (IEWL).
  - The affected SYSMOD entries have the APPLY status indicator set in the target zone. Since the target system libraries might be unusable, correct the error and resubmit the job.
  - No assembler input could be found in either the target zone or the distribution library specified in the DISTSRC or ASMLIB operand list when APPLY attempted to reassemble a module because of a macro modification. The module was not reassembled, but the APPLY status indicator was set for the affected SYSMOD entries in the target zone.

- 08—APPLY processing completed, but processing errors were encountered. At least one SYSMOD had its processing terminated. Possible error conditions are:
  - APPLY invoked a system program to perform some work, and the system program returned a nonzero and unacceptable return code. One of the following system programs could generate this return code:
    - Assembler (ASMBLR)
    - IEBCOPY - invoked to copy modules, macros, or source modules
    - IEBUPDTE - invoked to update or replace source modules or macros
    - IMASPZAP - invoked to perform a ZAP operation
    - Linkage editor (IEWL).
  - The affected SYSMOD entries have the APPLY and ERROR status indicators set in the target zone.
  - SMP/E encountered an error while scanning IMASPZAP control statements. Check SMPOUT output for error messages to determine the cause of the problem. The affected SYSMOD entry has the APPLY and ERROR indicators set, although no update has been made to the module unless an EXPAND linkage editor control statement was included in the modification. In this case, the module was link-edited to expand its size and the load module was replaced in the target system library.
  - An IMASPZAP VERIFY REJECT was encountered by program IMASPZAP. Check SYSPRINT output for error messages to determine the problem. The affected SYSMOD entry has the APPLY and ERROR indicators set, although no update made to the module unless an EXPAND linkage editor control statement was included in the modification. In this case, the module was link-edited to expand the size, and the load module replaced in the target system library.
  - A DD statement was missing. APPLY did not process any SYSMOD that required the missing DD statement.
  - A SYSMOD specified in the SELECT operand list has an entry in the target zone that indicates it has been superseded by another SYSMOD.
  - A TXLIB or LKLIB member cannot be found. The affected SYSMOD entry has the APPLY and ERROR status indicators set in the target zone.
  - A SYSMOD specified in the SELECT operand list was not found on the PTS.

- PEMAX was too small to process one or more SYSMOD entries and/or selected element entries being modified. If the latter situation is true, the affected SYSMOD entries may have the APPLY and ERROR status indicators set in the target zone. Check SMPOUT output for error messages to determine which SYSMODs and/or elements were affected.
- An error occurred while attempting to open a target system or distribution library. The affected SYSMOD entry may have the APPLY and ERROR status indicators set in the target zone.
- 12—APPLY processing terminated. Possible error conditions are:
  - A function SYSMOD was selected for processing and terminated before any updating of target system libraries.
  - No SYSMODs met APPLY specifications.
  - A GETMAIN failure occurred during APPLY processing.
  - An error occurred while opening or closing an SMP/E data set.
  - A syntax error was detected in the APPLY command.
  - The APPLY command was not processed because a previous command returned a nonacceptable return code.
  - A DD statement was missing.
- 16—A severe error occurred and SMP/E processing was terminated. Possible error conditions are:
  - IEBCOPY, invoked to compress a data set, returned a nonacceptable return code. APPLY processing did not occur, but the elements within the subject SYSMODs that were candidates for replacement may have been deleted from the appropriate target system libraries.

*Note: The target system libraries might be unusable. Examine the IEBCOPY output to determine the status of the data set when IEBCOPY failed.*

  - A severe error occurred while accessing an SMP/E data set.
  - An error occurred while writing a message.

## APPLY Error Recovery

After completion or abnormal termination of the APPLY function, examine SMPOUT and SYSPRINT output to determine the relative success of the function. Note that partially applied SYSMODs have the APPLY and ERROR status indicators set in the SYSMOD entries on the target zone.

You can rerun APPLY for a SYSMOD that has failed by correcting any conditions that caused the SYSMOD to be terminated. If a SYSMOD that failed APPLY processing had inline JCLIN that was successfully processed, you should specify the NOJCLIN keyword with that SYSMOD-ID as an operand on the APPLY command for the subsequent reapplication. The following processing takes place:

- All linkage editor processing is repeated.
- All IEBCOPY processing is repeated.
- All macro and source updating is repeated.
- All assemblies are repeated.
- All IMASPZAP processes are repeated. However, if any IMASPZAP processing completed through the IMASPZAP REPLACE stage, or if any IMASPZAP process produced an IMASPZAP VERIFY REJECT in the previous APPLY, this rerun of APPLY will fail. To correct this problem:
  1. Use the utility IEBTPCH to obtain the IMASPZAP control cards from the PTS for the modules involved in the SYSMOD by punching the SYSMOD.
  2. REJECT the SYSMOD from the PTS.
  3. Correct any IMASPZAP processed that caused a VERIFY REJECT.
  4. RECEIVE and APPLY the corrected SYSMOD.

Rerun APPLY with CHECK to determine the appropriate actions.

## CLEANUP Return Codes

- 00—CLEANUP processing completed successfully.
- 08—CLEANUP processing completed, but a processing error occurred. A possible error condition is:
  - There was no corresponding target zone entry for an entry in one of the data sets being cleaned up (SMPMTS, SMPSTS, or SMPSCDS).
- 12—CLEANUP processing stopped. A possible error condition is:
  - SMP/E could not open one of the data sets it needed: the target zone, distribution zone, SMPMTS, SMPSTS, or SMPSCDS.
- 16—A severe error occurred, and SMP/E processing stopped.

## CLEANUP Error Recovery

If an error occurs during CLEANUP processing, follow the directions in the “Programmer Response” section of each message.

## CONVERT Return Codes

- 00—CONVERT processing completed successfully.
- 04—CONVERT processing completed, but with unexpected results. The error condition is:
  - SMP/E may have only partially converted the specified entry due to an error that SMP/E found when it was converting that entry.
- 12—CONVERT processing stopped. Possible error conditions are:
  - You did not specify the SMP data sets you want SMP/E to convert.
  - The zone type to receive data is not correct for the DD statement that SMP/E specified on the CDS, ACDS, PTS, CRQ, or ACRQ CONVERT operand.
  - An error occurred while SMP/E was reading an entry from the SMP data set or storing the converted entry on the SMP data set.
- 16—A severe error occurred, and SMP/E processing stopped.

## CONVERT Error Recovery

If an error occurs during CONVERT processing, follow the directions in the “Programmer Response” section of each message.

## GENERATE Return Codes

- 00—GENERATE processing completed successfully.
- 04—GENERATE processing completed, but the system issued warning messages that indicate *potential* errors when the system runs the generated jobstream. Refer to the messages to find the cause of the error.

- 08—GENERATE processing completed, but SMP/E found potential errors while the system analyzed the various target zone entries. The conditions that SMP/E reports may cause problems when SMP/E runs the generated jobs. Review each message, find the cause of the message, and follow the directions in the “Programmer Response” section of each message. Possible error conditions are:
  - SMP/E did not find the JOBCARD member in the specified data set.
  - SMP/E did not find an ASSEM or SRC entry in the target zone.
  - SMP/E could not assemble the identified module.
  - The MOD, MAC, or SRC is missing a subentry (FMID or DISTLIB).
  - The LMOD entry had no MOD entries to show that the module was a part of that load module.
- 12—GENERATE processing terminated. A possible error condition is:
  - An operand was missing from the command.
- 16—An I/O error occurred and processing stopped.

## GENERATE Error Recovery

If an error occurs during GENERATE processing, follow the directions in the “Programmer Response” section of each message.

## JCLIN Return Codes

- 00—JCLIN processing completed successfully.
- 04—JCLIN processing completed, but a premature end-of-file statement was encountered for the JCLIN input data set.
- 08—JCLIN processing terminated because a syntax error was encountered in the JCLIN input.
- 12—JCLIN processing terminated. Possible error conditions are:
  - A syntax error existed in the JCLIN command.
  - Not enough storage was available.
  - Out-of-space condition on the target zone.
  - PEMAX was too small to process one or more entries on the target zone.
  - A DD statement was missing.
  - The JCLIN command was not processed because a previous command returned a nonacceptable return code.
- 16—A severe error occurred, and SMP/E processing was terminated.

## JCLIN Error Recovery

If an error occurs in the JCLIN data set, examine SMPOUT output to determine the job, job step, and record that caused the error. Correct the problem and rerun JCLIN.

If the error occurred in your user-specified JCLIN input data set, see “JCLIN Processing” in the *SMP/E Reference*, for further information.

If an out-of-space condition occurred on the SMPCSI during JCLIN processing, see “Resolving Direct Access Storage Problems” in the *SMP/E Diagnosis Guide* for information on how to handle the problem, and then rerun JCLIN.

## LIST Return Codes

- 00—LIST processing completed successfully.
- 04—LIST processing completed, but at least one requested item was not listed. Possible error conditions are:
  - An entry specified in the LIST command was not found on the data set being listed.
  - An entry specified in the LIST command was found, but was not eligible as requested. For example, the SYSMOD-ID requested in “LIST SYSMOD(UZ00004) FUNCTION.” was found, but it was not a function SYSMOD.
  - PEMAX was too small to process a selected entry.
  - A DD statement was missing.
- 08—LIST processing terminated. The error condition is:
  - The LIST command was specified without any accompanying keywords.
- 12—LIST processing terminated. Possible error conditions are:
  - A syntax error occurred in the LIST command.
  - Not enough storage was available.
  - An invalid date range was specified in the LIST LOG command.
  - A DD statement was missing.
- 16—A severe error occurred, and SMP/E processing terminated.



## LIST Error Recovery

If an error occurs during LIST processing, follow the directions in the “Programmer Response” section of each message.

## LOG Return Codes

- 00—LOG processing completed successfully and without errors.
- 12—LOG processing terminated. Possible error conditions are:
  - A syntax error occurred in the LOG command.
  - A DD statement was missing.
  - The LOG command was not processed because a previous command returned a nonacceptable return code.
- 16—An I/O error occurred and processing was terminated.

## RECEIVE Return Codes

- 00—RECEIVE processing completed successfully.
- 04—RECEIVE processing completed with *possible* error conditions:
  - The HMASMUXD user-written exit routine took some action for at least one SYSMOD.
- 08—RECEIVE processing completed, but errors were encountered and processing terminated for at least one SYSMOD. Possible error conditions are:
  - A selected SYSMOD was not found on the SMPPTFIN dataset and was not processed.
  - PEMAX was too small to process a SYSMOD.
  - A GETMAIN failure occurred while processing a SYSMOD, resulting in the termination of processing for that SYSMOD.
  - The modification name specified on the ++MACUPD, ++SRCUPD, or ++UPDTE modification control statement was different from the modification name specified on the IEBUPDTE “./ CHANGE” control statement. This resulted in the termination of processing for the affected SYSMOD.

- RECEIVE processing detected a syntax error in a SYSMOD while scanning the modification control statements for a SYSMOD in the SMPPTFIN dataset. This resulted in the termination of processing for that SYSMOD. Syntax errors include validation check errors.
- RECEIVE processing encountered an end-of-file on the SMPPTFIN dataset during the processing of a SYSMOD, resulting in the termination of processing for that SYSMOD.
- A return code from the HMASMUXD user-written exit routine required RECEIVE to stop processing a SYSMOD.
- Two modification control statements within a SYSMOD referred to the same element, resulting in the termination of processing for that SYSMOD.
- The same SYSMOD-ID was specified more than once on the TO operand of a single ++ASSIGN MCS statement.
- 12—RECEIVE processing terminated. Possible error conditions are:
  - A GETMAIN failure occurred that caused the termination of RECEIVE processing.
  - A return code from the HMASMUXD user-written exit routine required RECEIVE processing to terminate.
  - A syntax error was detected in the RECEIVE command.
  - A failure occurred during STOW processing while attempting to place a SYSMOD or MCS entry on the PTS.
  - None of the SYSMODs specified in the SELECT operand list were found, or no SYSMODs were found in the SMPPTFIN dataset.
  - A DD statement was missing.
  - The RECEIVE command was not processed, because a previous command returned a nonacceptable return code.
- 16—A severe error occurred and SMP/E processing was terminated. Possible error conditions are:
  - An I/O error occurred.
  - A return code from the HMASMUXD user-written exit routine required the termination of all processing.

## RECEIVE Error Recovery

If RECEIVE issued the message “GIM344 SYSMOD sysmod SUCCESSFULLY RECEIVED,” the SYSMOD was completely stored, and the SYSMOD and MCS entries in the PTS have been created.

If you are unsure about the status of a SYSMOD, issue LIST SYSMOD for the GLOBAL zone to obtain a listing of the SYSMODs on the PTS. If the SYSMOD entry is present with the ERROR status indicated, the SYSMOD is not ready to be applied or accepted. If you are still unsure if the SYSMOD was successfully received, use the REJECT command to delete the SYSMOD. After correcting any conditions that might have caused problems during the previous RECEIVE pass, reissue RECEIVE for the SYSMOD.

If an out-of-space condition occurs on the PTS during RECEIVE processing, see “Resolving Shortages of Direct Access Storage” in the *SMP/E Diagnosis Guide* for information on handling the problem. Then rerun RECEIVE.

## REJECT Return Codes

- 00—REJECT processing completed successfully.
- 04—REJECT processing completed with *possible* error conditions:
  - IEBCOPY, invoked to compress a data set, returned an acceptable, but nonzero, return code.
  - A SYSMOD specified in the SELECT operand list was processed by APPLY or ACCEPT. The SYSMOD is not processed by REJECT.
- 08—REJECT processing completed, but errors were encountered and processing terminated for at least one SYSMOD. A possible error is:
  - PEMAX was too small to process a SYSMOD entry on the SMPCSI, and that SYSMOD has not been rejected.
- 12—REJECT processing terminated. Possible error conditions are:
  - No SYSMODs met REJECT specifications.
  - GETMAIN failed during REJECT processing.
  - An error occurred while opening or closing an SMP/E data set.
  - SMP/E detected a syntax error in the REJECT command.
  - A DD statement was missing.
  - The REJECT command was not processed because a previous command returned a nonacceptable return code.

- 16—A severe error occurred and SMP/E processing terminated. Possible error conditions are:
  - IEBCOPY, invoked to compress a data set, returned a nonacceptable return code. REJECT processing did not occur.
  - A severe error occurred while accessing an SMP/E data set.
  - An error occurred while writing a message.

## REJECT Error Recovery

If a failure occurs during REJECT processing, issue the REJECT command for those SYSMODs that were not successfully rejected. If a function SYSMOD is being rejected, check the SMPCSI GLOBAL zone definition entry to see if the FMID subentry for that SYSMOD-ID has been deleted. If it was not and should have been (that is, the SYSMOD was never applied or accepted), use the UCLIN function to delete the FMID subentry.

## REPORT Return Codes

- 00—REPORT processing completed successfully.
- 08—REPORT processing completed, but processing errors occurred. A possible error conditions is:
  - SMP/E could not find an FMID or FMIDSET that matched the value specified on the FORFMID operand.
- 12—The REPORT command terminated. Possible error conditions are:
  - The ZONESET entry is not defined in the GLOBAL zone.
  - There is no ZONEINDEX for one of the zones in the ZONESET.
  - One of the zone names in the FORZONE operand is not included in the ZONESET.
  - The zones in the ZONESET are not all the same type (target or distribution).
  - Output is to be written to SMPPUNCH (NOPUNCH was not specified), but the DD statement or DDDEF entry needed for the SMPPUNCH data set was missing.

## REPORT Error Recovery

If an error occurs during REPORT processing, follow the directions in the "Programmer Response" section of each message.

## RESTORE Return Codes

- 00—RESTORE processing completed successfully.
- 04—RESTORE processing completed with *possible* error conditions:
  - RESTORE invoked a system program to perform some work and the system program returned a nonzero, but still acceptable, return code. One of the following system programs could have generated this return code:
    - Assembler (ASMBLR)
    - IEBCOPY - invoked to copy a module, macro, or source module, or to compress a data set
    - IEBUPDTE - invoked to update a macro or source module
    - Linkage Editor - (IEWL).
  - The affected SYSMOD is restored and is marked RESTORE in the target zone.
  - During RESTORE processing, assembly input for a selected module was not found on the target zone.
- 08—RESTORE processing completed, but processing errors were encountered that resulted in the termination of at least one SYSMOD. Possible error conditions are:
  - RESTORE invoked a system program to perform some work and the system program returned a nonzero unacceptable return code. One of the following system programs could have generated this return code:
    - Assembler (ASMBLR)
    - IEBCOPY - invoked to copy a module, macro, or source module
    - IEBUPDTE - invoked to update a macro or source module
    - Linkage editor (IEWL).

- The affected SYSMOD entries have the RESTORE and ERROR status indicators set in the target zone.
  - A DD statement was missing. RESTORE did not process any SYSMOD that requires the missing DD statement.
  - A SYSMOD selected for RESTORE processing was never processed by APPLY, but a SYSMOD entry exists that was created by the processing of another SYSMOD that superseded it. RESTORE processing did not affect the superseded and superseding SYSMODs.
  - RESTORE processing requires an element entry that cannot be found on the target zone. RESTORE processing terminated for all affected SYSMODs.
  - PEMAX was too small to process a selected SYSMOD or element entry.
  - A SYSMOD selected for RESTORE has already been processed by ACCEPT. RESTORE did not process the affected SYSMOD.
  - During RESTORE processing an error occurred while opening a required data set. RESTORE processing was terminated for all affected SYSMODs.
- 12—RESTORE processing terminated. Possible error conditions are:
    - No SYSMODs met RESTORE specifications.
    - GETMAIN failed during RESTORE processing.
    - An error occurred while opening an SMP/E data set.
    - A syntax error was detected in the RESTORE command.
    - A DD statement was missing.
    - The RESTORE command was not processed because a previous command returned a nonacceptable return code.
  - 16—A severe error occurred, and SMP/E processing was terminated. Possible error conditions are:
    - IEBCOPY, invoked to compress a data set, returned a nonacceptable return code. RESTORE processing did not occur, but the modules within the subject SYSMODs that were candidates for replacement during RESTORE were deleted from the appropriate target system libraries.
 

*Note: The target system libraries might be unusable; that is, the system or some of its components might not run.*
    - A severe error occurred while deleting members from a target system library before compression of that library.
 

*Note: The target system libraries might be unusable; that is, the system or some of its components might not run.*
    - An error occurred while writing a message.
    - A severe error occurred while accessing an SMP/E data set.

## RESTORE Error Recovery

After the RESTORE function completes, examine SMPOUT and SYSPRINT output to determine the relative success of the function. Note that partially restored SYSMOD entries have the RESTORE and ERROR status indicators set in the target zone.

You should rerun RESTORE for a SYSMOD that failed during previous RESTORE processing. The following processing takes place:

- All linkage editor processing is repeated.
- All IEBCOPY processing is repeated.
- All assemblies are repeated.
- All IEBUPDTE processing is repeated.

If an out-of-space condition occurs on any library during RESTORE processing, see “Resolving Direct Access Storage Shortage Problems” in the *SMP/E Diagnosis Guide* for information on how to handle the problem. Then rerun RESTORE.

## SET Return Codes

- 00—SET processing completed successfully.
- 08—SET processing completed, but errors were encountered.
- 12—SET processing stopped. Possible error conditions are:
  - SMP/E could not open the zone.
  - You specified a zone that does not exist.
- 16—An I/O error occurred when SMP/E was trying to open a zone and SMP/E processing stopped.

## SET Error Recovery

If an error occurs when SMP/E is processing a SET statement, correct the data (zone name, OPTIONS entry, or data in zone definition entry). Follow the directions in the “Programmer Response” section of each message.

## UCLIN Return Codes

- 00—UCLIN processing completed successfully.
- 04—UCLIN processing completed, but with unexpected results.
  - SMP/E met an end-of-file statement in the SMPCNTL data set before it processed an ENDUCL command.
  - No UCL statement followed the UCLIN command.
- 08—UCLIN processing completed with errors. Possible error conditions are:
  - SMP/E detected a syntax error in at least one UCL input statement.
  - At least one UCL statement does not meet conditions for update.
- 12—UCLIN processing stopped.
  - SMP/E detected a syntax error in the UCLIN command.
  - SMP/E did not do any processing due to an unacceptable return code from a previous command.
  - A DD statement was missing for a required data set.
- 16—A severe error occurred and SMP/E processing stopped.

## UCLIN Error Recovery

If a failure occurs when processing a UCL statement, follow the directions in the “Programmer Response” section of each message.

## ZONECOPY Return Codes

- 00—ZONECOPY processing completed successfully.
- 04—ZONECOPY processing completed, but with unexpected results.
- 12—ZONECOPY processing stopped. Possible error conditions are:
  - The name of the zone in the SET BOUNDARY command and the name of the receiving zone are different.
  - The input and receiving zone have the same name.
  - The input or receiving zone is GLOBAL.



- The combination of input and receiving zone types is invalid.  
Valid combinations are:
  - Distribution to distribution
  - Target to target
  - Distribution to target.
- The input and receiving zones exist in the same CSI.
- When SMP/E opened the input zone for read access and opened the receiving zone for update processing, the open failed.
- Zone already exists.
- 16—An I/O error occurred, and processing stopped.

## ZONECOPY Error Recovery

If an error occurs during ZONECOPY processing, follow the directions in the “Programmer Response” section of each message.

## ZONEDELETE Return Codes

- 00—ZONEDELETE processing completed successfully.
- 04—ZONEDELETE processing completed, but SMP/E could not acquire the GLOBAL zone to delete the zone index entry.
- 12—ZONEDELETE processing stopped when SMP/E met an I/O error.
- 16—A severe error occurred, and SMP/E processing stopped.

## ZONEDELETE Error Recovery

If SMP/E could not delete the GLOBAL zone index entry, use UCLIN to delete the zone index entry. Follow the directions in the “Programmer Response” section of each message.

## ZONEEDIT Return Codes

- 00—ZONEEDIT processing completed successfully.
- 04—ZONEEDIT processing completed with *possible* error conditions (warnings). No SYSMODs had their processing terminated.
- 08—ZONEEDIT processing completed, but processing errors were encountered. Possible error conditions are:
  - A changed data set name would have been more than 44 characters.
  - SMP/E met an end-of-file statement in the SMP\_CNTL data set before it processed an ENDZONEEDIT command.

- 12—ZONEEDIT processing stopped. Possible error conditions are:
  - The entry type, field name, or field value was not valid.
  - The entry does not exist in the specified zone.
  - When SMP/E opened the zone for update processing, the opening failed.
  - The same field name was specified more than once on a single change statement.
- 16—A severe error occurred, and SMP/E processing stopped.

## **ZONEEDIT Error Recovery**

If an error occurs during ZONEEDIT processing, follow the directions in the “Programmer Response” section of each message.

## **ZONEEXPORT Return Codes**

- 00—ZONEEXPORT processing completed successfully.
- 04—ZONEEXPORT processing completed, but the GLOBAL zone was not available for update processing.
- 12—ZONEEXPORT processing stopped. Possible error conditions are:
  - The input was the GLOBAL zone, but you specified the PURGE operand.
  - The SET BOUNDARY zone name and the input zone name were not the same.
  - SMP/E opened the input zone for read access when the PURGE parameter was not specified, but the OPEN failed.
  - SMP/E opened the zone for update when the PURGE was specified, but the OPEN failed.
  - The OPEN was successful, but the position to the first record failed.
  - The first record in the input zone, other than the GLOBAL zone, was not a zone definition record (category 10 relation 1000).
  - The output data set OPEN failed.
  - The OUTFILE ddname is the same as the zone name specified on the SET command.
- 16—An I/O error occurred, and SMP/E processing stopped.

## ZONEEXPORT Error Recovery

If an error occurs during ZONEEXPORT processing, follow the directions in the “Programmer Response” section of each message.

## ZONEIMPORT Return Codes

- 00—ZONEIMPORT processing completed successfully.
- 04—ZONEIMPORT processing completed, but with unexpected results.
  - The receiving zone type is target, and the input zone type is DLIB. (SMP/E changes the type in the zone definition record to target).
  - The receiving zone name and the name from the zone definition record are different. (SMP/E inserts the receiving zone name into subject 1 field of the zone definition record).
- 12—ZONEIMPORT processing stopped. Possible error conditions are:
  - The SET to zone and receiving zone names are different.
  - The receiving zone already exists.
  - The input data set could not be opened.
  - The OPEN was successful, but the zone definition record is not valid for a zone import.
  - The receiving zone type and the type in the zone definition record are an invalid type change.
  - The name of the zone that SMP/E is importing and the name from the zone definition record are different.
  - The OPEN of the receiving zone for update processing failed.
  - The receiving zone is GLOBAL and SMP/E already defined a zone in the CSI data set.
  - An end-of-file condition occurred on the input data set before SMP/E read the record with a zone encode value of hex FFFF.
  - The INFILE ddname is the same as the zone name specified on the SET command.

## **ZONEIMPORT Error Recovery**

If an error occurs during ZONEIMPORT processing, follow the directions in the “Programmer Response” section of each message.

## **ZONEMERGE Return Codes**

- 00—ZONEMERGE processing completed successfully without errors.
- 12—ZONEMERGE processing stopped, but with unexpected results.
  - SMP/E met an I/O error.
  - The job was out of storage.
- 16—A severe error occurred, and SMP/E processing stopped.

## **ZONEMERGE Error Recovery**

If the message shows that the job was out of core, resubmit the job using a larger REGION. In addition, follow the directions in the “Programmer Response” section of each message.

## **ZONERENAME Return Codes**

- 00—ZONERENAME processing completed successfully.
- 12—ZONERENAME processing stopped, but with unexpected results.
  - SMP/E did not find the name of the zone that it is renaming in the zone index.
  - SMP/E found the new name of the zone that SMP/E is renaming in the zone index.
  - SMP/E found the new zone name already in the data set.
  - SMP/E could not rename the old zone name to the new zone name.
  - SMP/E found index the name of the zone that it is renaming in the zone index, but it was not defined.
  - The data set that has the zone that SMP/E is renaming could not be opened.
  - The old zone name was the same as the new zone name.
  - You specified mutually exclusive operands together on the command.

- You specified the zone name GLOBAL as either the new zone name or the old zone name.
- You specified TOTYPE, but either the GLOBAL zone ZONEINDEX entry or the zone definition for the old zone name, is the target type.
- 16 A severe error occurred, and SMP/E processing stopped.

## **ZONERENAME Error Recovery**

If an error occurs during ZONERENAME processing, follow the directions in the "Programmer Response" section of each message.

## Chapter 2. SMP/E Messages

This chapter lists the SMP/E diagnostic messages. The messages are in alphanumeric order.

### Message Format

Messages are in the format of *GIMnnns yy text*, where:

- *nnn* is the message serial number.
- *s* is the severity code, as follows:
  - 0 — Informational message (return code = 0)
  - 1 — Warning message (return code = 4)
  - 2 — Error message (return code = 8)
  - 3 — Severe error message (return code = 12)
  - 4 — Terminal error message (return code = 16).

SMP/E sets the severity code of a message and the return code at the same time. If a message does not cause SMP/E to set a return code, the severity code of that message is 0. Each message has its own severity code.

- *yy* is the severity highlighting code, as follows:
  - Blanks    Severity 0 and 1 messages
  - \*\*        Severity 2, 3, and 4 messages.
- *text* is the message text. Optional text is indicated by brackets.
- “|” indicates an “or”. Only one out of each group of items separated by a vertical bar will appear in a message or can be chosen.

# GIM2XX

## Diagnostic Messages

### GIM201s *command* FAILED FOR LIBRARY SPECIFIED BY *lib*

**Explanation:**

- *command* - OPEN or CLOSE
- *lib* - ddname of the library that SMP/E could not open or close.

**System Action:** Command processing stops. The messages that follow show the action SMP/E takes.

**Programmer Response:** Do one of the following:

- If OPEN failed, check for the following:

- A missing DD statement
- An incorrect data set name.

Fix the problem and rerun the job.

- If CLOSE failed, rerun the job. If CLOSE continues to fail, SMP/E needs data set maintenance.

### GIM202s UNABLE TO OBTAIN STORAGE FOR WORK AREAS

**Explanation:** SMP/E could not allocate internal tables because of insufficient storage.

**System Action:** The messages that follow in the output listing show SMP/E's actions.

**Programmer Response:** Do one of the following:

- Increase the REGION parameter on the EXEC statement.
- Decrease the number of SYSMODs that SMP/E is processing in this run.
- Decrease the PEMAX value in the OPTIONS entry being used.

Rerun the job.

### GIM203s SYNTAX ERROR IN { *type|utility* INPUT | EXEC PARM | ZONEEDIT } STATEMENT AT COL *nn*

**Explanation:**

- *type* - CONTROL or UCL
- *utility* - LINKAGE EDITOR, ASSEMBLER, or IEBCOPY
- *nn* - column number.

**When *type* = CONTROL:**

SMP/E found a syntax error in the modification control statement or in the command at the specified column. Note that the message shows the line immediately before the one with the syntax error.

**System Action:** One of the following occurs:

- If the error was in the modification control statement, processing stops for this SYSMOD.
- If the error was in an SMP/E command, processing stops for the command.

**Programmer Response:** Do the following:

1. Check the keyword on the modification control statement or command.
2. Correct the syntax error and rerun the job.

**When *type* = UCL:**

SMP/E detected a syntax error in the UCL statement at the specified column.

**System Action:** SMP/E ignores the UCL statement. Processing continues with the next UCL statement.

**Programmer Response:** Correct the UCL statement and rerun the statement.

**When *utility* = LINKAGE EDITOR, ASSEMBLER, or IEBCOPY:**

During JCLIN processing, SMP/E found a syntax error on an input statement for the job step it is scanning.

**System Action:** The scan stops.

**Programmer Response:** Correct the error and rerun the job.

**When SMP/E produces EXEC PARM:**

SMP/E found an invalid parameter on the EXEC statement.

**System Action:** SMP/E processing stops.

**Programmer Response:** Correct the problem and rerun the job.

**When SMP/E produces ZONEEDIT:**

SMP/E found an invalid parameter in a ZONEEDIT change statement.

**System Action:** SMP/E processing stops.

**Programmer Response:** Correct the problem and rerun the job.



# GIM2XX

**GIM205s** {GIMSMP|*command*|UNKNOWN} PROCESSING COMPLETED - HIGHEST RETURN CODE IS *rc*

**Explanation:**

- *command* - command that SMP/E is processing
- *rc* - return code for that function.

One of the following is true:

- If the message shows GIMSMP, *rc* is the return code for the job step.
- If the message shows UNKNOWN, SMP/E could not find the command that was being processed.

**System Action:** The return code sets the system action.

**Programmer Response:** See the return codes for *command* in Chapter 1 to determine the success or failure of the command that SMP/E ran.

**GIM206s** USER EXIT RETURN CODE INDICATES TERMINATION OF {SYSMOD|*command*|SMP}

**Explanation:**

- *command* - current command.

The return code from a user exit routine showed one of the following:

- The SYSMOD in process stopped.
- The current command stopped.
- All SMP/E processing stopped.

**System Action:** Command processing stops.

**Programmer Response:** Find out why the user exit routine ended the request. Check that the exit routine issued the correct return code for this request.

**GIM207s** UNKNOWN USER EXIT RETURN CODE (*rc*) - {*command* | SMP} TERMINATED

**Explanation:**

- *rc* - user exit return code value
- *command* - current command.

The user exit routine issued an undefined return code.

**System Action:** Based on the exit routine SMP/E called, either the current command stops or SMP/E stops.

**Programmer Response:** Check the logic of the user exit routine to make sure it only returns defined codes.

**GIM214s STORE FAILED FOR *type name* [ON *lib* LIBRARY IN *zonename* ZONE]****Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of data set
- *zonename* - name of zone that SMP/E is updating.

SMP/E cannot store the directory entry or zone information for this entry. A previous message in the output listing shows the reason.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Find the cause of the error from the previous messages. Correct the error and rerun the job.

**GIM216s UPDATE {FAILED|SUCCESSFUL} - MEMBER=*name* - LIBRARY=*lib* - SYSMOD=*sysmod* - RETURN CODE=*rc*****Explanation:**

- *name* - entry name
- *lib* - ddname of data set
- *sysmod* - SYSMOD-ID
- *rc* - return code from IEBUPDTE.

An IEBUPDTE run completed for *name* in library *lib* with return code *rc*. The element *name* was part of SYSMOD *sysmod*.

**System Action:** Processing continues, as shown by the messages that follow in the output listing.

**Programmer Response:** Do one of the following:

- If IEBUPDTE failed, look at the output to find the cause of the error.
- If IEBUPDTE error message "MEMBER NAME NOT FOUND" was also issued, be sure that the member exists on the DISTLIB and/or SYSLIB, as shown in the target zone or in the modification control statement.

**GIM218s SUCCESSFULLY STORED *type name* [ON *lib* LIBRARY IN *zonename* ZONE]****Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of the SMP/E data set
- *zonename* - name of zone being updated.

SMP/E successfully stored or restored *name*.

# GIM2XX

**System Action:** SYSMOD processing continues.

**Programmer Response:** None.

## GIM219s PEMAX EXCEEDED FOR *type name* [ON *lib* LIBRARY | IN *zonename* ZONE]

### Explanation:

- *type* - entry type
- *name* - entry name
- *lib* - ddname of SMP/E data set
- *zonename* - name of zone being updated.

SMP/E could not create, update, or list the entry because the SYSMOD entry needed a PEMAX value greater than the value in the current OPTIONS entry.

*Note:* This message may indicate that the entry has invalid data.

**System Action:** SYSMOD or function processing stops.

**Programmer Response:** Do one of the following:

- Use UCLIN to increase the PEMAX value in the OPTIONS entry.

*Note:* Do not decrease the PEMAX value after SMP/E processes SYSMODs with a larger PEMAX, or existing SYSMOD entries may be too large for SMP/E to process.

- Ensure that the OPTIONS entry containing the increased PEMAX value is the same as the OPTIONS entry named in the zone definition entry.
- If the entry has invalid data, either correct the data or restore the data set and rerun the job.

## GIM224s SUCCESSFULLY DELETED *type name* [ON *lib* LIBRARY | FROM *zonename* ZONE]

### Explanation:

- *type* - entry type
- *name* - entry name
- *lib* - ddname of the data set
- *zonename* - name of zone that SMP/E is updating.

SMP/E successfully deleted *name* from the named library or zone.

**System Action:** Processing continues.

**Programmer Response:** None.

**GIM226s** *command* PROCESSING TERMINATED FOR SYSMOD *sysmod***Explanation:**

- *command* - RECEIVE, REJECT, APPLY, RESTORE or ACCEPT
- *sysmod* - SYSMOD-ID.

A preceding message describes the reason for the failure. SMP/E found the error for one SYSMOD only. SMP/E will continue to process other SYSMODs.

*Note:* You may find additional information in the LOG data set.

**System Action:** SYSMOD or function processing stops.

**Programmer Response:** Do the following:

1. Check previous messages to determine the cause of error.
2. Correct the error and rerun the job.

**GIM227s** *command* PROCESSING SUCCESSFULLY COMPLETED FOR SYSMOD *sysmod***Explanation:**

- *command* - ACCEPT, APPLY, REJECT, or RESTORE
- *sysmod* - SYSMOD-ID.

Function processing successfully completed.

**System Action:** None.

**Programmer Response:** None.

**GIM228s** IEANUC01 NOT FOUND ON NUCLEUS LIBRARY

**Explanation:** SMP/E did not find the nucleus, IEANUC01, on the nucleus library because of a BLDL operation.

**System Action:** Processing for all SYSMODs that affect IEANUC01 stops.

**Programmer Response:** Do one of the following:

- Create IEANUC01.
- Specify a different NUCLEUS DD statement.

## GIM2XX

### GIM229s CONTROL STATEMENT IGNORED DUE TO PREVIOUS ERROR

**Explanation:** SMP/E did not process this command because of an error described in a previous message. SMP/E did, however, check the command for syntax errors.

**System Action:** Processing continues with the next statement.

**Programmer Response:** Correct the error in the previous command and rerun the job.

### GIM231s IMASPZAP CONTROL STATEMENT ERROR IN MODULE: *mod* FOR SYSMOD *sysmod*

**Explanation:**

- *mod* - module name
- *sysmod* - SYSMOD-ID.

SMP/E detected a syntax error in the IMASPZAP statement for *mod* in *sysmod*. The statement with the error may be in the line before this message.

**System Action:** SYSMOD processing stops. Processing continues with the next SYSMOD.

**Programmer Response:** Correct the syntax error and rerun the GIMSMP step.

### GIM234s BLDL FAILED FOR PROGRAM *program* REQUIRED FOR GIMSMP EXECUTION

**Explanation:**

- *program* - program name.

SMP/E could not find *program*, which it needs to run GIMSMP.

**System Action:** Step processing stops. SMP/E processing also stops if you specified an invalid utility name in the GLOBAL zone UTILITY entry being used.

**Programmer Response:** Do one of the following:

- If program *program* is not present, add it to the JOBLIB, STEPLIB, or LINKLIB.
- If you specified an invalid utility name, correct it and rerun the job.

### GIM237s ZAP {VERIFY | REPLACE} PASS {FAILED | SUCCESSFUL} - MOD = *mod* - LMOD = *lmod* - LIBRARY = *lib* - SYSMOD = *sysmod* - RETURN CODE = *rc*

**Explanation:**

- *mod* - module name
- *lmod* - load module name
- *lib* - library name
- *sysmod* - SYSMOD-ID
- *rc* - return code.

IMASPZAP completed for module *mod* in load module *lmod* in library *lib* with a return code of *rc*. Module *mod* was part of SYSMOD *sysmod*.

**System Action:** SYSMOD processing stops if the return code is nonzero and one of the following is true:

- The return code is greater than the return code you specified.
- The return code is greater than the default return code.

**Programmer Response:**

1. Make sure the default utility return code or the return code you specified in the UTILITY entry is the one you want to use. Make sure the OPTIONS entry you are using contains the name of that UTILITY entry.
2. Check the output from IMASPZAP to find the cause of the error. Fix the error and rerun the job.

GIM238s COPY {FAILED | SUCCESSFUL} - MOD = *mod* - LMOD = *lmod* - LIBRARY = *lib* - SYSMOD = *sysmod* - RETURN CODE = *rc*

**Explanation:**

- *mod* - module name
- *lmod* - load module name
- *lib* - library name
- *sysmod* - SYSMOD-ID
- *rc* - return code.

IEBCOPY completed for module *mod* in load module *lmod* in library *lib* with a return code of *rc*. Module *mod* was part of SYSMOD *sysmod*.

Multiple SYSMODs might have LMODs copied in one invocation; therefore, some SYSMODs might have modules successfully copied, even though an error code resulted.

- This message shows that all the modules and/or load modules that SMP/E handled during this invocation of IEBCOPY failed, although only one module may have an error.
- The message may also appear for modules within a SYSMOD that SMP/E never copied because other modules in the SYSMOD were in error.

**System Action:** SYSMOD processing stops if the return code is nonzero and one of the following is true:

- The return code is greater than the return code you specified.
- The return code is greater than the default return code.

# GIM2XX

## Programmer Response:

1. Make sure the default utility return code, or the return code you specified in the UTILITY entry, is the one you want to use. Make sure the OPTIONS entry you are using contains the name of that UTILITY entry.
2. If the copy failed, check the output from IEBCOPY to find the error. Fix the error and rerun the job.

**GIM239s LINK {FAILED | SUCCESSFUL} - MOD = *mod* - LMOD = *lmod* - LIBRARY = *lib*  
{- SYSMOD = *sysmod* | - LOAD MODULE BUILD PROCESSING} - RETURN  
CODE = *rc***

## Explanation:

- *mod* - module name
- *lmod* - load module name
- *lib* - library name
- *sysmod* - SYSMOD-ID
- *rc* - return code.

A link-edit run completed for module *mod* into load module *lmod* in library *lib* with a return code of *rc*. The module was being link-edited, either for the indicated SYSMOD or as part of load module build processing.

Multiple SYSMODs might cause SMP/E to link-edit modules in one invocation; therefore, some SYSMODs might have modules that SMP/E link-edited successfully, even though an error code resulted.

- This message says that all of the modules and/or load modules that SMP/E handled during this invocation of the linkage editor failed, but only one module may have had an error.
- The message may also appear for modules within a SYSMOD that SMP/E never link edited because other modules in the SYSMOD were in error.

**System Action:** SYSMOD processing stops if the return code is nonzero and one of the following is true:

- The return code is greater than the return code you specified.
- The return code is greater than the default return code.

*Note:* An exception to this is if CSECT deletes the *rc* = 12 and changes it to *rc* = 0.

If the message mentioned load module build processing, message GIM239 names the affected SYSMODs.

**Programmer Response:**

1. Make sure the default utility return code, or the return code you specified in the UTILITY entry, is the one you want to use. Make sure the OPTIONS entry you are using contains the name of that UTILITY entry.
2. If the link-edit failed, check the output from the linkage editor to determine the error. Fix the error and rerun the job.

**GIM240s ASSEMBLY {FAILED | SUCCESSFUL} - MOD = *mod* - LIBRARY = *lib***  
**{- SYSMOD = *sysmod* | - LOAD MODULE BUILD PROCESSING}**  
**- RETURN CODE = *rc***

**Explanation:**

- *mod* - module name
- *lib* - library name
- *sysmod* - SYSMOD-ID
- *rc* - return code.

An assembly completed for module *mod* from library *lib* with a return code of *rc*. The module was being assembled either for the indicated SYSMOD, or as part of load module build processing.

**System Action:** If the message named a SYSMOD, SYSMOD processing stops if the return code is greater than the one you specified, or is greater than the default return code.

If the message mentioned load module build processing, and the return code is greater than the one you specified or greater than the default return code, this is just a warning message. The load module will be link-edited without the module.

**Programmer Response:** If the assembly failed during SYSMOD processing, check the assembler output to determine the cause of the error. Then correct the error and rerun the job.

If the assembly failed during load module build processing, do the assembly and relink the affected load modules outside of SMP/E. Message GIM673 identifies these load modules and their system libraries.

**GIM245s SYSMOD *sysmod* SELECTED FOR RESTORE *error condition***

**Explanation:**

- *sysmod* - SYSMOD ID
- *error condition* - (see below).

SMP/E cannot restore SYSMOD *sysmod* because of one of the following error conditions:

- DELETES SYSMODS - SYSMOD *sysmod* deleted other SYSMODs when SMP/E applied it. SMP/E cannot restore the elements from the deleted SYSMODs; therefore, SMP/E cannot restore SYSMOD *sysmod*.



## GIM2XX

- HAS BEEN ACCEPTED - SMP/E accepted SYSMOD *sysmod* into the system's distribution libraries. Therefore, SMP/E cannot use the elements on the distribution libraries to restore the target system libraries.
- IS DELETED - Another SYSMOD, which SMP/E applied, deleted SYSMOD *sysmod*.
- IS NOT APPLIED - SMP/E did not apply SYSMOD *sysmod* and, therefore, cannot restore it.
- IS SUPERSEDED - SMP/E found SYSMOD *sysmod* as a superseded-only entry on the target zone. This means that SMP/E never applied the SYSMOD; instead, the SYSMOD was superseded by one or more SYSMODs that SMP/E applied. In this situation, SMP/E cannot determine the set of SYSMODs it should restore.

**System Action:** SYSMOD *sysmod* processing stops. If SYSMOD *sysmod* is a function SYSMOD, this message will be followed by GIM370, showing that the SMP/E RESTORE command is stopping.

**Programmer Response:** Delete SYSMOD *sysmod* from the SELECT or GROUP list to correct the list of SYSMODs that SMP/E selects for RESTORE.

**GIM246s** *type name* NOT FOUND ON *lib* LIBRARY [FOR SYSMOD *sysmod* | FOR LOAD MODULE BUILD PROCESSING]

**Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of the data set
- *sysmod* - SYSMOD-ID.

An entry for the named element does not exist on the specified library and is needed for this function.

**System Action:** Examine subsequent messages in the output listing to determine system action.

However, if the entry type was ASSEM and the message named a SYSMOD:

1. SMP/E will not issue any further messages.
2. SMP/E will not do assemblies for the SYSMOD.
3. SYSMOD processing continues.

If the entry type was ASSEM and the message referred to load module build processing, message GIM673 will name the affected load modules and their system libraries.

**Programmer Response:** Use the SMPLOG to determine why SMP/E could not find *name* on the library.

If the message named a SYSMOD, *sysmod* may not apply to your system.

If the assembly failed and the message referred to load module build processing, do the assembly and relink the affected load modules outside of SMP/E.

**GIM247s BLDL FAILED IN LIBRARY *lib* FOR LOAD MODULE *lmod* IN SYSMOD *sysmod*****Explanation:**

- *lib* - ddname of data set
- *lmod* - load module name
- *sysmod* - SYSMOD-ID.

The BLDL, which SMP/E issued to get linkage editor parameters, failed for this load module.

*Note: This message is normal when SMP/E accepts elements to the DLIBs for the first time.*

**System Action:**

- For ACCEPT processing, SMP/E uses a default set of linkage editor parameters: RENT, REUS, DC, and REFR.
- For APPLY processing, SMP/E uses the parameters you specify during system generation.

**Programmer Response:** If you are applying the SYSMOD, check for an incorrect library name.

**GIM248s THE *command* FUNCTION WAS REQUESTED - NO SYSMODS MEET SPECIFICATIONS****Explanation:**

- *command* - RECEIVE, APPLY, ACCEPT, RESTORE, or REJECT.

There were no SYSMODs that met the specifications on the command.

**System Action:** Command processing stops. Processing continues with the next command.

**Programmer Response:** Do the following:

1. Review the other messages that SMP/E issued during this function.
2. Check that the operands specified on the command are correct.

**GIM249s SYSMOD *sysmod* FAILED BECAUSE OF NAME CARD CONFLICT IN MODULE *mod*****Explanation:**

- *sysmod* - SYSMOD-ID
- *mod* - module name.

## GIM2XX

The same GIMSPZAP command contains name cards of different types (NAME CSECT and NAME LMOD CSECT).

**System Action:** SYSMOD processing stops. Processing continues with the next SYSMOD.

**Programmer Response:** Correct the NAME cards and rerun the job.

### GIM252s INCOMPLETE GIMSMP CONTROL STATEMENT

**Explanation:** SMP/E found an incomplete command, and an end-of-file occurred before the end of the statement. SMP/E lists the command in error before this message.

**System Action:** SMP/E does not perform the function. For SMP/E's action, check the messages that follow in the output listing.

**Programmer Response:** Check for the following:

- A missing comment terminator (\*).
- A missing statement terminator (.
- A previous LOG command that has missing parentheses.
- The statement for the minimum data requirement.

Correct the statement and rerun for UCLIN processing.

### GIM253s *type* ENTRY *name* TO BE DELETED DOES NOT EXIST

**Explanation:**

- *type* - entry type
- *name* - entry name.

SMP/E could not find the entry you want to delete in the specified zone.

**System Action:** SMP/E ignores the UCL statement. Processing continues with the next UCL statement.

**Programmer Response:** Correct the UCL statement and rerun the statement.

### GIM255s UPDATE COMPLETE FOR *name*

**Explanation:**

- *name* - entry name.

UCLIN processing for the entry completed.

**System Action:** Update processing continues with the next UCL statement.

**Programmer Response:** None.

**GIM256s UPDATE PROCESSING TERMINATED - UPDATE NOT COMPLETE**

**Explanation:** UCLIN processing for the entry did not finish because of an error SMP/E identified in a previous message. SMP/E did not change the entry.

**System Action:** Update processing stops.

**Programmer Response:** Fix the error and rerun the statement.

**GIM257s SPECIFIED UPDATE RESULTS IN INSUFFICIENT DATA - *data* REQUIRED**

**Explanation:**

- *data* - UCL data.

If SMP/E made the indicated update, the affected entry would have insufficient or inconsistent data. *data* indicates the additional data that SMP/E needs.

**System Action:** SMP/E ignores the UCL statement and does not change the entry. Processing continues with the next UCL statement.

**Programmer Response:** Check the *SMP/E Reference* to find out which subentries are required for the entry you are updating. Provide the missing information in the SMPCSI or UCLIN input and rerun the statement.

**GIM258s END OF FILE ON {UCL | ZONEEDIT } INPUT STREAM - PROCESSING TERMINATED**

**Explanation:** Before SMP/E found the ENDUCL or ENDZONEEDIT statement, an end-of-file occurred on the SMPCNTL data set.

**System Action:** SMP/E ignores the current UCL or ZONEEDIT statement, if any, and stops command processing.

**Programmer Response:** Add the ENDUCL or ENDZONEEDIT statement to the input data stream and rerun the statements that SMP/E did not process.

**GIM259s *type name* PEMAX EXCEEDED ATTEMPTING TO ADD *element name element type***

**Explanation:**

- *type* - entry type
- *name* - entry name
- *element name* - element name
- *element type* - element type.

As SMP/E tried to add a subentry to the specified entry, the number of elements in the entry exceeded the maximum allowed number (PEMAX or fixed value.)

## GIM2XX

### System Action:

- For UCL processing, SMP/E ignores the UCL statement, and processing continues with the next UCL statement.
- For RECEIVE, APPLY, ACCEPT and RESTORE processing, the SYSMOD associated with the specified entry stops processing.

You may see this happen for the GLOBAL zone during RECEIVE processing. In this case, SMP/E does not add the named FMID entry to the GLOBAL zone. However, SMP/E successfully receives the function SYSMOD.

**Programmer Response:** Ensure that the OPTIONS entry containing the increased PEMAX value is the OPTIONS entry named in the zone definition entry. Then do one of the following:

- For UCL processing, do either of the following:
  - Reduce the number of subentries you specified in the UCL statement.
  - Increase the PEMAX value in the OPTIONS entry being used.
- For RECEIVE, APPLY, ACCEPT and RESTORE processing, increase the PEMAX value in the OPTIONS entry being used.

If this situation occurs for the GLOBAL zone, do the following:

1. Increase the PEMAX value in the OPTIONS entry being used.
2. Run UCLIN against the GLOBALZONE entry to add FMIDs that SMP/E received but did not add to the GLOBALZONE entry.

### GIM261s *keyword* ENTERED IS NOT EQUAL TO *keyword value* IN ENTRY

#### Explanation:

- *keyword* - UCL keyword
- *value* - keyword value in the existing entry.

The keyword on the UCL DEL command did not match the value of the existing entry.

**System Action:** UCL processing stops for the UCL statement.

**Programmer Response:** Resubmit the UCL statement with the correct value.

### GIM262s ERROR FORCES JCLIN SCAN TO TERMINATE

**Explanation:** An error, which SMP/E explained in a previous message, stopped the JCLIN scan of the Stage I system generation file.

**System Action:** JCLIN processing stops.

**Programmer Response:** Fix the previous error and rerun the job.

**GIM263s ERROR OCCURRED IN STEP *stepname* OF JOB *jobname*****Explanation:**

- *stepname* - step name
- *jobname* - job name.

This message:

1. Is preceded by the command in error and its description
2. Tells the job and step where the error occurred
3. Is followed by an error description.

**Notes:**

1. If "NONAME" appears as *stepname* or *jobname*, then the error exists in *stepname* or *jobname*.
2. If the error is on the EXEC statement of the failing step, this message may show the step name or job name that precedes the one in error.

**System Action:** None.

**Programmer Response:** Your response depends on the type of error:

- ERR LOCATING MOD KEYWORD

*Description:* SMP/E did not find the MOD= keyword on the EXEC statement during assembler processing.

*Response:* Fix the error and rerun the job.

- ERROR ON MOD NAME STOW

*Description:* An undefined error occurred during linkage-editor processing when SMP/E was updating the module.

*Response:* Make corrections and rerun the job.

- INVALID IEBCOPY STATEMENT

*Description:* The statement is syntactically invalid.

*Response:* Consult the *Utilities* manual for the correct IEBCOPY format. Fix the error and rerun the job.

- INVALID KEYWORD FOR MOD

*Description:* Linkage editor processing found invalid characters in the MOD= keyword.

*Response:* Consult the *Linkage Editor and Loader* manual for the correct format. Fix the the error and rerun the job.

## GIM2XX

- INVALID MACNAME SPECIFIED

*Description:* During assembler processing, SMP/E found a macro name with an incorrect length. (The length must be from 1 to 8 characters.)

*Response:* Fix the error and rerun the job.

- LAST LINE PROCESSED

*Description:* An I/O error occurred in the last line that SMP/E processed.

*Response:* Fix the error and resubmit the job, if necessary.

- LAST LKED CNTL STMT

*Description:* SMP/E found a syntax error in a linkage-editor statement during linkage editor processing.

*Response:* Consult the *Linkage Editor and Loader* manual for the correct format. Correct the error and rerun the job.

- MACRO TABLE EXCEEDED

*Description:* Assembler processing exceeded the space SMP/E allocated for macro tables.

*Response:* Allocate more storage and rerun the job.

- NO DSNAME CODED

*Description:* You did not code DSNAME on the EXEC statement for the linkage-editor procedure.

*Response:* Fix the error and rerun the job.

- NO MOD KEYWORD FOUND

*Description:* SMP/E did not find the NAME link edit statement or the MOD = keyword on the EXEC statement during linkage-editor processing.

*Response:* Fix the error and rerun the job.

- NO MODNAME FOUND IN STMT

*Description:* The module name is not on the SYSLMOD DD statement or on a NAME link-edit statement.

*Response:* Fix the error and rerun the job.

- NO SYSLMOD CARD FOUND

*Description:* SMP/E did not find a SYSLMOD DD statement when you specified PGM = on the EXEC statement.

*Response:* Fix the error and rerun the job.

- TABLE OVERFLOW

*Description:* During linkage editor processing, this was the last line SMP/E processed before it used up the work area.

*Response:* Allocate more main storage and rerun the job.

**GIM266s ERROR OCCURRED IN {LINKAGE EDITOR | IEBCOPY | ASSEMBLER} INPUT**

**Explanation:** This message shows the type of system generation step that JCLIN processing was scanning when an error occurred. A prior message indicates the error.

**System Action:** JCLIN processing stops.

**Programmer Response:** Fix the error and rerun the job.

**GIM267s DIRECTORY SPACE EXCEEDED ATTEMPTING TO STORE *type name* ON *lib* LIBRARY**

**Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of the data set.

When SMP/E was attempting to store *name*, SMP/E exceeded the number of directory blocks allocated to the library.

**System Action:** SMP/E does not store the member. Messages that follow this message show SMP/E's action.

**Programmer Response:** Increase the directory block allocation for the indicated library and rerun the job.

**GIM268s I/O ERROR OCCURRED ATTEMPTING TO STORE *type name* ON *lib* LIBRARY**

**Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of data set.

SMP/E found an I/O error when it was storing *name* in *lib*.

**System Action:** SMP/E does not store the entry. SMP/E processing ends, as shown by the messages that follow this message.

**Programmer Response:** Fix the error and rerun the job.



## GIM2XX

### GIM269s I/O ERROR OCCURRED ATTEMPTING TO BLDL FOR *type name* ON *lib* LIBRARY

**Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of data set.

A BLDL operation caused an I/O error on the library, *lib*.

**System Action:** The messages that follow this message show SMP/E's action.

**Programmer Response:** Fix the error and rerun the job.

### GIM273s INPUT TEXT NOT FOUND

**Explanation:** One of the following items caused this message:

- SMP/E could not find either inline text or object records that were to follow an SMP/E element modification control statement.
- SMP/E could not find any input in the JCLIN input file.

**System Action:** SMP/E does not receive the SYSMOD it is processing.

**Programmer Response:** Do one of the following:

- Insert an object deck or text deck after the element modification control statements for the elements not on a TXLIB, LKLIB, or RELFILE data set, or
- Insert JCL in the JCLIN input data set.

### GIM274s I/O ERROR - *jobname, stepname, address, device, lib, action, description* { *access\_method* } | { *rbn, access\_method* } | { *track\_address, block\_number, access\_method* }

**Explanation:** SMP/E found an I/O error when it was processing data set *lib*. Information that the message gives corresponds to the SYNADAF information listed in the *Data Management Services Guide* or *Data Administration Guide*.

**System Action:** The messages that follow this message show SMP/E's action.

**Programmer Response:** Fix the error and rerun the job.

**GIM276s ILLEGAL UPDATE REQUEST FOR *lib*****Explanation:**

- *lib* - ddname of data set.

SMP/E does not allow the requested combination of UCLIN operations.

**System Action:** SMP/E does not update the member in the data set.

**Programmer Response:** Fix the statement in error and resubmit the statement.

**GIM277s *type name* TO BE REPLACED DOES NOT EXIST - ADD ASSUMED****Explanation:**

- *type* - entry type
- *name* - entry name.

SMP/E could not find the entry specified on the UCL REP command.

**System Action:** UCLIN assumes that you asked for an ADD operation.

**Programmer Response:** Do one of the following:

- If "ADD" was the correct assumption, you do not need to take any action.
- If "ADD" was not the correct assumption, do the following:
  - Check whether you used the correct entry name.
  - Check whether you specified SET BDY before processing.
  - Check the ZONEINDEX for the correct CSI data set.

Fix the problem and rerun the statement.

**GIM281s DUPLICATE ELEMENT *name* IN SYSMOD *sysmod*****Explanation:**

- *name* - element name as specified in the SYSMOD
- *sysmod* - SYSMOD-ID.

During RECEIVE processing, SMP/E found two modification control statements that specify the same element in one SYSMOD. SMP/E prints the modification control statement that specifies the duplicate element before this message.

**System Action:** SYSMOD processing ends.

**Programmer Response:** Fix the modification control statements so duplicate elements do not exist and rerun the job.

## GIM2XX

**GIM283s** *lib* DDCARD MISSING [FOR LOAD MODULE *mod* | FOR COMPRESS | FOR MODULE *mod* IN SYSMOD *sysmod* ]

**Explanation:**

- *lib* - ddname of data set
- *mod* - module name
- *sysmod* - SYSMOD-ID.

The DD statement that the function needs does not exist.

**System Action:** The messages that follow this message in the output listing show SMP/E's action.

**Programmer Response:** Do one of the following:

- Add the required DD statement.
- Correct the ddname.

**GIM284s** SYSMOD *sysmod* HAD A LOAD MODULE SPECIFICATION ERROR IN MODULE *mod*

**Explanation:**

- *sysmod* - SYSMOD-ID
- *mod* - module name.

The load module name you specified on an IMASPZAP NAME statement is not in the target zone as a valid load module for the specified module.

**System Action:** SYSMOD processing ends. Processing continues with the next SYSMOD.

**Programmer Response:** Fix or remove the load module name from the IMASPZAP NAME statement and rerun the job.

*Note: If SMP/E used a valid load module name in an ALIAS operand, use the name that is in the target zone and not in the ALIAS operand.*

**GIM285s** *lib* REFERENCES AN UNMOVABLE DATA SET

**Explanation:**

- *lib* - ddname of data set.

SMP/E cannot compress the library that you requested because it has location-dependent data.

**System Action:** SMP/E bypasses compress processing for the data set. Processing continues with the next data set.

**Programmer Response:** None.

**GIM287s CONTROL STATEMENT NOT PROCESSED- A USER SPECIFIED RETURN CODE HAS BEEN EXCEEDED**

**Explanation:** You used the RC operand, which determines the processing of the current command, to specify a return code for another SMP/E command. SMP/E found that the return code for the command is greater than the limit you chose.

**System Action:** SMP/E did not process the current control statement. Processing continues.

**Programmer Response:** Look at the SMPOUT data to find what command caused SMP/E to bypass the current control statement. Fix any errors and reevaluate your course of action.

**GIM288s NO GIMSMP UPDATE FUNCTIONS HAVE BEEN PERFORMED**

**Explanation:** An error shown by a previous message caused JCLIN processing to end without updating the target zone.

**System Action:** Processing continues.

**Programmer Response:** Fix the error and rerun JCLIN processing.

**GIM292s INVALID MEMBER NAME ON IEBUPDTE CONTROL STATEMENT**

**Explanation:** The member name on the IEBUPDTE control statement did not match the name on the ++UPDTE, ++MACUPD, or ++SRCUPD modification control statement.

**System Action:** RECEIVE processing stops for the SYSMOD.

**Programmer Response:** Correct the IEBUPDTE control statement and rerun the job.

**GIM302s *command* PROCESSING TERMINATED FOR SYSMOD *sysmod* - REASON=*reason***

**Explanation:**

- *command* - APPLY, ACCEPT, or RESTORE
- *sysmod* - SYSMOD-ID
- *reason* - reason for termination.

## GIM3XX

During *command* processing, SMP/E stopped processing SYSMOD *sysmod* for one of the following reasons:

- ALL DELETING SYSMODS FAILED

The SYSMODs that would have deleted *sysmod* failed.

- ALL SUPERSEDING SYSMODS FAILED

Processing stopped for SYSMOD *sysmod* because all the processed SYSMODs that superseded *sysmod* failed.

- DELETE PROCESSING FAILURE

While processing a SYSMOD that contained the DELETE operand, SMP/E found an error. SMP/E stops processing the SYSMOD.

- EXCLUDED SYSMODS

Processing stopped for SYSMOD *sysmod* because SMP/E excluded a SYSMOD (EXCLUDE keyword) that would have resolved *sysmod*'s requisites or HOLD reason-id.

- FMID SYSMOD FAILURE

Processing stopped for the SYSMOD *sysmod* because the SYSMOD that *sysmod* specified as the FMID value failed.

- HOLD REASON IDS NOT RESOLVED

Processing stops for SYSMOD *sysmod* because SMP/E did not resolve one or more HOLD conditions. Message GIM359, which follows this message, names the reason-id and the type of HOLD.

- IFREQ SYSMOD FAILURE

Processing stopped for the SYSMOD *sysmod* because one of its IFREQ SYSMODs failed.

- INLINE JCLIN FAILURE

Inline JCLIN processing for a SYSMOD failed. See previous messages in the output listing for reasons for failure.

- MISSING/NOGO REQUISITES

Processing stopped for SYSMOD *sysmod* because one or more requisites (PRE, REQ, IFREQ, or FMID) were missing or failed during APPLY or ACCEPT processing. Message GIM359, which follows this message, names the missing or NOGO requisite SYSMODs. Because SMP/E considers the reason-ids for "held" SYSMODs as requisites, SMP/E reports in GIM359 the "held" SYSMODs whose reason-ids are unresolved.

- PRE SYSMOD FAILURE

Processing stopped for the SYSMOD *sysmod* because one of its prerequisite SYSMODs failed.

- RELATED SYSMOD FAILURE

SYSMOD *sysmod* had a version of a module, macro, or source module that SMP/E did not select for processing because another SYSMOD was processed that had a higher level of the element. SMP/E stopped processing the SYSMOD with the higher level of the element. The SMP/E reports and messages will help you determine which element caused the problem.

- REQ SYSMOD FAILURE

Processing stopped for the SYSMOD *sysmod* because one of its requisite SYSMODs failed.

- RESTORE LIST INCOMPLETE

Processing stopped for SYSMOD *sysmod*, which supersedes the SYSMOD that SMP/E names in message GIM359. Include the superseded SYSMODs in the RESTORE list if you previously APPLIED the indicated SYSMOD.

- SMP UPDATE FAILURE

Processing completed for all elements of SYSMOD *sysmod*, but SMP/E found an error when it was updating a data set with the SUP and MODID data. The message before this one shows the cause of the error.

- SYSTEM ABEND

SMP/E intercepted a system ABEND, shown by message GIM432. SMP/E stopped all SYSMOD processing.

- SYSTEM UTILITY FAILURE

SMP/E invoked one of the system utilities (assembler, update, copy, zap, or linkage editor) for one of the elements of SYSMOD *sysmod*. The utility returned a code that either you or SMP/E defined as an error code. Thus, SMP/E stops processing for the SYSMOD related to that element.

**System Action:** SYSMOD processing stops. SMP/E does not attempt to process any other element of the SYSMOD.

**Programmer Response:** Do one of the following:

- Examine SMP/OUT for messages relating to elements of the SYSMOD.
- Use the APPLY/RESTORE/ACCEPT SUMMARY reports to find the associated SYSMOD that caused the failure.

Fix the error and rerun the job.

# GIM3XX

**GIM303s COMPRESS {FAILED|SUCCESSFUL} - LIBRARY = *lib* -RETURN CODE = *rc***

**Explanation:**

- *lib* - ddname of the data set
- *rc* - return code.

SMP/E ran IEBCOPY to compress library *lib* and received return code *rc* from IEBCOPY.

**System Action:** SMP/E stops processing the current function if the return code is nonzero and greater than one of the following:

- The return code you specified
- The default return code.

**Programmer Response:** Check the IEBCOPY output to find the error. Fix the error and rerun the job.

**GIM319s SYSMOD *sysmod1* DOES NOT PRE OR SUP {SMPCSI | SELECTED} ELEMENT *modid sysmod2***

**Explanation:**

- *sysmod1* - SYSMOD-ID
- *modid* - RMID or UMID
- *sysmod2* - SYSMOD-ID that is not a prerequisite or that *sysmod1* does not supersede.

This message describes the ID check that GIM382 reports. SMP/E issues this message for every element in a service SYSMOD (or in a function SYSMOD you are reinstalling) that does not name the RMID and UMIDs of the previously processed version of the named element in the PRE or SUP operands. When this happens, SMP/E cannot determine the relationship between the element in SYSMOD *sysmod1* and the previously processed version of the element. You may also get this message if you are trying to reinstall a SYSMOD after you have already installed additional service for it.

If *source* is:

- **SELECTED:** SYSMOD *sysmod1* supplied a version of the element that it selected from another SYSMOD. SMP/E was processing this other SYSMOD during the current APPLY or ACCEPT processing.
- **SMPCSI:** SYSMOD *sysmod1* supplied a version of the element that it selected from another SYSMOD. SMP/E processed this other SYSMOD during a previous APPLY or ACCEPT.

If *modid* is:

- **RMID:** *sysmod2* is the SYSMOD-ID of the last SYSMOD that supplied a total replacement ( ++MOD, ++MAC, ++SRC) for the named element.
- **UMID:** *sysmod2* is the SYSMOD-ID of a SYSMOD that supplied an update ( ++ZAP, ++MACUPD, ++UPDTE, or ++SRCUPD) for the named element.

**System Action:** This message is informational. Look at the preceding GIM382 message to determine system action.

**Programmer Response:** Determine the relationship between the elements in the SYSMODs involved. If the relationship is invalid, do one of the following:

- LIST *sysmod1* to see whether it was already installed.
- Reject and modify SYSMOD *sysmod1* to change the PRE and SUP operands you previously specified.
- Use UCLIN to modify the RMID and UMID attributes of the elements on the target zone or distribution zone.

In addition, SMP/E may need to apply other SYSMODs before it processes this one to establish the correct relationship.

**GIM324s** *type name* SUBENTRY IN SYSMOD *sysmod2* REGRESSED BY SYSMOD *sysmod1*

**Explanation:**

- *type* - subentry type
- *name* - element name
- *sysmod2* - SYSMOD-ID of the MODID from the element entry
- *sysmod1* - SYSMOD-ID of the SYSMOD that SMP/E is processing.

One of the following situations occurred:

- SYSMOD *sysmod1* did not specify SYSMOD *sysmod2* as a prerequisite.
- SYSMOD *sysmod1* did not supersede SYSMOD *sysmod2*.
- SMP/E did one of the following:
  - Applied SYSMOD *sysmod2*
  - Accepted SYSMOD *sysmod2*
  - Concurrently processed SYSMOD *sysmod2* with a user modification to the element.

Therefore, SMP/E considers SYSMOD *sysmod2* to be regressed by SYSMOD *sysmod1*.

**System Action:** The message shows the warning severity. Processing of SYSMOD *sysmod1* continues.

**Programmer Response:** None.



## GIM3XX

**GIM325s** SYSMOD *sysmod1* WHICH SUPERSEDES SYSMOD *sysmod2* DOES NOT CONTAIN ELEMENT *name*

**Explanation:**

- *sysmod1* - superseding SYSMOD-ID
- *sysmod2* - superseded SYSMOD-ID
- *name* - element name.

The superseding SYSMOD does not have all the elements that the superseded SYSMOD had.

**System Action:** Processing continues, return code 4.

**Programmer Response:** Review the SYSMODs and make necessary corrections to the elements shown in the message.

**GIM327s** INPUT TEXT FOUND AND LKLIB, TXLIB, RELFILE OR DELETE KEYWORD SPECIFIED ON CONTROL STATEMENT

**Explanation:** Input text appeared where it should not have. One of the following occurred:

- Following an element modification control statement, SMP/E found modification text indicating that the text is on one of the following:
  - LKLIB
  - TXLIB
  - Relative file.
- SMP/E found modification text following an element modification control statement that specified the DELETE keyword.

**System Action:** Processing stops for this SYSMOD but continues for any remaining SYSMODs.

**Programmer Response:** The SYSMOD is built incorrectly. Review the SYSMOD for any of the following:

- A missing element modification control statement
- An incorrectly coded “+ +” character
- A conflict between where SMP/E placed the modification text and the specification of the LKLIB, TXLIB, RELFILE, or DELETE keywords.

**GIM338s** *mcs* MODIFICATION CONTROL STATEMENT NOT VALID IN FUNCTION SYSMOD**Explanation:**

- *mcs* - ++UPDTE, ++MACUPD, ++SRCUPD, or ++ZAP.

A FUNCTION SYSMOD cannot contain the specified modification control statement.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Use only allowable types of modification control statements to construct the function SYSMOD.

**GIM339s** RMID KEYWORD ONLY VALID IN FUNCTION SYSMOD

**Explanation:** Only a FUNCTION SYSMOD may contain the RMID keyword, which is optional on the ++MAC, ++MOD and ++SRC modification control statements.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Specify the RMID operand only on FUNCTION SYSMODs.

**GIM340s** RELFILE KEYWORD INVALID WITHOUT FILES KEYWORD ON HEADER MODIFICATION CONTROL STATEMENT

**Explanation:** You did not specify the FILES operand on the header modification control statement when you specified the RELFILE operand. (The RELFILE operand is optional on the ++JCLIN, ++MAC, ++MOD, or ++SRC modification control statements.)

**System Action:** SYSMOD processing stops.

**Programmer Response:** Specify the FILES operand on the header modification control statement and receive this SYSMOD again.

**GIM341s** KEYWORDS *keyword1* AND *keyword2* ARE MUTUALLY EXCLUSIVE**Explanation:**

- *keyword1* - first keyword
- *keyword2* - second keyword.

You cannot specify keywords *keyword1* and *keyword2* in the same command. The previous message lists the command in error.

**System Action:** One of the following occurs:

- If the error occurred on a modification control statement, SYSMOD processing stops.
- If the error occurred on an SMP/E command, command processing stops.

**Programmer Response:** Remove either keyword and run the job again.

## GIM3XX

### GIM342s ONLY ONE { + +JCLIN | *keyword* } ALLOWED IN A {SYSMOD | CONTROL STATEMENT}

**Explanation:**

- *keyword* - keyword.

The SYSMOD is built incorrectly for one of these reasons:

- It contains two + +JCLIN modification control statements.
- A modification control statement in the SYSMOD contains the *keyword* keyword more than once.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Do one of the following:

- Remove one of the + +JCLIN modification control statements.
- Remove any duplicate keywords.

Rerun the job.

### GIM343s SMPPTFIN WITH RELFILES MUST BE STANDARD LABEL TAPE

**Explanation:** SMP/E could not RECEIVE a SYSMOD built with relative files because the SMPPTFIN data set that contains the SYSMOD is not a standard labeled tape.

**System Action:** SYSMOD processing stops.

**Programmer Response:** You may have mounted the wrong tape. Rerun the job using the correct standard labeled tape.

### GIM345s INVALID MODIFICATION CONTROL STATEMENT

**Explanation:** SMP/E found an invalid modification control statement during RECEIVE processing.

**System Action:** SYSMOD processing stops.

SMP/E checks the subsequent statements in the SMPPTFIN input stream until it finds the next header modification control statement (+ +PTF, + +FUNCTION, + +USERMOD, or + +APAR). RECEIVE processing then continues normally.

**Programmer Response:** Check the following:

1. The modification control statement tagged by the message:
  - Is its syntax correct?
  - Is it properly positioned within a set of modification control statements?
2. The input that SMP/E is processing:
  - Is the input in UPPER CASE letters?
3. The modification type:
  - Is it is on the same card as the ++?

Rebuild the SYSMOD and run the job again.

#### **GIM346s INVALID IEBUPDTE CONTROL STATEMENT**

**Explanation:** SMP/E found an IEBUPDTE control statement other than “./ CHANGE” or “./ ENDUP” following an ++UPDTE, ++MACUPD or ++SRCUPD modification control statement.

**System Action:** Processing stops for the SYSMOD that has the invalid statement.

**Programmer Response:** Fix the SYSMOD and run RECEIVE again.

#### **GIM347s INVALID RECORD. MODIFICATION CONTROL STATEMENT EXPECTED.**

**Explanation:** SMP/E expected a modification control statement, but instead found that the SMPPTFIN data set input stream has a non-SMP/E statement. This happens when input text follows a modification control statement that has a syntax error.

**System Action:** SMP/E ignores all following non-SMP/E statements. Processing stops for the SYSMOD that has the invalid statement.

**Programmer Response:** Fix the problem and run RECEIVE again.

#### **GIM348s SYSMOD CONTAINS MORE THAN ONE ++VER MODIFICATION CONTROL STATEMENT FOR THE SAME SREL-FMID PAIR**

**Explanation:** The SYSMOD that SMP/E was receiving had more than one ++VER modification control statement naming the same SREL and FMID. (The ++VER modification control statement that caused the error immediately precedes this message.) A SYSMOD like this creates an ambiguous situation when you want to APPLY/ACCEPT.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Fix the problem and run RECEIVE again.

## GIM3XX

### GIM349s ++IF MODIFICATION CONTROL STATEMENT NOT ASSOCIATED WITH ANY PRECEDING ++VER MODIFICATION CONTROL STATEMENT

**Explanation:** The ++IF modification control statements must follow a ++VER modification control statement. This allows SMP/E to associate the ++IF statements with the ++VER statements chosen at APPLY or ACCEPT time.

In this case, SMP/E found an ++IF modification control statement that did not follow a ++VER modification control statement. (The ++IF modification control statement that caused the error immediately precedes this message).

**System Action:** SYSMOD processing stops.

**Programmer Response:** Fix the problem and run RECEIVE again.

### GIM350s RELFILE GREATER THAN NUMBER OF FILES IN THE SYSMOD

**Explanation:** The RELFILE value specified on a ++JCLIN, ++MOD, ++MAC, or ++SRC modification control statement is greater than the number of files specified in the FILES keyword on the header modification control statement. (The element modification control statement that caused the error immediately precedes this message.)

**System Action:** RECEIVE processing stops. SMP/E generates the RECEIVE Summary Report.

**Programmer Response:** Fix the problem and run RECEIVE again.

### GIM351s *sysmod* TERMINATED WHILE LOADING RELFILES. COPY RETURN CODE = *rc*

**Explanation:**

- *sysmod* - SYSMOD-ID
- *rc* - return code from copy processing.

While SMP/E was loading the relative files, SYSMOD *sysmod* stopped.

**System Action:** RECEIVE processing stops for SYSMOD *sysmod*.

**Programmer Response:** Look at the copy SYSPRINT output to determine the problem.

### GIM352s ALLOCATE SUCCESSFUL FOR *lib* ON VOLUME *volser* [- EXISTING DATASET FOUND]

**Explanation:**

- *lib* - data set name
- *volser* - volume serial number.

During RECEIVE processing, SMP/E successfully allocated data set *lib* on volume *volser*. SMP/E allocated the data set when it was loading a relative file for a SYSMOD.

If the message also says “EXISTING DATASET FOUND”, SMP/E found a preallocated data set on the specified volume and will try to load the relative files in the existing data set.

**System Action:** SMP/E uses the allocated data set to load the relative files.

**Programmer Response:** None.

**GIM353s ALLOCATE FAILED FOR *lib* [ON VOLUME *volser*] - *reason***

**Explanation:**

- *lib* - data set name
- *volser* - volume serial number
- *reason* - reason for the error.

During RECEIVE processing, SMP/E tried to allocate data set *lib* on volume *volser*, in order to load one of the relative files for a SYSMOD. However, SMP/E found an error when it was allocating the data set. The error, *reason*, is one of the following:

- CURRENTLY DOCUMENTED REASONS

Unchanged.

- DATASET FOUND IS NOT A PDS

SMP/E found an existing data set on the specified volume; however, the data set was not a partitioned data set, and SMP/E could not use it to load the relative files.

- DATASET NOT FOUND

During APPLY or ACCEPT processing, SMP/E tried to find one of the data sets that it allocated and loaded from a relative file during RECEIVE processing. SMP/E could not find the data set.

- ERROR CODE = x“nn”

Error code x“nn” resulted from DADSM. See the *DADSM Logic* manual for your operating system for an explanation of the error codes.

- NO VOLUMES SPECIFIED

The SMPTLIB DD statement did not specify a list of volumes for SMP/E to search, in order to process or allocate a RELFILE data set.

- SMPTLIB HAS DUMMY STATUS

The SMPTLIB DD statement specified either the DUMMY keyword or DSN=NULLFILE.

## GIM3XX

**System Action:** Processing stops for the SYSMOD associated with that data set.

**Programmer Response:** Find the cause of the error by looking at the following:

1. The DADSM return code.
2. The volumes SMPTLIB DD statement specifies.

If the error occurred during APPLY or ACCEPT processing, check that the same volumes are available that SMP/E loaded at RECEIVE time.

### GIM354s SCRATCH SUCCESSFUL FOR *lib* ON VOLUME *volser*

**Explanation:**

- *lib* - data set name
- *volser* - volume serial number.

SMP/E scratched data set *data* from volume *volser*. This data set is one of the data sets that SMP/E allocated to process relative files.

**System Action:** SYSMOD processing continues normally.

**Programmer Response:** None.

### GIM355s ERROR PROCESSING *type* ENTRY FOR SYSMOD *sysmod* ON THE *lib*

**Explanation:**

- *type* - entry type
- *sysmod* - SYSMOD ID
- *data* - SMP/E data set.

SMP/E found an error while processing SYSMOD *sysmod*.

- If the entry type is MCS, SMP/E found an error when it was trying to parse the MCS entry on the SMPPTS data set. The error could be one of the following:
  - An I/O error
  - A mismatch between the MCS and SYSMOD entries on the SMPPTS.
- If the entry type is not MCS, look at the preceding SMP/E output to find the cause of the error for the named entry type.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Do one of the following:

- If the entry type is MCS, reject the SYSMOD and receive again.
- If any other case, follow the recommended action in the preceding error message(s).

**GIM356s** *type name* TO BE ADDED TO ENTRY ALREADY EXISTS [AS *value*]**Explanation:**

- *type* - subentry type
- *name* - subentry name
- *value* - existing value of the subentry.

A UCL statement tried to add subentry *type name* to the entry, but the subentry was already there. Either the entry or the subentry is incorrect.

**System Action:** SMP/E does not process any of the changes for this UCL statement.

**Programmer Response:** Fix the UCL statement and rerun the job.

**GIM357s** *type name* TO BE DELETED FROM ENTRY DOES NOT EXIST**Explanation:**

- *type* - subentry type
- *name* - subentry name.

A UCL statement tried to delete subentry *type name*, but the subentry was not there.

**System Action:** SMP/E does not process any of the changes in this UCL statement.

**Programmer Response:** Rerun the UCL statement without the specified subentry.

**GIM358s** *type name* TO BE REPLACED IN ENTRY DOES NOT EXIST - ADD ASSUMED**Explanation:**

- *type* - subentry type
- *name* - subentry name.

A UCL statement tried to replace subentry *name*, but it does not exist. SMP/E assumes that you meant to add *type*, and processing continues.

**System Action:** SMP/E processes the UCL statement as if ADD were specified.

**Programmer Response:** Do one of the following:

- If "ADD" is the right assumption, no response is required.
- If SMP/E should not have added the subentry, do the following:
  1. Use the UCL DEL function to delete the subentry.
  2. Specify the correct entry and subentry.
  3. Rerun the UCL REP request.



# GIM3XX

**GIM359s** *sysmod1* requisite [BYPASSED] [CAUSER = *sysmod2*] [BECAUSE CLASS *class* BYPASSED]

## Explanation:

- *sysmod1* - first SYSMOD-ID
- *requisite* - requisite condition (see below)
- *sysmod2* - second SYSMOD-ID
- *class* - the HOLD class being bypassed.

This message follows message GIM302 or GIM420 (referred to as GIMxxx below) and lists the requisite conditions that the SYSMOD named in GIM302 or GIM420 did not meet.

*Note: If the message text specifies PRE for a RESTORE, SMP/E lists only those SYSMODs PRE'd by the failing SYSMODs. SMP/E does not list all PREs in the chain.*

SMP/E may find the following requisite conditions:

- PRE - *sysmod1* is a prerequisite for the SYSMOD in GIMxxx.
- REQ - *sysmod1* is a requisite for the SYSMOD in GIMxxx.
- IFREQ - *sysmod1* is a conditional requisite for the SYSMOD in GIMxxx. If the causer is not the SYSMOD in GIMxxx, SMP/E will list as CAUSER the SYSMOD that led SMP/E to generate this requisite.

For PRE, REQ, and IFREQs that SMP/E does not resolve at APPLY or ACCEPT, this message shows the status of the requisite as NOGO, MISSING, or EXCLUDED.

For example:

```
GIM3590 --- UZ11111 PRE MISSING
```

The following may appear for *requisite*:

- EXCLUDED - SMP/E found the named requisite in the GLOBAL zone, but it was excluded by the EXCLUDE keyword.

Since superseding SYSMODs may resolve requisites, this status may appear if a SYSMOD that superseded the requisite was excluded.

- FMID - *sysmod1* is the FMID of the SYSMOD in GIMxxx. This may occur at APPLY or ACCEPT, since you must APPLY or ACCEPT *sysmod1* along with (or before) the SYSMOD in GIMxxx.
- FMIDED - *sysmod1* is the FMID of the SYSMOD in GIMxxx. This may occur at RESTORE when SMP/E restores *sysmod1* along with the SYSMOD in GIMxxx.

- HOLD - *sysmod1* is a HOLD reason-id that SMP/E did not resolve.

SMP/E also displays the type of hold. For example:

```
GIM3590 --- AZ00000 HOLD ERROR
```

shows that SMP/E could not process the SYSMOD because SMP/E was holding the SYSMOD for an ERROR identified by APAR number AZ00000. SMP/E automatically resolves ERROR-type reason-ids when it processes a SYSMOD that supersedes the indicated APAR. Further, BYPASS(HOLDERR(AZ00000)) on the APPLY or ACCEPT command will let SMP/E process the SYSMOD, even though it does not supersede the APAR reason-id.

USER and SYSTEM are the two other types of holds that may be in this message. You can resolve USER and SYSTEM hold reason-ids by using BYPASS(HOLDUSER(...)) and BYPASS(HOLDSYS(...)) on the APPLY or ACCEPT commands.

- MISSING - SMP/E did not find the requisite in the GLOBAL zone. (If you specifically selected a set of SYSMODs using the SELECT keyword, the SYSMOD may be on the GLOBAL zone but SMP/E did not SELECT it).
- NOGO - SMP/E found the requisite in the GLOBAL zone that was eligible for processing, but SYSMOD processing stopped. Look at previous messages to find out why the requisite failed.

Since superseding SYSMODs may resolve requisites, this failure could be from the superseding SYSMOD failing rather than the requisite itself.

- PREING - *sysmod1* is a SYSMOD that has the SYSMOD in GIMxxx as a prerequisite.
- SUPED - *sysmod1* is a superseded SYSMOD that is SUPED by the SYSMOD in message GIM302.
- SUPING - *sysmod1* is a SYSMOD that supersedes the SYSMOD in GIMxxx.
- *sysmod2* - is the SYSMOD that supplied the ++IF statement, which resulted in the missing requisite condition this message describes.

When BYPASSED appears in this message, the message follows GIM420 and shows the requisite conditions that SMP/E bypassed to APPLY, ACCEPT, or RESTORE the SYSMOD in GIM420.

*Note: If BECAUSE CLASS class BYPASSED also appears, the reason-id for a held SYSMOD was ignored because BYPASS HOLDCLASS(class) was specified on the APPLY or ACCEPT command.*

Note that these unsatisfied requisite conditions can occur for any of the following reasons:

1. The named SYSMODs failed before.
2. SMP/E could not find the SYSMODs in the GLOBAL zone at APPLY or ACCEPT.
3. SMP/E could not find the SYSMODs in the distribution zone at RESTORE.
4. The SELECT or GROUP list incorrectly specified the set of SYSMODs for SMP/E to APPLY, ACCEPT, or RESTORE.

# GIM3XX

**System Action:** See GIM302 or GIM420.

**Programmer Response:** When this message follows GIM302, check the following:

1. Did SMP/E successfully install the named requisite installed on the target system?
2. Did SMP/E select the named requisite for installation during the APPLY or ACCEPT step for the SYSMOD named in message GIM302?

If SMP/E is installing the requisite SYSMOD during the current APPLY or ACCEPT step, look at the preceding messages on SMPOUT to find why the requisite SYSMOD stopped.

*Note: If a SYSMOD that supersedes the named requisite fails, so might the requisite SYSMOD.*

When this message follows GIM420, you do not need to do anything.

## GIM362s INVALID UPDATE INPUT TEXT

### Explanation:

The text following an ++UPDTE, ++MACUPD, or ++SRCUPD modification control statement had one of the following errors:

- There was no ./ CHANGE statement.
- There was more than one ./ ENDUP or ./ CHANGE statement.
- Sequence numbers of the UPDATE input text card, columns 73 through 80, are lower in the collation sequence than those of the previous text card. For both SMP/E and IEBUPDTE processing, sequence numbers must be in increasing, collating sequence.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Fix the problem and run RECEIVE again.

## GIM363s SYSMOD CONSTRUCTION ERROR:

```
{value1 APPEARS AS operand1 AND operand2 OPERAND}  
{++IF FMID OPERAND NAMES ++VER FMID OPERAND}  
{++VER MODIFICATION CONTROL STATEMENTS INCONSISTENT}  
{++VER value2 OPERAND INVALID ON SERVICE SYSMOD}  
{++ASSIGN MCS OUT OF SEQUENCE}
```

### Explanation:

- *value1* - value of *operand1* and *operand2* operands
- *operand1* - first operand
- *operand2* - second operand
- *value2* - ++VER modification control statement operand.

The following conditions are possible:

- *value1* APPEARS AS *operand1* AND *operand2* OPERAND

You specified the same value, *value1*, for operands *operand1* and *operand2*.

- ++IF FMID OPERAND NAMES ++VER FMID OPERAND

The FMID value on the ++IF modification control statement is the same as the FMID value on the ++VER modification control statement.

- ++VER MODIFICATION CONTROL STATEMENTS ARE INCONSISTENT

One of the following caused the error message:

- The ++VER modification control statement specified an FMID, but an applicable ++VER modification control statement, which SMP/E previously met, did not specify an FMID.
- The ++VER modification control statement did not specify an FMID operand, but a ++VER modification control statement, which SMP/E previously met, did specify an FMID operand.

- ++VER *value2* OPERAND INVALID ON SERVICE SYSMOD

The operand *value2* is invalid for a service SYSMOD.

- ++ASSIGN MCS OUT OF SEQUENCE

SMP/E expects the next statement after ++ASSIGN to be one of the following:

- ++FUNCTION, ++PTF, ++APAR, or ++USERMOD
- Another ++ASSIGN
- An end-of-file.

None of these followed the ++ASSIGN MCS statement.

**System Action:** SYSMOD processing stops.

If ++ASSIGN MCS OUT OF SEQUENCE appears in the message, none of the SYSMODs between this ++ASSIGN statement and the next valid ++ASSIGN statement, or an end-of-file, will be received. Likewise, no ++ASSIGN SOURCEID TO statements up to the next valid ++ASSIGN statement or end-of-file will be processed.

**Programmer Response:** Fix the problems and rerun the job.

# GIM3XX

**GIM364s DELETE ERROR, SYSMOD *sysmod1* DELETES *sysmod2* BOTH OF WHICH ARE BEING {APPLIED | ACCEPTED}**

**Explanation:**

- *sysmod1* - deleting SYSMOD-ID
- *sysmod2* - deleted SYSMOD-ID.

Function SYSMOD *sysmod1* and function SYSMOD *sysmod2* are both eligible for SMP/E to process. However, *sysmod1*'s ++VER modification control statement specifies *sysmod2* as a DELETE operand value.

**System Action:** SYSMOD *sysmod1* processing stops. Message GIM370 follows to show that the function also stops.

**Programmer Response:** SMP/E cannot process the two SYSMODs concurrently. Use the SELECT, GROUP, or EXCLUDE operands on the APPLY or ACCEPT command.

**GIM369s SYSMOD *sysmod* SELECTED. NOT FOUND ON SMPPTFIN**

**Explanation:**

- *sysmod* - SYSMOD-ID.

The RECEIVE command specified SYSMOD *sysmod* in the SELECT list, but *sysmod* was not in the SMPPTFIN data set.

**System Action:** RECEIVE processing continues for other SYSMODs in the SELECT list.

**Programmer Response:** Put SYSMOD *sysmod* in SMPPTFIN and rerun the job.

**GIM370s *command* PROCESSING TERMINATED BECAUSE FUNCTION SYSMOD *sysmod* FAILED**

**Explanation:**

- *command* - SMP/E command
- *sysmod* - SYSMOD-ID.

When a function SYSMOD fails, SMP/E processing stops.

**System Action:** Command processing stops.

**Programmer Response:** Do one of the following:

- Fix the error that caused the SYSMOD to fail.
- Exclude the failing SYSMOD from processing.

**GIM372s** NPRES ERROR - SYSMOD *sysmod1* NPRES *sysmod2* BOTH OF WHICH ARE BEING {APPLIED | ACCEPTED}

**Explanation:**

- *sysmod1* - SYSMOD-ID of a function SYSMOD
- *sysmod2* - SYSMOD-ID of a function SYSMOD.

Function SYSMOD *sysmod1* uses the NPRES operand to list function SYSMOD *sysmod2* as a negative prerequisite. However, APPLY or ACCEPT is processing both *sysmod1* and *sysmod2* concurrently.

**System Action:** Message GIM370 follows this message. Function processing stops.

**Programmer Response:** Exclude one of the named SYSMODs from processing and rerun the job.

**GIM373s** SYSMOD *sysmod* HAS MORE THAN ONE APPLICABLE ++VER MODIFICATION CONTROL STATEMENT

**Explanation:**

- *sysmod* - SYSMOD-ID.

During APPLY or ACCEPT, SYSMOD *sysmod* had two or more ++VER modification control statements whose FMID operand specified one of the following:

- Functions that SMP/E applied or accepted
- Functions that SMP/E was applying or accepting concurrently.

As a result, SMP/E could not determine the FMID value to assign to this SYSMOD.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Fix the ++VER modification control statement for the named SYSMOD.

**GIM374s** {PREREQ | FMID} SYSMOD *sysmod* CANNOT BE PROCESSED

**Explanation:**

- *sysmod* - SYSMOD-ID.

This message follows message GIM226. Processing for the SYSMOD in GIM226 stopped because SMP/E could not determine which SYSMOD to process first: the one in GIM226 or the one in GIM374.

# GIM3XX

The following example shows the error:

```
++ FUNCTION(F000001).
++ VER(Z038) PRE(P000001).
++ MOD(IEYMYMOD).

++ PTF(P000001).
++ VER(Z038) FMID(F000001).
```

In this example, the function SYSMOD names a PTF SYSMOD as a prerequisite. However, SMP/E cannot process this prerequisite SYSMOD until it processes the function. This situation might also occur when two SYSMODs name each other as prerequisites.

**System Action:** Processing for the SYSMOD in the GIM226 message stops.

**Programmer Response:** Do the following:

1. Reject the SYSMODs in error.
2. Fix the FMID/PREREQ relationship.
3. Receive and either apply or accept the SYSMODs.

**GIM376s COPY PROCESSING FOR SMPTLIB FAILED, SYSMOD = *sysmod*, RC = *rc***

**Explanation:**

- *sysmod* - SYSMOD-ID
- *rc* - return code from IEBCOPY.

IEBCOPY processing for elements on relative files returned the code *rc*.

**System Action:** SYSMOD processing stops if the return code is nonzero and one of the following is true:

- The return code is greater than the return code you specified.
- The return code is greater than the default return code.

**Programmer Response:** Check the IEBCOPY output listing to find the cause of error. Fix the error and rerun the job.

**GIM377s SCRATCH FAILED FOR *lib* ON VOLUME *volser* - *reason***

**Explanation:**

- *lib* - ddname of data set
- *volser* - volume serial number
- *reason* - reason for failure.

SMP/E found an error while it was trying to scratch a relative file data set. The cause of the error is one of the following:

- CURRENTLY DOCUMENTED REASONS

Unchanged.

- DATASET NOT FOUND

SMP/E could not find the data set it wanted to scratch on any of the volumes that the SMPTLIB DD statement specified.

- ERROR CODE = X“nn”

See the SCRATCH macro return codes in the *System Programming Library: Data Management* manual for your operating system.

- SMPTLIB HAS A DUMMY STATUS

The SMPTLIB DD statement has either the DUMMY keyword or DSN=NULLFILE.

**System Action:** Processing continues for the SYSMOD for which SMP/E allocated the data set.

**Programmer Response:** Find the error, and if SMP/E requires it, scratch the data set without using SMP/E.

**GIM378s** *mcs1* AND *mcs2* FOR ELEMENT *name* APPEAR IN SAME SYSMOD

**Explanation:**

- *mcs1* - MOD, MAC, SRC, ZAP, MACUPD, or SRCUPD
- *mcs2* - MOD, MAC, SRC, ZAP, MACUPD, or SRCUPD
- *name* - element name.

Modifications *mcs1* and *mcs2* for the same element are invalid.



# GIM3XX

The following table shows the valid pairs of modifications to an element in the same SYSMOD:

	MOD	ZAP	SRC	SRCUPD	MAC	UPDTE/ MACUPD
MOD			X	X	X	
ZAP					X	X
SRC	X				X	X
SRCUPD	X				X	X
MAC	X	X	X	X		
UPDTE/ MACUPD	X	X	X	X		

**System Action:** SMP/E does not receive the SYSMOD.

**Programmer Response:** Fix the problem and rerun RECEIVE.

**GIM379s SYSMOD *sysmod* SELECTED FOR *command* { IS SUPERSEDED | IS IN ERROR | IS DELETED | HAS NO APPLICABLE ++VER MODIFICATION CONTROL STATEMENTS | HAS HAD RESTORE ATTEMPTED }**

**Explanation:**

- *sysmod* - SYSMOD-ID
- *command* - APPLY or ACCEPT.

SMP/E cannot APPLY or ACCEPT SYSMOD *sysmod* for the following reason:

- IS SUPERSEDED - SMP/E has already applied or accepted one or more SYSMODs that supersede SYSMOD *sysmod*.
- IS IN ERROR - SMP/E marked SYSMOD *sysmod* with the ERROR status in the GLOBAL zone.
- HAS NO APPLICABLE ++VER MODIFICATION CONTROL STATEMENT - a SELECT or GROUP list specified SYSMOD *sysmod*, but the ++VER modification control statement in *sysmod* did not specify an SREL/FMID that matched the SREL/FMID in the target zone or distribution zone.
- IS DELETED - SMP/E deleted SYSMOD *sysmod* and cannot restore it.
- HAS HAD RESTORE ATTEMPTED - SMP/E partially restored SYSMOD *sysmod* and marked it as a RESTORE error.

**System Action:** SYSMOD *sysmod* processing stops.

**Programmer Response:**

- If SMP/E marked the SYSMOD with the ERROR status, do the following:
  1. RECEIVE or RESTORE the SYSMOD again.
  2. Attempt the APPLY or ACCEPT again.
- If the SYSMOD has no applicable ++VER modification control statement, make sure that:
  1. The SREL on the ++VER matches the SREL in the zone definition entry.
  2. The FMID on the ++VER matches a function SYSMOD entry currently on the zone you are processing.

**GIM380s SYSMOD *sysmod* SELECTED FOR {APPLY | ACCEPT} IS IN ERROR ON THE SMPPTS****Explanation:**

- *sysmod* - SYSMOD-ID.

A SYSMOD in the SMPPTS data set was selected for APPLY or ACCEPT processing, but was in error.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Receive the SYSMOD again to remove the error condition in the SMPPTS. Then try again to apply or accept it.

**GIM381s *mcs* FROM SYSMOD *sysmod* APPLIES TO ELEMENT *name* DELETED BY ANOTHER SYSMOD BEING PROCESSED****Explanation:**

- *mcs* - ZAP, MACUPD, or SRCUPD
- *sysmod* - SYSMOD-ID of SYSMOD supplying *mcs*
- *name* - element name.

The specified modification control statement referred to element *name*, which no longer exists.

**System Action:** APPLY or ACCEPT processing for the SYSMOD stops.

**Programmer Response:** Fix the problem and rerun the job.

# GIM3XX

**GIM382s ID CHECK PROCESSING SYSMOD *sysmod* - {*type name1* | ASSEMBLY *name2* FOR *type name1*}**

**Explanation:**

- *sysmod* - SYSMOD supplying element *eee*
- *name1* - element name
- *type* - element type
- *name2* - assembly element name.

An error or warning condition occurred when SMP/E was checking the relationship between the following:

- An element in SYSMOD *sysmod*
- Elements in other SYSMODs that SMP/E has already installed or is installing on your system.

One or more GIM319 messages for APPLY/ACCEPT and one or more GIM430 messages for RESTORE follow this message, describing the validity check condition(s) that failed.

If ASSEMBLY is in the message, an ID check error exists because of the relationship between the following:

- An assembly for a source module or macro
- The element on the target system that SMP/E will replace by the object module from the assembly.

The severity of this message depends on the BYPASS options you specified on the APPLY or ACCEPT command.

**System Action:** SYSMOD processing stops if the following conditions occur:

1. You did not specify BYPASS(ID).
2. SMP/E did not previously install the SYSMOD on the target system.
3. The SYSMOD supplies any of the following:
  - A replacement element - This occurs if the SYSMOD does not specify the RMID and all UMIDs of the previously processed version of the element in the PRE or SUP operands.
  - An update element - This occurs if the SYSMOD's PRE and SUP operands do not specify the RMID of the previously processed version of the element.

SMP/E does not stop processing the SYSMOD:

- If the SYSMOD supplies an update element and specifies the RMID of the previously processed version of the element in the PRE or SUP operands. However, the SYSMOD does not PRE or SUP all UMIDs of the previously processed version of the element.

In this case, GIM319 follows this message and names the updates in the previously processed version of the element that the SYSMOD does not supersede or specify as a prerequisite. SMP/E performs the update that the SYSMOD supplies against the previously processed version of the element.

- If you did not specify BYPASS(ID) and the SYSMOD is a function SYSMOD that SMP/E previously installed on the target system. SMP/E also does not stop processing if it can determine that the target system is at a higher level than the SYSMOD it is reinstalling.

The target system is considered to be at a higher level if one of the following is true:

- a. The RMID of the element from the SYSMOD differs from the RMID of the element on the target system. The RMID of the element from the SYSMOD is on the target system.

*Note: SMP/E does not issue GIM382. SMP/E does not process the element from the SYSMOD it is reinstalling, and the higher-level version of the element stays on the target system.*

- b. The RMID of the element from the SYSMOD is the same as the RMID of the element on the target system, and UMIDs are associated with the target system element.

*Note: SMP/E issues GIM382, followed by GIM319. These messages name the updates to the target system element that the SYSMOD does not supersede or specify as prerequisites. SMP/E does not process the element from the SYSMOD that it is reinstalling, and the higher level element stays on the target system.*

If you specified BYPASS(ID), processing of the SYSMOD does not stop for any ID checks that message GIM382 reports. SMP/E selects the element named for installation, and processing of the SYSMOD continues.

#### **Programmer Response:**

Carefully check the messages that follow this message in the output listing. If SMP/E already installed the element named, it may regress IBM-supplied service and/or user-supplied modifications included in the SYSMODs that are named in subsequent GIM319 messages.

Specify BYPASS(ID) on the APPLY or ACCEPT command to prevent SMP/E from stopping SYSMOD processing. However, you may lose the modifications included in the SYSMODs named in subsequent GIM319 messages. Rework and reinstall these modifications.

## GIM3XX

**GIM383s** FUNCTION *sysmod1* (FMID *value1*) SUPERSEDES FUNCTION *sysmod2* (FMID *value2*) BUT THE FMIDS ARE NOT EQUAL

**Explanation:**

- *sysmod1* - SYSMOD-ID of a function SYSMOD
- *value1* - FMID value for SYSMOD-ID *sysmod1*
- *sysmod2* - SYSMOD-ID of a function SYSMOD
- *value2* - FMID value for SYSMOD-ID *sysmod2*.

SMP/E is concurrently processing function SYSMOD *sysmod2* and function SYSMOD *sysmod1*, which supersedes SYSMOD *sysmod2*. However, the two SYSMODs have different FMID values, which causes SMP/E to select the wrong SYSMOD.

**System Action:** Processing for SYSMOD *sysmod1* stops. Message GIM370 follows this message and SMP/E command processing stops.

**Programmer Response:** Fix the ++VER modification control statement for one of the function SYSMODs and rerun the job.

**GIM384s** DELETE FUNCTION *sysmod1* IS SUPERSEDED BY FUNCTION *sysmod2*

**Explanation:**

- *sysmod1* - SYSMOD-ID
- *sysmod2* - SYSMOD-ID.

SMP/E is concurrently processing function SYSMOD *sysmod1* and function SYSMOD *sysmod2*, which supersedes SYSMOD *sysmod1*. However, SYSMOD *sysmod1* deletes other functions.

**System Action:** Message GIM370 follows this message and SMP/E command processing stops.

**Programmer Response:** Decide which SYSMOD not to process and rerun the job.

**GIM385s** SYSMOD *sysmod* TERMINATED BY USER EXIT RETURN CODE

**Explanation:**

- *sysmod* - SYSMOD-ID.

The SMP/E user exit procedure set a return code of 8 or greater.

**System Action:** SMP/E does not receive SYSMOD *sysmod*.

**Programmer Response:** None.

**GIM386s SYSMOD *sysmod* ALREADY RECEIVED****Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E has already successfully received SYSMOD *sysmod* into the SMPPTS data set.

**System Action:** SMP/E does not receive the SYSMOD *sysmod* again.

**Programmer Response:** To receive the new version of the SYSMOD, do the following:

1. Use the REJECT command to reject the SYSMOD from the SMPPTS.
2. Execute RECEIVE against the new version of the SYSMOD.

**GIM387s SYSMOD *sysmod* NOT SELECTED****Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E found *sysmod* in SMPPTFIN, but you did not specify it as a value of the SELECT operand.

**System Action:** SMP/E does not receive SYSMOD *sysmod*.

**Programmer Response:** None.

**GIM388s SYSMOD *sysmod* EXCLUDED****Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E found *sysmod* in SMPPTFIN, but you specified it as a value of the EXCLUDE operand.

**System Action:** SMP/E does not receive SYSMOD *sysmod*.

**Programmer Response:** None.

**GIM389s SYSMOD *sysmod* HAS NO APPLICABLE ++VER MODIFICATION CONTROL STATEMENT****Explanation:**

- *sysmod* - SYSMOD-ID.

SYSMOD *sysmod* did not have a ++VER modification control statement that named an SREL and/or an FMID in the GLOBAL zone GLOBALZONE entry.

## GIM3XX

**System Action:** SMP/E does not receive SYSMOD *sysmod*.

**Programmer Response:** To receive the SYSMOD, do one of the following:

- Specify BYPASS(FMID) on the RECEIVE command.
- Update the GLOBAL zone GLOBALZONE entry using a UCL statement to include the required SREL and/or FMID.

However, if SYSMOD *sysmod* is a service SYSMOD ( ++PTF, ++USERMOD, or ++APAR), one of the following must occur.

- The SYSMOD must have at least one ++VER modification control statement with an FMID.
- You must specify PARM = "FMID = *fmid*" when SMP/E runs. *fmid* is the SYSMOD-ID of a function.

### GIM390s SYSMOD *sysmod* SELECTED BUT COULD NOT BE RECEIVED

**Explanation:**

- *sysmod* - SYSMOD-ID.

You specified SYSMOD *sysmod* as an operand of the SELECT keyword on the RECEIVE command, but SMP/E did not successfully receive it.

**System Action:** RECEIVE processing continues.

**Programmer Response:** Refer to preceding messages in the output listing to find why SMP/E did not receive the SYSMOD.

### GIM391s SYSMOD *sysmod* TERMINATED DURING PROCESSING OF RELFILE ELEMENTS

**Explanation:**

- *sysmod* - SYSMOD-ID.

Errors occurred when SMP/E was trying to load elements from the IEBCOPY unloaded data sets to the TLIB data sets.

The following conditions stop RECEIVE from processing relative files:

- SMP/E could not position the SMPPTFIN data set because of one of the following:
  - An I/O error
  - An invalid data set name on a SMPPTFIN relative file.

For correct naming conventions see "Relative File Technique" in the *SMP/E Reference*.

- SMP/E could not allocate a data set on the TLIB volume.
- The IEBCOPY return code was nonzero. (Look at the IEBCOPY SYSPRINT output to find the cause.)

**System Action:** SMP/E does not receive SYSMOD *sysmod*, and RECEIVE processing stops. SMP/E scratches the TLIBs and deletes the SYSMOD from the SMPPTS.

**Programmer Response:** Fix any errors and run RECEIVE again for the SYSMOD.

#### GIM392s SYSMOD *sysmod* NOT RECEIVED

**Explanation:**

- *sysmod* - SYSMOD-ID.

There is not a valid SYSMOD and/or MCS entry in the SMPPTS that represents SYSMOD *sysmod*.

**System Action:** SMP/E does not receive the SYSMOD.

**Programmer Response:** Refer to preceding messages in the output listing to find why SMP/E did not receive the SYSMOD.

#### GIM393s SYSMOD *sysmod* SUCCESSFULLY RECEIVED

**Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E successfully received the SYSMOD and created a SYSMOD and an MCS entry for SYSMOD *sysmod* on the SMPPTS.

**System Action:** None.

**Programmer Response:** None.

#### GIM394s SYSMOD *sysmod* RELFILE ELEMENTS LOADED [MAX COPY RETURN CODE = *rc*]

**Explanation:**

- *sysmod* - SYSMOD-ID
- *rc* - maximum nonzero return code from copy.

SMP/E successfully loaded the elements from the unloaded copy data sets for *sysmod* to a TLIB data set. This TLIB data set is for further processing by APPLY and/or ACCEPT.

If "MAX COPY RETURN CODE" is part of the message, copy processing returned a nonzero return code less than or equal to the acceptable return code that you specified in the GLOBAL zone UTILITY entry for copy.



## GIM3XX

**System Action:** SYSMOD processing continues. SMP/E updates the SYSMOD entry in the SMPPTS to show that it successfully received SYSMOD *sysmod*.

**Programmer Response:** Look at the copy SYSPRINT output to find the cause of the nonzero return code so that the return code does not adversely affect further processing.

### GIM395s SYSMOD *sysmod* HAS RELFILE ELEMENTS

**Explanation:**

- *sysmod* - SYSMOD-ID.

SYSMOD *sysmod* put some of its elements from an unloaded IEBCOPY data set in a subsequent file on the SMPPTFIN data set.

**System Action:** If GIM392 does not accompany this message, SMP/E loads the elements from the unloaded IEBCOPY data sets.

**Programmer Response:** Look for the following pair of messages:

1. GIM394 SYSMOD *sysmod* RELFILE ELEMENTS LOADED
2. GIM393 SYSMOD *sysmod* SUCCESSFULLY RECEIVED.

These messages will appear after all modification control statements for all SYSMODs that SMP/E lists in SMPOUT.

### GIM396s SYSMOD *sysmod* HAS NO ++VER MODIFICATION CONTROL STATEMENT

**Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E did not find a ++VER modification control statement for SYSMOD *sysmod*.

**System Action:** SMP/E does not receive SYSMOD *sysmod*.

**Programmer Response:** To receive the SYSMOD, include an applicable ++VER modification control statement and run RECEIVE again.

### GIM397s SYSMOD *sysmod* HAS NO ELEMENTS

**Explanation:**

- *sysmod* - SYSMOD-ID.

SYSMOD *sysmod* has no ++MOD, ++MAC, ++SRC, ++ZAP, ++MACUPD, or ++SRCUPD modification control statements. It does, however, have an applicable ++VER modification control statement and may have inline JCL.

**System Action:** SMP/E received SYSMOD *sysmod*.

**Programmer Response:** None.

**GIM398s SYSMOD *sysmod* SYNTAX OR CONSTRUCTION ERROR****Explanation:**

- *sysmod* - SYSMOD-ID.

RECEIVE processing detected a modification control statement syntax error or a SYSMOD construction error.

**System Action:** SMP/E does not receive SYSMOD *sysmod*.

**Programmer Response:** Scan the SMPOUT stream for more specific information about the syntax or construction error for SYSMOD *sysmod*.

**GIM399s ENTER JULIAN DATE OR "U" FOR GIMSMP**

**Explanation:** SMP/E is requesting the date it needs to record this SMP/E job.

**System Action:** None.

**Programmer Response:** Do one of the following:

- Enter the date as yyddd (yy = year, ddd = day).
- Enter "U" for the system IPL date.

**GIM400s SYSMOD *sysmod* ENCOUNTERED PREVIOUSLY ON SMPPTFIN****Explanation:**

- *sysmod* - SYSMOD-ID.

SYSMOD *sysmod* was either in a previous SMPPTFIN input stream, or it appeared more than once in the same SMPPTFIN input stream. SMP/E may or may not have successfully received this previous version of *sysmod*.

For example, if a PTF is applicable to several licensed products, the PUT tape it is shipped on contains multiple copies of the PTF—one for each product. As a result, when you receive such a PUT tape, you may get this message with a return code of 8. However, no real error occurred.

**System Action:** SMP/E does not receive SYSMOD *sysmod*. The previous version of *sysmod* is not changed.

**Programmer Response:** Do one of the following:

- To keep the earlier version of the SYSMOD, do nothing.
- To keep this version of the SYSMOD, do the following:
  1. Reject the SYSMOD.
  2. Correct the SMPPTFIN input stream so that only the SYSMOD you want appears.
- If you received a PUT tape with multiple versions of the same PTF, do nothing.

# GIM4XX

**GIM401s** SYSMOD *sysmod* SELECTED FOR {APPLY | ACCEPT | RESTORE} NOT FOUND ON *type* ZONE

**Explanation:**

- *sysmod* - SYSMOD-ID
  - *type* - zone type.
1. For APPLY or ACCEPT, SMP/E selected SYSMOD *sysmod* but did not find it on the GLOBAL zone.
  2. For RESTORE, SMP/E selected SYSMOD *sysmod* but did not find it in the target zone.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Check that you correctly specified SYSMOD-ID on the control statement.

**GIM402s** *keyword* IS INVALID FOR {SUPERSEDED ONLY | DELETED} SYSMOD

**Explanation:**

- *keyword* - UCL keyword.

If SUPERSEDED ONLY is part of the message, UCL processing produced a SYSMOD entry that did not have any MOD, MAC, or SRC subentries. SMP/E assumes that this SYSMOD entry is produced as a result of being superseded by another SYSMOD. However, your UCL request also either added or left the data specified by *keyword* in the SYSMOD entry. This data is only valid for SYSMODs that are not superseded.

If DELETED is part of the message, you specified the DEL operand on the UCL command. However, you also specified other operands that you may not specify with the DEL operand.

**System Action:** SMP/E does not do the UCL processing that you requested.

**Programmer Response:** Fix your UCL statement in one of the following ways:

- Supply a MOD, MAC, or SRC subentry.
- Delete *keyword* from the SYSMOD (if a DEL request).
- Do not specify *keyword* (if an ADD or a REP request).

Rerun the UCL statement.

**GIM403s ENQ FAILED FOR DATASET *lib*****Explanation:**

- *lib* - ddname of data set.

SMP/E issued an exclusive ENQ on data set *lib*, but the data set was not available.

**System Action:** Processing stops for the SYSMOD that needs the data set.

**Programmer Response:** When the data set is available for exclusive use, rerun the job for the affected SYSMOD.

**GIM404s *type name* TO BE RESTORED TO SMPCSI FROM SMPSCDS FOR SYSMOD *sysmod*  
NOT FOUND ON *lib*****Explanation:**

- *type* - entry type
- *name* - entry name
- *sysmod* - SYSMOD-ID
- *lib* - ddname of data set.

SMP/E tried to copy entry *type* and entry *name* from the SMPSCDS to the target zone during RESTORE processing of SYSMOD *sysmod*. However, SMP/E could not find the specified member on the data set *lib*.

**System Action:** One of the following occurs:

- If SMP/E could not find the entry on the target zone, it places the backup entry on the target zone.
- If SMP/E could not find the entry at all, it cannot do a backup.

In both cases, SYSMOD processing continues.

**Programmer Response:** Do the following:

1. Look at the LOG to see if SMP/E ran UCL processing for the member.
2. List the copied target zone entry to check that SMP/E copied the correct version of the entry.
3. If the entry is incorrect, use a UCL statement to make the appropriate updates.

# GIM4XX

## GIM405s DISTLIB ERROR FOR *type name* FROM SYSMOD *sysmod*

### Explanation:

- *type* - entry type
- *name* - entry name
- *sysmod* - SYSMOD-ID.

The DISTLIB specified on a modification control statement in SYSMOD *sysmod* is different from the distribution library that SMP/E found for the element on the target zone or distribution zone.

**System Action:** APPLY/ACCEPT processing for SYSMOD *sysmod* stops.

**Programmer Response:** Correct the distribution library on one of the following:

- The element modification control statement
- The distribution zone or target zone.

Rerun the job.

## GIM406s START OF SMPADDIN CONTROL STATEMENTS

**Explanation:** SMPADDIN control statements follow this message.

**System Action:** SMP/E processes the SMPADDIN statements.

**Programmer Response:** None.

## GIM407s END OF SMPADDIN CONTROL STATEMENTS

**Explanation:** SMPADDIN control statements end.

**System Action:** UNLOAD processing continues.

**Programmer Response:** None.

## GIM408s SYSMOD *sysmod* NOT APPLIED OR NOT ACCEPTED

### Explanation:

- *sysmod* - SYSMOD-ID.

SMP/E selected SYSMOD *sysmod* for REJECT processing and either applied or accepted the SYSMOD, but did not do both. SMP/E did not set the APP or ACC indicator in the GLOBAL zone SYSMOD entry.

**System Action:** Processing continues for this SYSMOD.

**Programmer Response:** None.

**GIM409s COPY {FAILED|SUCCESSFUL} - *type* = *name* LIBRARY = *lib* - SYSMOD = *sysmod* - RETURN CODE = *rc***

**Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of data set
- *sysmod* - SYSMOD-ID
- *rc* - return code.

Copy processing completed for the indicated member and data set with a return code of *rc*.

SMP/E copies multiple members to various data sets in one copy invocation. If this invocation fails, SMP/E assumes that all members failed. SMP/E marks all related SYSMODs with the ERROR status set.

**System Action:** SYSMOD processing stops if *rc* is nonzero and one of the following is true:

- The return code is greater than the return code you specified.
- The return code is greater than the default return code.

**Programmer Response:** If the copy failed and the SYSMODs have the ERROR status set, check the copy output listing to find the cause of error. Fix the error and rerun the job.

**GIM410s SYSMOD *sysmod1* AND *sysmod2* ZAP MODULE *mod***

**Explanation:**

- *sysmod1* SYSMOD-ID of SYSMOD that supplies ZAP for module
- *sysmod2* SYSMOD-ID of SYSMOD that supplies ZAP for module
- *mod* - module name.

SMP/E APPLY and ACCEPT will not process more than one ZAP to the same element during one APPLY or ACCEPT pass.

**System Action:** APPLY or ACCEPT processing stops for SYSMOD *sysmod1*.

**Programmer Response:** Do the following:

1. APPLY or ACCEPT SYSMOD *sysmod2*.
2. APPLY or ACCEPT SYSMOD *sysmod1*.

## GIM4XX

**GIM411s** *type name* TO BE RESTORED TO THE TARGET ZONE FROM SMPSCDS FOR SYSMOD *sysmod1* HAS BEEN MODIFIED BY SUBSEQUENT SYSMOD {SYSMOD2 | JCLIN | UCLIN}

**Explanation:**

- *type* - entry type
- *name* - entry name
- *sysmod1* - SYSMOD-ID.

While attempting to restore SYSMOD *sysmod1*, SMP/E tried to copy entry *type* and entry *name* from the BACKUP entry to the target zone. The target zone entry in the LAST UPDATE field showed that SMP/E processed *sysmod2* and used inline JCLIN or UCLIN to modify the entry.

**System Action:** SMP/E does not copy the entry from the BACKUP entry to the target zone. SYSMOD processing continues.

**Programmer Response:** Do the following:

1. List the specified target zone entry.
2. Check that the entry is valid, even after SMP/E restores the SYSMOD.
3. If the entry requires any modifications, use a UCL statement to make the changes.

**GIM412s** IN THE CURRENT ENVIRONMENT THE RELATIONSHIP BETWEEN THE FOLLOWING SET(S) OF SYSMODS IS INCORRECT OR AMBIGUOUS

**Explanation:** SMP/E could not determine the order to process the SYSMODs that a subsequent message lists. The current environment contains the SYSMODs already on the system and those that SMP/E is currently processing. SMP/E determines processing by the information on the ++VER modification control statement (FMID, VERSION, PRE, SUP).

**System Action:** Function processing stops.

**Programmer Response:** Do one of the following:

- Fix the ++VER modification control statement for the specified SYSMODs.
- Selectively process each SYSMOD in the order that SMP/E requires.

**GIM413s** SYSMOD = *sysmod* FMID = *value* PRE = *sysmods*

**Explanation:**

- *sysmod* - SYSMOD-ID
- *value* - value of the FMID
- *sysmods* - list of prerequisite SYSMODs.

This message follows message GIM412 and lists the SYSMODs for which SMP/E could not determine a processing order.

**System Action:** Command processing stops.

**Programmer Response:** See message GIM412.

**GIM415s** ELEMENT *name* DOES NOT EXIST ON {TARGET ZONE | DLIB ZONE } FOR *mcs*  
FROM SYSMOD *sysmod*

**Explanation:**

- *name* - element name
- *mcs* - ZAP, MACUPD, or SRCUPD
- *sysmod* - SYSMOD-ID.

SMP/E cannot update element *name* because there is no entry for that element in the target zone or distribution zone.

*Note:* This can happen if the SYSMOD that supplied the element stopped abnormally. In this case, there may be an element entry on the distribution zone or target zone that has no FMID and no RMID.

**System Action:** APPLY or ACCEPT processing stops for SYSMOD *sysmod*.

**Programmer Response:** Do the following:

1. List the target zone or distribution zone to find out if there is an entry for the element.

If you find an entry with no RMID, the entry represents an element that SMP/E does not consider to be in the target system.

2. Do one of the following:

- Install a SYSMOD to supply the element.
- Use UCLIN to properly initialize the FMID and RMID fields in the target zone or distribution zone entry.

**GIM418s** INLINE JCLIN PROCESSING {FAILED|SUCCESSFUL} FOR SYSMOD = *sysmod*

**Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E finished JCLIN processing for SYSMOD *sysmod*. Processing either completed successfully or failed.

**System Action:** One of the following occurs:

- If JCLIN completed successfully, processing continues for the SYSMOD.
- If JCLIN processing failed, processing stops for the SYSMOD.



## GIM4XX

**Programmer Response:** Do one of the following:

- If processing completed successfully, do nothing.
- If processing failed, look at the previous messages from JCLIN processing to find the cause of the error.
  1. Restore the SYSMOD.
  2. Correct the inline JCLIN element.
  3. RECEIVE and APPLY the SYSMOD again.

### GIM419s *mcs* MODIFICATION CONTROL STATEMENT NOT FOUND IN SYSMOD *sysmod* MCS ENTRY

**Explanation:**

- *mcs* - ++JCLIN, ++MOVE, ++DELETE, or ++RENAME
- *sysmod* - SYSMOD-ID.

SMP/E had set an indicator in the GLOBAL zone SYSMOD entry to show that the *mcs* MCS statement was contained in SYSMOD *sysmod*. However, SMP/E could not find the MCS statement in the SMPPTS MCS entry.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Do the following:

1. Be sure that you did not modify the SMPPTS MCS entry after SMP/E received the SYSMOD.
2. Reject the SYSMOD and correct the error by either removing or adding the indicated MCS statement.
3. Receive the SYSMOD again so that the status indicator in the GLOBAL zone SYSMOD entry shows the contents of the SMPPTS MCS entry.

### GIM420s FOLLOWING CONDITIONS BYPASSED FOR SYSMOD *sysmod*

**Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E bypassed SYSMOD termination because you specified the BYPASS option on the APPLY or ACCEPT control statement. However, the SYSMOD did not satisfy certain requisite and/or hold conditions. Message GIM359 follows this message and names the requisites or hold conditions that the SYSMOD did not satisfy.

**System Action:** APPLY or ACCEPT processing continues for SYSMOD *sysmod*.

**Programmer Response:** None.

**GIM422s MULTIPLE NAME CARDS FOUND IN LMODIN INPUT**

**Explanation:** When SMP/E was processing the UCLIN ++LMODIN modification control statement, it found more than one name card.

**System Action:** SMP/E does not do the UCLIN changes you requested.

**Programmer Response:** Change the UCLIN ++LMODIN modification control statement so that it applies only to the LMOD that the UCLIN change request identifies.

If you are changing more than one LMOD, break the ++LMODIN statement into those applicable to only one LMOD. Rerun the UCLIN to the multiple LMODs.

**GIM423s THE DATE RANGE SPECIFIED IS INVALID**

**Explanation:** SMP/E found an incorrect date range.

**System Action:** LIST processing stops.

**Programmer Response:** Correct the date range by specifying the following on the LIST LOG command:

1. Specify *mm* as 01 through 12.
2. Specify *dd* as 01 through 31.
3. Specify *yy* as 00 through 99.

**GIM424s GIMSMP EXEC PARM = *parameters***

**Explanation:**

- *parameters* - parameters on the EXEC statement.

This message lists the parameters that you specified on the EXEC statement.

**System Action:** None.

**Programmer Response:** None.

**GIM425s *keyword* KEYWORD REQUIRED WHEN OTHER QUALIFYING KEYWORDS ARE SPECIFIED**

**Explanation:**

- *keyword* - required keyword.

You specified the *keyword* keyword on an SMP/E command, but it is only valid when specified with another keyword.

For example, LIST PTF would be invalid if you omitted the SYSMOD keyword. To correct the error, specify LIST SYSMOD PTF.

## GIM4XX

**System Action:** SMP/E does not process the command.

**Programmer Response:** Specify the required keyword and rerun the job.

### GIM426s DELETE ERROR - SYSMOD *sysmod1* DELETES *sysmod2* WHICH IS ALREADY {APPLIED|ACCEPTED}

**Explanation:**

- *sysmod1* - SYSMOD-ID
- *sysmod2* - SYSMOD-ID.

During APPLY or ACCEPT processing, *sysmod1* would have deleted *sysmod2*, which was already installed. However, *sysmod1* was not specified on the SELECT operand.

**System Action:** SYSMOD processing stops. Message GIM370 follows this message and shows that the SMP/E command also fails.

**Programmer Response:** Do one of the following:

- If you do not want SMP/E to process SYSMOD *sysmod1*, do one of the following:
  - Use the REJECT command to reject SYSMOD *sysmod1*.
  - Use the EXCLUDE operand to exclude SYSMOD *sysmod1* from processing.
- If you want SMP/E to APPLY or ACCEPT SYSMOD *sysmod1*, specify it on the SELECT operand.

### GIM427s NPRES ERROR - SYSMOD *sysmod1* NPRES *sysmod2* WHICH IS ALREADY {APPLIED|ACCEPTED}

**Explanation:**

- *sysmod1* - SYSMOD-ID
- *sysmod2* - SYSMOD-ID.

During APPLY or ACCEPT processing, SYSMOD *sysmod1* specified SYSMOD *sysmod2* as a negative prerequisite. However, SMP/E already applied or accepted SYSMOD *sysmod2*.

**System Action:** SYSMOD *sysmod1* processing stops. Message GIM370 follows this message and shows that the SMP/E command failed.

**Programmer Response:** Do one of the following:

- If you do not want SMP/E to process SYSMOD *sysmod1*, do one of the following:
  - Use the REJECT command to reject SYSMOD *sysmod1*.
  - Use the EXCLUDE operand to exclude SYSMOD *sysmod1* from processing.

- If you do want SMP/E to APPLY or ACCEPT SYSMOD *sysmod1*, remove SYSMOD *sysmod2*.
  - If SMP/E is running APPLY, use the RESTORE command to remove SYSMOD *sysmod2*.
  - If SMP/E is running ACCEPT, the only way to remove SYSMOD *sysmod2* is to use the UCLIN command.

**GIM428s** RESTORE CANDIDATE *sysmod* TERMINATED BECAUSE IT {DELETES SYSMODS | HAS BEEN ACCEPTED | IS DELETED | IS NOT APPLIED | IS SUPERSEDED}

**Explanation:**

- *sysmod* - SYSMOD-ID.

During RESTORE processing, SMP/E found that SYSMOD *sysmod* was in the stated condition.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Do one of the following:

- If you do not want SMP/E to process SYSMOD *sysmod*, do one of the following:
  - Do not select SYSMOD *sysmod*.
  - Do not select any SYSMODs that refer to SYSMOD *sysmod*.
- If you want SMP/E to restore SYSMOD *sysmod*, note the following conditions:
  - DELETES SYSMODS - SMP/E cannot restore SYSMOD *sysmod*.
  - HAS BEEN ACCEPTED - SMP/E cannot restore SYSMOD *sysmod*.
  - IS DELETED - SMP/E cannot restore SYSMOD *sysmod*.
  - IS NOT APPLIED - Apply SYSMOD *sysmod* so SMP/E can restore it.
  - IS SUPERSEDED - SMP/E can restore the SYSMOD if you remove it from the SELECT list and specify GROUP for SYSMODs that refer to SYSMOD *sysmod*.

**GIM429s** *name* FOR SYSMOD *sysmod1* IS ALSO IN SYSMOD *sysmod2* WHICH IS {IN ERROR AND NOT BEING RESTORED | APPLIED BUT NOT ACCEPTED}

**Explanation:**

- *name* - element name
- *sysmod1* - SYSMOD-ID
- *sysmod2* - SYSMOD-ID.

## GIM4XX

SYSMOD *sysmod2* on the target zone has the same element name as SYSMOD *sysmod1*. However, one of the following situations has occurred:

- SYSMOD *sysmod2* is marked in ERROR.
- SMP/E applied SYSMOD *sysmod2* but is not restoring it.

**System Action:** SYSMOD *sysmod1* processing stops.

**Programmer Response:** Fix the problem by doing one of the following:

- Restore SYSMOD *sysmod2* along with SYSMOD *sysmod1*.
- Accept SYSMOD *sysmod2*.

### GIM430s {TARGET ZONE | DLIB ZONE} *type modids error condition*

**Explanation:** This message further explains the MODID error that GIM382 defines during RESTORE processing.

- *type* - FMID, RMID, or UMID
- *modids* - list of MODIDs
- *error condition* - see below.

SMP/E may detect the following error conditions:

- DOES NOT PRE/SUP distribution zone MODID - The target zone, RMID or UMID, SYSMOD does not have a correct PRE or SUP relationship with any distribution zone RMID or UMID. To be correct, one of the following must be true:
  - IS NOT BEING RESTORED DUE TO NOGO STATUS - Multiple updates to the same element are on the target zone. SMP/E cannot restore the named UMID to the element; therefore, SMP/E cannot restore any of the UMIDs. SMP/E marks the element as NOGO, and SYSMOD processing stops.
  - IS NOT FOUND AS target zone ELEMENTS FMID - The target zone element is invalid or missing.
  - The named SYSMOD must PRE or SUP a distribution zone RMID or UMID.
  - The named SYSMOD must PRE or SUP another SYSMOD that SMP/E is concurrently restoring and has a correct PRE or SUP relationship with a distribution zone RMID or UMID.
- IS ACCEPTED BUT NOT APPLIED - SMP/E does not apply a SYSMOD in the distribution zone MODID list. If you try to restore the element involved, SMP/E will effectively apply a SYSMOD that has not been applied to the operating system. Note that in this case, SMP/E does not apply a superseded SYSMOD on the target zone.
- IS ACCEPTED IN ERROR - Despite the error, SMP/E accepts a SYSMOD in the distribution zone MODID list.

- IS APPLIED - SMP/E applied a target zone MODID SYSMOD but did not accept it and is not restoring it.
- IS APPLIED BUT NOT ACCEPTED - SMP/E is not restoring a target zone MODID SYSMOD. The MODID SYSMOD is not in the distribution zone MODID list.
- IS BEING RESTORED - SMP/E is not restoring a target zone MODID SYSMOD, and there is no corresponding type of MODID (FMID, RMID, or UMID) in the distribution zone MODID list.
- IS DELETED - SMP/E deleted a target zone or distribution zone SYSMOD named in the MODID list. This can happen in one of the following situations:
  - A previous APPLY or ACCEPT failed while SMP/E was updating the target zone or distribution zone entries for the SYSMOD.
  - UCLIN modified the element or SYSMOD entry.
- IS NOT APPLIED - SMP/E did not find the listed distribution zone MODID in the element's target zone MODID list. This situation tells SMP/E that the distribution zone MODID SYSMOD was ACCEPTED without being APPLIED. RESTORE cannot continue without causing a potential mismatch between elements on the operating system.
- IS NOT BEING RESTORED - SMP/E applies a target zone MODID SYSMOD but does not accept it and is not restoring it.
- IS SUPERSEDED - A target zone or distribution zone SYSMOD named in the MODID list is a superseded SYSMOD. This can only happen in one of the following situations:
  - A previous APPLY or ACCEPT failed while SMP/E was updating the target zone or distribution zone entries for the SYSMOD.
  - UCLIN modified the element or SYSMOD entry.

**System Action:** Processing of the SYSMOD specified in GIM382 stops unless you specified BYPASS(ID).

**Programmer Response:** Do one of the following:

- If SMP/E is processing the wrong set of SYSMODs, change the select or group list to process the correct set.
- If, however, you specified the correct set, use BYPASS(ID) to RESTORE the SYSMODs.

# GIM4XX

**GIM431s** INSUFFICIENT INFORMATION AVAILABLE TO DETERMINE TARGET LIBRARY, *type* = *name* - SYSMOD = *sysmod*

**Explanation:**

- *type* - element type (MOD, MAC, SRC)
- *name* - element name
- *sysmod* - SYSMOD-ID.

One of the following conditions has occurred:

- During APPLY or ACCEPT processing, SMPLE found an element whose target library (system library for APPLY, distribution library for ACCEPT) could not be determined from the target zone for APPLY or distribution zone for ACCEPT.
- During APPLY processing, the DISTLIB for a module was missing from both the MOD entry and the ++MOD modification control statement.

**System Action:** APPLY or ACCEPT processing stops for the named SYSMOD.

**Programmer Response:** Give the necessary JCLIN or UCLIN information so SMP/E can create the necessary target zone or distribution zone entries to finish processing the element.

**GIM432s** ABNORMAL TERMINATION - CODE = {SYSTEM | USER} *code* - PROGRAM = *program*

**Explanation:**

- *code* - ABEND code
- *program* - program name.

An abnormal stop occurred, and SMP/E invoked the SMP/E recovery routine.

**System Action:** If the directories were in storage for write processing, SMP/E rewrote the directories of the affected SMP/E data sets to disk. SMP/E generates the SMP/E summary reports with the current element and SYSMOD status.

**Programmer Response:** Look at the following:

1. The SMP/E summary reports to find the element and SYSMOD status.
2. The ABEND dump for problem determination.

**GIM433s SYSMOD *sysmod1* IS SUPERSEDED BY *sysmod2*, WHICH IS NOT BEING RESTORED****Explanation:**

- *sysmod1* - SYSMOD-ID of the superseded SYSMOD that SMP/E is restoring
- *sysmod2* - SYSMOD-ID of superseding SYSMOD.

SYSMOD *sysmod1* is part of a RESTORE group. Because SYSMOD *sysmod2* supersedes SYSMOD *sysmod1*, SMP/E considers SYSMOD *sysmod2* to be part of the restore group. Since SMP/E is not restoring SYSMOD *sysmod2*, SMP/E cannot restore SYSMOD *sysmod1*.

**System Action:** SYSMOD *sysmod1* processing stops.

**Programmer Response:** Do one of the following:

- Eliminate SYSMOD *sysmod1* from RESTORE processing.
- Include SYSMOD *sysmod2* in RESTORE processing.

**GIM434s {MOD | ASSEMBLY} *name*: - SYSMOD = *sysmod* WILL NOT UPDATE ANY SYSTEM LIBRARIES****Explanation:**

- *name* - element name
- *sysmod* - SYSMOD-ID.

SMP/E will not apply the named element to any target system libraries.

**System Action:** Processing of the named SYSMOD continues.

**Programmer Response:** Do one of the following:

- If the element belongs on a target system library, check that the proper JCLIN ran to define the target system library that will get the element. Run the JCLIN and reapply the SYSMOD.
- If the element does not belong on a target system library, do nothing.

**GIM435s VERNUM MUST BE ENTERED BEFORE ANY OPTIONS REQUIRING VERNUM**

**Explanation:** You specified the VERNUM option after an option that required VERNUM in the SYSMOD entry. For example:

```
REP SYSMOD(UZ00001) PRE(UZ00001) VERNUM(003)
```

This is an error because the PRE option has a VERNUM associated with it, but you specified the VERNUM option after the PRE option.

**System Action:** SMP/E does not do the UCLIN changes requested.

**Programmer Response:** Change the order of options so that the VERNUM option is first.



# GIM4XX

## GIM436s UPDATE RESULTS IN SYSMOD WITH MULTIPLE VERNUM VALUES

**Explanation:** You specified the VERNUM option, but the resulting update caused the SYSMOD to have subentries with different VERNUM values.

**System Action:** SMP/E does not do the UCLIN changes requested.

**Programmer Response:** Do the following:

1. Check the existing subentries in the SYSMOD for their VERNUM values.
2. Do one of the following:
  - Specify the existing VERNUM values and rerun the UCLIN.
  - Replace all existing subentries that require the VERNUM value.

## GIM437s ESTAE/STAE PROCESSING TERMINATED WITH A RETURN CODE = *rc*

**Explanation:**

- *rc* - return code.

The error occurred in the ESTAE-STAE processor. The return code from the ESTAE-STAE processor was *rc*.

**System Action:** System processing stopped because of a nonzero return code from the ESTAE-STAE processing.

**Programmer Response:** For further explanation of the *rc* return code from the ESTAE-STAE processor, see the *Supervisor Services and Macro Instruction* manual (ESTAE Processing) for your operating system.

## GIM438s UNABLE TO INITIALIZE SMP UTILITY INTERFACE SUBTASK - REASON CODE = *rc* {- RETURN CODE = *reason* }

**Explanation:**

- *rc* - two-digit code that shows why SMP/E could not initialize the subtask (see below)
- *reason* - two-digit return code associated with reason codes 01, 02, 03. Reason code 04 has no associated return code.

SMP/E could not initialize the SMP/E subtask that interfaces with UTILITY programs for one of the following reasons:

- The IDENTIFY for the entry point of the subtask program received a return code greater than 4. The return code given with this reason code is the return code from IDENTIFY.

For an explanation of the IDENTIFY return codes, see the *Supervisor Services and Macro Instruction* manual for your operating system.

- The ATTACH for the subtask received a nonzero return code. The return code given with this reason code is from ATTACH.

For an explanation of the ATTACH return codes, see the *Supervisor Services and Macro Instruction* manual for your operating system.

- Subtask initialization failed since the subtask could not establish its ESTAE coverage.

The return code given with this reason code is from ESTAE.

For an explanation of the ATTACH return codes, see the *Supervisor Services and Macro Instruction* manual for your operating system.

- Subtask initialization failed for an unknown reason. There is no return code associated with this reason code.

**System Action:** SMP/E processing stops.

**Programmer Response:** Fix the error and rerun the job.

#### GIM439s RETRY {FAILED | SUCCESSFUL} for TARGETDD = *lib*

**Explanation:**

- *lib* - target ddname for the invoked utility.

SMP/E ran a RETRY operation after it found a B37-04, D37-04, or E37-04 ABEND on a utility target library. If an error occurs during the RETRY operation, the RETRY failed; otherwise, the RETRY is considered successful.

**System Action:** The system does one of the following:

- If the RETRY failed, command processing stops. (SMP/E may stop processing depending on where and how the retry operation failed.)
- If the RETRY is successful, SMP/E processing continues normally.

**Programmer Response:** Do one of the following:

- If the RETRY was successful, do nothing.
- If the RETRY failed, increase the size of the target library.

# GIM4XX

**GIM440s SMP SUBTASK {ABNORMAL | UNEXPECTED} TERMINATION -**  
**CODE = {type code-reason1 | ecb} -**  
**TARGETDD = ddname1 - PROGRAM = program**  
**{ABENDDDD = ddname2}**  
**{RETRY WILL BE ATTEMPTED | RETRY WILL NOT BE ATTEMPTED -**  
**REASON = reason2}**

**Explanation:** If the SMP/E subtask that interfaces with utility programs abended, the message lists the following:

1. The type of ABEND (SYSTEM or USER)
2. The ABEND code
3. The ABEND reason code.

If the SMP/E subtask stopped unexpectedly (non-VS2 environment), the message lists the value of the subtask termination ECB. The message gives the target DDNAME and the name of the utility program name that SMP/E is calling for either an abnormal or unexpected termination.

If SMP/E is not going to attempt a RETRY, the reason code shows why as follows:

- *type* - shows the type of ABEND that SMP/E met. The type of ABEND is either "SYSTEM" or "USER" for abnormal termination.
- *code* - shows the ABEND code in hexadecimal for abnormal termination.
- *reason1* - shows the ABEND reason code in hexadecimal for abnormal termination.
- *ecb* - shows the hexadecimal value of the subtask termination ECB for an unexpected stop.
- *ddname1* - shows the target (output) DDNAME for the invoked utility.

*Note: This DD does **not** necessarily show the data set where the out-of-space condition occurred.*

- *ddname2* - shows the DDNAME of the data set where the out-of-space condition occurred. SMP/E will display this DD when it attempts retry and when SMP/E does not attempt retry for reason codes 06, 07 and 08.
- *program* - shows the name of the utility program that SMP/E invoked before the abnormal or unexpected stop.
- *reason2* - a two-digit reason code that shows why SMP/E could not attempt the RETRY:
  - 01 - No SDWA (system diagnostic work area) was provided to the STAE/ESTAE routine.
  - 02 - A user ABEND occurred.

- 03 - The SMP/E command did not show RETRY.
- 04 - RETRY was already in progress when the ABEND occurred.
- 05 - The ABEND code was not B37, D37, or E37, or the ABEND reason code was not "04."
- 06 - You specified that the data set that caused the ABEND was not a candidate for RETRY.
- 07 - The ABEND occurred for a library other than TARGETDD. TARGETDD does not show the library that caused the ABEND.

Look at the system messages to determine the cause of the failure.

- 08 - User exit 2 cancelled the RETRY.
- 09 - SMP/E is running in a non-VS2 environment.

**System Action:** One of the following occurs:

- If SMP/E is to attempt a RETRY, RETRY processing occurs.
- If SMP/E does not attempt a RETRY, the SMP/E command processing stops with a return code of 12 in the situations described by reason codes "01" through "07" and "09."
- For reason code "08", the SMP/E command processing stops with a return code equal to the return code of the user exit routine (either 12 or 16).

*Note: If the user exit gives a return code of 16, all SMP/E processing will stop.*

**Programmer Response:** Do one of the following:

- If SMP/E is to attempt a RETRY, your action depends on the success or failure of the RETRY.
- If SMP/E does not attempt a RETRY, your action depends on the ABEND code and the reason why SMP/E did not attempt the RETRY.

**GIM441s** UMID NOT IN SUP LIST - SYSMOD = *sysmod1* ELEMENT = *type name* UMID = *sysmod2*

**Explanation:**

- *sysmod1* - SYSMOD-ID
- *type* - MOD, MAC or SRC
- *name* - element name
- *sysmod2* - SYSMOD-ID specified on MCS statement that is not superseded by the SYSMOD.

The SYSMOD did not supersede the UMID specified on a ++MOD, ++MAC or ++SRC modification control statement.

# GIM4XX

**System Action:** SYSMOD processing stops.

**Programmer Response:** If SMP/E included the update identified by the UMID in the element, supersede the UMID in the SYSMODs + + VER modification control statement.

## GIM443s UNUSUAL CONDITION FOUND - *identifier* - *text*

### **Explanation:**

- *identifier* - an identifier for the module that issued this message
- *text* - up to 100 bytes of text that describes the unusual condition.

SMP/E could not determine the cause of an abnormal condition that occurred during processing. These are the types of errors that may occur:

- Data area errors

SMP/E could not locate or access data that is supposed to be in the CSI or another data set.

- Internal processing errors

Several types of internal processing errors may occur:

- Parameters passed between modules were overlaid. The receiving module did not recognize the erroneous data.
- Unsupported I/O functions were requested. This may happen when a parameter list is overlaid.
- Requested resources were held for no reason.

- Phase errors

Each SMP/E command has a set of processing phases. A phase error occurs if the proper phases are not processed in order. For example, one of the following may have occurred:

- The first phase processed was not Phase 1 for the command.
- The current phase should have preceded the previous phase.
- The cleanup phase was not finished for the previous command.
- The current phase is unknown for this command.
- An unknown phase was requested. This may happen when a phase is skipped.

- VSAM errors.

VSAM errors may occur when SMP/E tries to access the CSI data set. For example, one of the following may have occurred:

- A physical or logical I/O error occurred when the CSI was being opened.
- SMP/E tried to disconnect from VSAM services but was not previously connected.
- The subentry SMP/E was looking for does not exist.

**System Action:** Generally, SMP/E will continue processing; however, system action depends on the severity associated with the message.

**Programmer Response:** Report the error to your support group. If you need more information about the error, use the identifier shown in GIM443 to determine which module issued the message. Then look up the issuing module and possible error in the following list.

<b>Module</b>	<b>Error</b>
<b>GIMCMFSA</b>	SMP/E tried to access a nonexistent entry.
<b>GIMDSMGR</b>	SMP/E could not get extra resources to process a command, because they were held without a reason.  SMP/E needed to update a data set, but the calling module did not provide the data set name.  A phase error occurred.
<b>GIMGLUPD</b>	A parameter passed to this module was overlaid.
<b>GIMMPAR3</b>	SMP/E could not find a regressed SYSMOD in the target or distribution zone.
<b>GIMMPBDL</b>	SMP/E tried to check the GIMUTTBL table for valid utilities, but the table was either too small or did not have an end-of-list entry.
<b>GIMMPCL</b>	A SYSMOD that was successfully processed has an element in common with a SYSMOD that failed. Therefore, the successful SYSMOD should also fail. However, SMP/E could not find it in the target or distribution zone.
<b>GIMMPDS1</b>	The subentry SMP/E was looking for does not exist. This error may occur when SMP/E is checking one of the following: <ul style="list-style-type: none"> <li>● The zone definition entry for the set-to zone</li> <li>● An OPTIONS entry</li> <li>● A UTILITY entry.</li> </ul>

## GIM4XX

**GIMMPMRM** SMP/E tried to add a subentry to a zone, but a duplicate subentry already existed.

An input IOP control block had invalid data.

A request was made for I/O functions that this module does not perform. These include requests to:

- Purge records written to the output DCB or the RCDS in the output buffer since the last STOW
- Rename a member in a partitioned data set
- Decode a member name.

**GIMMPTDD** When checking the libraries required for a load module, SMP/E found an ICT entry with a missing or invalid SYSLIB value.

**GIMMPUCA** UCLIN processing tried to delete a subentry but could not find it. However, a previous check showed that the subentry existed.

**GIMMPUCV** UCLIN processing tried to add a subentry, but a duplicate subentry already exists. However, a previous check showed that it did not exist.

**GIMVSMMSG** A VSAM error detected by another module needed further analysis.

**GIMZNDL** A parameter passed to this module has invalid data.

**GIM444s** **ASSEMBLY *name1* FOR {MAC | SRC} *name2* IN SYSMOD *sysmod* WILL NOT BE DONE - {ASSEMBLER INPUT NOT FOUND | DISTLIB NOT DETERMINED}**

### Explanation:

- *name1* - ASSEM entry that SMP/E is to assemble
- *name2* - MAC entry against which SMP/E will do the assembly
- *sysmod* - SYSMOD that issued the assembly.

One of the following has occurred:

- SMP/E could not find the ASSEM entries.
- SMP/E could not determine the DISTLIB for the modules specified to be assembled by SMP/E.

Therefore, SMP/E assumes that the module does not belong on your system.

**System Action:** SMP/E continues processing.

**Programmer Response:** For the following conditions:

- Assembler Input Not Found - Specify the assembler input by doing one of the following:
  - Put the DISTSRC keyword on the ++MAC/++MACUPD ( ++SRC/ ++SRCUPD) card.
  - Create a target/distribution zone assembler entry.
- DISTLIB Not Determined - Specify the DISTLIB for module by doing one of the following:
  - Put the DISTMOD keyword on the ++MAC/++MACUPD ( ++SRC/ ++SRCUPD) card.
  - Initiate a target/distribution zone MOD ENTRY for the module.

**GIM445s THE FOLLOWING MODIFICATIONS WILL BE OVERLAID BY SYSMOD**  
*{sysmod | DELETE}*

**Explanation:**

- *sysmod* - SYSMOD that SMP/E is installing.

When SMP/E was installing a FUNCTION SYSMOD, SMP/E detected USERMODs and/or APARS in elements that will be replaced by the FUNCTION installation.

*Note: If the message refers to "DELETE," SMP/E is losing the USERMOD because the SMP/E deleted the function to which the USERMOD was applied.*

**System Action:** SYSMOD processing continues. Message GIM446, which displays the overlaid USERMODs and/or APARS, follows this message.

**Programmer Response:** None.

**GIM446s {USERMOD | SYSMOD} *sysmod* IN *type name***

**Explanation:**

- *sysmod* - SYSMOD-ID
- *type* - element type (MOD, MAC or SRC)
- *name* - element name.

This message lists the USERMODs that the FUNCTION SYSMOD, which was named in the preceding GIM445 message, will overlay.

**System Action:** None.

**Programmer Response:** None.



# GIM4XX

## GIM447s SAVED OBJECT FOR MODULE *mod* FROM *sysmod* WILL BE REUSED

### Explanation:

- *mod* - module name for assembly that SMP/E is bypassing
- *sysmod* - SYSMOD-ID that is both current and last causer for assembly of module.

As a result of the REUSE option, two things happen:

1. SMP/E will suppress the assembly of source for module *mod*.
2. SMP/E will use the existing object module found on SMPWRK3 for this job.

**System Action:** Processing of the named SYSMOD continues.

**Programmer Response:** None.

## GIM448s ASSEMBLY FOR *mod* IS BEING DONE DUE TO {APAR/USERMOD MODIFICATION | PREFIX PROCESSING | LOAD MODULE BUILD PROCESSING}

### Explanation:

- *mod* - module name for assembly being done.

This message explains why SMP/E is ignoring the object and performing the assembly.

- APAR/USERMOD MODIFICATION - A macro modification in an APAR/USERMOD caused SMP/E to assemble the module named *mod*. SMP/E does the assembly to prevent regression of the macro modifications supplied by one of the following:
  - The APAR/USERMOD SYSMOD
  - The SYSMODs that replace the module but do not have the modification.

The ASSEMBLE indicator in the target/distribution zone module entry records the “MUST ASSEMBLE” condition. The SYSMOD that SMP/E is processing is not necessarily an APAR or USERMOD SYSMOD. The “MUST ASSEMBLE” indicator may have been set with UCLIN.

- PREFIX PROCESSING - SMP/E is assembling the module named *mod* because you specified the PREFIX keyword on a ++MAC or ++MACUPD modification control statement. Module *mod* could be suffixed by the PREFIX value.
- LOAD MODULE BUILD PROCESSING - SMP/E is assembling an existing module, *mod*, to include it in a new load module.

If both the PREFIX and “MUST ASSEMBLE” conditions exist, SMP/E will generate the “MUST ASSEMBLE” variable part (APAR/USERMOD MODIFICATION).

**System Action:** Processing continues.

**Programmer Response:** Do one of the following:

- If SMP/E no longer needs the “MUST ASSEMBLE” condition, use UCLIN to turn the “MUST ASSEMBLE” bit off in the target/distribution zone.
- Otherwise, do nothing.

**GIM449s FIELD *field* ADDED**

**Explanation:**

- *field* - field name.

After SMP/E processed the UCL statement, SMP/E checked data in the entry. If SMP/E did not add the field, the entry would be invalid.

**System Action:** SMP/E adds field *field* to the entry, and processing continues with the next command.

**Programmer Response:** Check for correctness. If necessary, correct the UCL statement and rerun it.

Examples:

1. If you are deleting the ACC field, you must also delete the ACCDATE field.
2. SMP/E will not delete the RMID if FMID is present on MOD, MAC, or SRC entries.

**GIM450s I/O ERROR OCCURRED FOR *type name* ON *lib* LIBRARY**

**Explanation:**

- *type* - entry type
- *name* - entry name
- *lib* - ddname of data set.

This message follows message GIM274 when SMP/E can determine the entry that caused the I/O error.

**System Action:** SMP/E processing stops.

**Programmer Response:** Fix the I/O error and rerun the job. The following is one way to fix the error:

1. UCLIN DELETE the entry that caused the I/O error.
2. Rebuild the entry with UCLIN.

Note that this may not always work, since UCL processing may also get an I/O error when it tries to do the DELETE.

# GIM4XX

**GIM451s** {SMPLOG | SMPLOGA} DATA SET FULL - LOG RECORDING SWITCHED TO {SMPLOGA | SMPOUT}

**Explanation:** Because the indicated LOG data set is full, SMP/E is writing LOG messages to an alternate data set.

**System Action:** SMP/E processing continues.

**Programmer Response:** Reallocate a larger SMPLOG or SMPLOGA data set for subsequent SMP/E runs.

**GIM452s** *sysmod* TO BE UPDATED ON THE SMPPTS DOES NOT EXIST

**Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E tried to update the APPLY zone identifier in SYSMOD *sysmod* during a RESTORE operation, but SMP/E could not find the SYSMOD in the GLOBAL zone.

**System Action:** Processing of SYSMOD *sysmod* continues.

**Programmer Response:** None.

**GIM453s** INSUFFICIENT DATA SPECIFIED - *data* REQUIRED FOR SYSMOD *sysmod*

**Explanation:**

- *data* - data (category, FMID, reason) that is missing
- *sysmod* - SYSMOD-ID.

**System Action:** SYSMOD processing stops. Processing continues with the next SYSMOD.

**Programmer Response:** Provide the missing data and resubmit the job.

**GIM454s** *type name* EXCLUDED FROM SYSMOD *sysmod* DUE TO FMID MISMATCH

**Explanation:**

- *type* - element type
- *name* - element name
- *sysmod* - SYSMOD-ID.

**System Action:** SYSMOD processing continues.

**Programmer Response:** Check that the FMID value specified in the SYSMOD being installed is the correct value for all the elements in the SYSMOD.

**GIM455s** SYSMOD *sysmod* WILL NOT BE {APPLIED | ACCEPTED}

**Explanation:**

- *sysmod* - SYSMOD-ID.

SMP/E selects the information, but does not apply or accept it.

**System Action:** None.

**Programmer Response:** None.

**GIM500s** ERROR OCCURRED IN *mcs1 sysmod*, PROCESSING MCS TYPE = *mcs2*, {ELEMENT NAME = *name* | ASSOCIATED WITH ++VER NUMBER *value1* | NUMBER = *value2*}

**Explanation:**

- *mcs1* - ( ++FUNCTION, ++PTF, ++APAR, or ++USERMOD) or blank, if unknown
- *sysmod* - SYSMOD-ID or blank, if unknown
- *mcs2* - ( ++VER, ++IF, ++HOLD, ++MOD, ++MAC, ++SRC, ++ZAP, ++MACUPD, ++UPDTE, ++SRCUPD, or ++JCLIN) or blank, if unknown
- *name* - element name associated with *mcs* or blank, if unknown
- *value1* - number of the ++VER statement associated with the ++IF statement that is in error
- *value2* - number of the ++VER statement that is in error.

*Note:* This message immediately follows message GIM203I, which shows that the line immediately before GIM203I has the syntax error.

**System Action:** Processing stops for the SYSMOD in error.

**Programmer Response:** Do one of the following:

- To find the SYSMOD in error, check the format of the keyword on the specified command. Fix the syntax error and rerun the job.
- If you cannot find the SYSMOD in error, use the RECEIVE LIST option to rerun your PTFIN stream. SMP/E will print out the entire PTFIN stream. Fix the syntax error and rerun the job.

**GIM501s** SPECIFICATION OF {REDO | BYPASS(APPLYCHECK) | BYPASS(ACCEPTCHECK) | BYPASS(APPLYCHECK,ACCEPTCHECK) } IS ONLY VALID IN SELECT/GROUP MODE

**Explanation:** SMP/E only allows you to specify the identified operands when processing in SELECT/GROUP mode.

**System Action:** SMP/E sets the return code to 12, and processing continues to check the syntax in SMPCTL.

**Programmer Response:** Remove the operand(s) that SMP/E does not allow and rerun the job.

## GIM5XX

**GIM502s** SYSMOD *sysmod* WILL NOT BE *command* - {REDO | BYPASS(APPLYCHECK) | BYPASS(ACCEPTCHECK) | BYPASS(APPLYCHECK,ACCEPTCHECK)} NOT SPECIFIED

**Explanation:**

- *sysmod* - SYSMOD-ID
- *command* - SMP/E command.

Command *command* does not process SYSMOD *sysmod* for the reason stated.

**System Action:** SYSMOD processing stops. SMP/E sets the return code to 8, and processing continues.

**Programmer Response:** If you want SMP/E to process SYSMOD *sysmod*, supply the missing operand stated in the message and rerun the job.

**GIM503s** NO FMIDSET RECORD FOUND FOR VALUE *value* - NO PROCESSING WILL BE DONE FOR THIS VALUE

**Explanation:** SMP/E did not find a FMIDSET record for the *value* specified.

The value has a length less than or greater than seven characters. Since the value is not a FMID value (that is, a length of seven), SMP/E cannot process for this value.

**System Action:** SMP/E sets the return code to 8, and processing continues for the next SMP/E command requested.

**Programmer Response:** Do one of the following:

- If you wanted the value to be a FMIDSET record value, add the FMIDSET record for the name specified by *value*.
- Otherwise, do nothing.

**GIM504s** PARM CSI= ON EXEC STATEMENT NOT VALID WITH SMPCSI DD PRESENT

**Explanation:** SMP/E encountered the stated condition when it was initializing the SMPCSI. You specified the CSI= parameter on the EXEC statement, and this is not valid with the SMPCSI DD statement.

**System Action:** SMP/E sets the return code to 16, and SMP/E processing stops.

**Programmer Response:**

- SMP/E cannot determine which SMPCSI it should use. Do one of the following:
  - If SMP/E should use the SMPCSI DD statement, rerun the job without the CSI= parameter.
  - If SMP/E should use the CSI= EXEC parameter, rerun the job without the SMPCSI DD statement.

**GIM505s** *command* FAILED FOR ZONE *zonename* [-LOAD MODULE BUILD PROCESSING CANNOT CHECK THE DISTRIBUTION ZONE]

**Explanation:**

- *command* - SMP/E command that failed
- *zonename* - the zone for which the command failed.

SMP/E requested *command* processing for the zone *zonename*, but the command did not complete successfully. For example, SMP/E may have failed to OPEN a distribution zone needed to build a new load module.

**System Action:**

- If load module build processing is mentioned in the message, SMP/E will also issue the following messages:
  - GIM674, which lists the load modules that SMP/E terminated
  - GIM675, which lists the modules that SMP/E needs to complete the load modules
  - GIM226, which lists the SYSMOD(s) that SMP/E terminated because of load module termination.
- If load module build processing is *not* mentioned, SMP/E did not process the requested command. SMP/E sets the return code to 16 for unrecoverable failures, or to 12 for failures that another SET command can recover.

**Programmer Response:** Fix the error and rerun the job.

**GIM506s** LINK {FAILED|SUCCESSFUL} - LMOD = *lmod* - LIBRARY = *lib* - RETURN CODE = *rc* - THE FOLLOWING CSECTS HAVE BEEN DELETED

**Explanation:**

- *lmod* - load module name
- *lib* - ddname of the data set
- *rc* - return code.

SMP/E ran a link-edit to remove a distribution library module from *lmod* in the library *lib*. The link-edit completed with return code *rc*. Message GIM507 follows this message and identifies the distribution library modules and the CSECTS that SMP/E removed.

## GIM5XX

**System Action:** One of the following occurs:

- If the return code is greater than 12, processing stops for the SYSMOD causing the delete.
- If the return code is less than or equal to 12, processing continues.

**Programmer Response:** Do the following:

1. If an error occurred in the link-edit, look at the link-editor SYSPRINT output to find the cause of error.
2. Re-APPLY the failing SYSMOD.

*Note: RC=12 is a normal return code when the linkage editor replaces a dummy CSECT.*

### GIM507s CSECT *name* IN DLIB MODULE *mod*

**Explanation:**

- *name* - CSECT name
- *mod* - DLIB module.

This message lists the DLIB module and the CSECTs that are in the DLIB module. These CSECTs were deleted from the LMOD and library as identified in message GIM206.

**System Action:** None.

**Programmer Response:** None.

### GIM508s ZONE *zonename* UNKNOWN - MUST BE DEFINED IN GLOBAL ZONEINDEX BEFORE USE [- LOAD MODULE BUILD PROCESSING CANNOT CHECK THE DISTRIBUTION ZONE]

**Explanation:**

- *zonename* - name of the unknown zone.

SMP/E could not find a ZONEINDEX for the zone specified on one of the following:

- The BOUNDARY value on the SET command
- The FORZONE or ZONESET used with the REPORT command.

If the error occurred during load module build processing, the message indicates this.

**System Action:**

- If load module processing is mentioned in the message, SMP/E will also issue the following messages:
  - GIM674, which lists the load modules SMP/E terminated
  - GIM675, which lists the modules SMP/E requires to complete the load modules
  - GIM226, which lists the SYSMOD(s) SMP/E terminated because of load module termination.
- If load module processing was *not* mentioned in the message, the following occurs:
  1. SMP/E sets the return code to 12.
  2. SMP/E syntax checks commands until it finds a valid SET and RESETRC command.
  3. SMP/E processing continues.

**Programmer Response:** Do one of the following:

- If you want SMP/E to process zone *zonename*, use UCLIN to define the zone in a GLOBAL zone ZONEINDEX. Rerun the job.
- Otherwise, do nothing.

**GIM509s SET COMMAND {REQUIRED BEFORE *command* CAN BE PROCESSED | PROCESSING FAILED. SMP\_CNTL WILL BE SYNTAX CHECKED ONLY UNTIL NEXT SET}**

**Explanation:** For the following conditions:

- “REQUIRED BEFORE *command* CAN BE PROCESSED” - SMP/E read a command from SMP\_CNTL, and there was no prior SET command.
- “PROCESSING FAILED. ...” - SMP/E met a failure when it was processing the SET command.

**System Action:** SMP/E set the return code to 12 and checked the syntax in SMP\_CNTL.

**Programmer Response:** For the following conditions:

- “REQUIRED BEFORE *command* CAN BE PROCESSED” - Put a SET command in SMP\_CNTL before the function that requires it. This will point SMP/E to the zone that it is going to process. Rerun the job.
- “PROCESSING FAILED. ...” - Fix the SET command failure and rerun the job.



## GIM5XX

**GIM510s** SUB-ENTRY TYPE *subtype* DOES NOT EXIST IN ENTRY *type name*

**Explanation:**

- *subtype* - subentry type
- *type* - entry type
- *name* - entry name.

UCLIN found that the subentry type is missing. UCLIN processing cannot continue for this statement.

**System Action:** Processing for this statement stops with a return code of 8. UCLIN processing continues with the next statement.

**Programmer Response:** Fix the error and rerun the statement for processing.

**GIM511s** BOUNDARY ERROR DETECTED BY MODULE GIMMPUCD

**Explanation:** Module GIMMPUCD could not determine the boundary for UCLIN processing.

**System Action:** SMP/E sets the return code to 12, and SMP/E processing stops.

**Programmer Response:** Report the error to your support group.

**GIM512s** REQUEST ONLY VALID FOR *zonetype*

**Explanation:**

- *zonetype* - zone type.

The command you entered (UCLIN or ZONEEDIT) does not support the specified operand for the zone you are processing.

**System Action:**

- UCLIN - SMP/E sets the return code to 8 and continues by processing the next UCL statement.
- ZONEEDIT - SMP/E sets the return code to 12 and stops ZONEEDIT processing.

**Programmer Response:** Check the *SMP/E Reference* to determine which operands can be used for which zones.

**GIM513s THE DSPREFIX CONTAINS MORE THAN 26 CHARACTERS**

**Explanation:** This message can occur in two different situations:

1. If SMP/E issued this message during UCLIN processing, the UCLIN statement specifies a DSPREFIX that has more than 26 characters.
2. If SMP/E issued this message during SMP/E initialization, the CSI execution parameter specifies a DSPREFIX that has more than 26 characters.

**System Action:** The action differs with each situation.

1. If you see this message during UCLIN processing, processing stops for the statement in error. Processing continues with the next UCLIN statement.
2. If you see this message during SMP/E initialization, SMP/E processing stops.

**Programmer Response:** Do one of the following:

- Fix the UCLIN statement and rerun the job.
- Correct the SMP/E execution parameter as appropriate and rerun the job.

**GIM514s {ZONE DEFINITION | OPTION ENTRY | UTILITY ENTRY} NOT FOUND FOR {ZONE | OPTION | UTILITY} *name***

**Explanation:** SMP/E did not find the record shown in the message for *name*.

**System Action:** The action is as follows:

- For the following conditions:
  - Variable type = OPTION entry
  - You did not specify the OPTION entry name on the SET command.
  - Function = LIST/UNLOAD/UCLIN.

SMP/E sets the return code to 4, and processing continues using the SMP/E defaults.

- Otherwise, SMP/E sets the return code to 12, and processing continues to syntax-check the SMP\_CNTL input.

**Programmer Response:** Do one of the following:

- If the record shown in the message is missing, supply UCLIN to add the record. Rerun the job.
- Otherwise, do nothing.

# GIM5XX

**GIM515s** *typename* DELETED FROM DISTLIB *dlib* BECAUSE DISTLIB CHANGED

**Explanation:**

- *type* - MODULE or ALIAS name
- *name* - name of the module or alias
- *dlib* - name of the old distribution library.

The module (alias) was moved to a new distribution library as a result of accepting the function sysmod.

**System Action:**

The named entry is deleted from the specified data set.

**Programmer Response:**

None.

**GIM516s** *command* NOT VALID WHEN SET TO *zonetype* TYPE ZONE [- WHEN INPUT ZONE IS GLOBAL]

**Explanation:**

- *command* - SMP/E command
- *zonetype* - GLOBAL or non-GLOBAL zone.

*command* is not valid for the type of zone *zonetype*.

If WHEN INPUT ZONE IS GLOBAL is part of the message, the SET is to a non-GLOBAL type zone; however, the input zone is a GLOBAL type zone. SMP/E cannot process a GLOBAL type zone, using the zone utility functions, against a non-GLOBAL type zone.

**System Action:** SMP/E sets the return code to 12, and processing continues to syntax-check SMP\_CNTL input.

**Programmer Response:** Do the following:

1. Make sure that the type of zone SMP/E set corresponds to the functions SMP/E allows for that type of zone.
2. Fix the conflicting data.
3. Rerun the job.

**GIM517s** *command* CAN NOT BE DONE - A {SREL VALUE | RELATED ZONE} IS REQUIRED

**Explanation:**

- *command* - SMP/E command.

SMP/E requested *command* processing, but the data shown in the message is not defined.

**System Action:** SMP/E set the return code to 12, and processing continues to syntax-check SMP\_CNTL input.

**Programmer Response:** Supply UCLIN to define the proper SREL or RELATED values in the zone definition and rerun the job.

**GIM518s ZONETYPE OF *zonetype1* FROM GLOBAL ZONEINDEX CONFLICTS WITH ZONETYPE OF *zonetype2* FROM ZONE DEFINITION**

**Explanation:**

- *zonetype1* - zone type defined in the GLOBAL ZONEINDEX
- *zonetype2* - zone type in the zone definition.

During setup processing, SMP/E found different zone types for the set-to zone in the following:

- The GLOBAL zone ZONEINDEX subentry
- The zone definition entry for the set-to zone.

**System Action:** SMP/E set the return code to 12, and processing continues to syntax-check SMP\_CNTL.

**Programmer Response:** The zone types for a given zone name must be consistent. Run UCLIN to fix one of the zone types and rerun the job.

**GIM519s ++RELEASE OR ++NULL INVALID WITHIN A SYSMOD**

**Explanation:** During RECEIVE processing of a SYSMOD, SMP/E found a ++RELEASE or ++NULL MCS.

**System Action:** SMP/E does not receive the SYSMOD.

**Programmer Response:** Fix the SYSMOD construction and RECEIVE again.

**GIM520s SYSMOD SPECIFIED ON ++HOLD IS NOT EQUAL TO ID OF SYSMOD CONTAINING THE ++HOLD**

**Explanation:** During RECEIVE processing of a SYSMOD, SMP/E found a ++HOLD statement that specified a SYSMOD; however, this SYSMOD-ID was not the same as the SYSMOD that SMP/E was receiving.

*Note: Only use a ++HOLD within a SYSMOD to specify that you want SMP/E to hold the containing SYSMOD.*

**System Action:** SMP/E does not receive the SYSMOD.

## GIM5XX

**Programmer Response:** Do one of the following:

- Fix the ++HOLD statement if its SYSMOD is wrong.
- If it is designed to hold the SYSMOD specified, process the ++HOLD statement through the SMPHOLD file.

### GIM521s HOLD OR RELEASE CATEGORY SPECIFIED MORE THAN ONCE

**Explanation:** SMP/E found a ++HOLD or ++RELEASE MCS that specified one of the following:

- More than one HOLD category (ERROR, SYSTEM, or USER)
- The same category more than once.

**System Action:** The following actions are possible:

- If ++HOLD appeared within a SYSMOD, SMP/E did not receive the SYSMOD.
- If SMP/E was processing the ++HOLD outside a SYSMOD (that is, from the SMPHOLD file), SMP/E did not process that ++HOLD or ++RELEASE.

**Programmer Response:** Fix the ++HOLD or ++RELEASE and resubmit the SMP/E command.

### GIM523s SET ZONE IS INCORRECT TO LIST RECORD(S) OR THE RECORD(S) DO NOT EXIST IN SET ZONE

**Explanation:** SMP/E did not find the LIST operand in the specified zone.

**System Action:** SMP/E ignores the LIST command. Processing continues with the next command.

**Programmer Response:** Do one of the following:

- Set the zone to the correct zone.
- Correct the LIST command.

*Note: Check that you did not specify the operands with the zones under which they are listed below.*

GLOBAL Zone	Target Zone	Distribution Zone
ASSEM	FMSET, FMIDSET	FMSET, FMIDSET
LMOD	OPTIONS	OPTIONS
MAC	UTILITY	UTILITY
MOD	DZONE,DLIBZONE	TZONE, TARGETZONE
SRC	GZONE, GLOBALZONE	GZONE, GLOBALZONE
DLIB	ZONESET	RES, RESTORE
RES, RESTORE		ZONESET
SUP		
NOSUP		
DEL, DELETE		
BYP, BYPASS		
XREF		
TZONE, TARGETZONE		
DZONE, DLIBZONE		
BACKUP		

**GIM524s UCLIN COMPLETED PROCESSING FOR *zonetype* *zonename***

**Explanation:**

- *zonetype* - zone type
- *zonename* - zone name.

UCLIN finished processing for *zonetype* *zonename*.

**System Action:** Processing continues.

**Programmer Response:** None.

**GIM525s END OF FILE ENCOUNTERED AFTER {ENDUCL | ENDZONEEDIT}. PERIOD MISSING.**

**Explanation:** SMP/E found an ENDUCL or ENDZONEEDIT command followed by an end-of-file. The period after the command is missing.

**System Action:** All prior UCLIN or ZONEEDIT processing was completed. A message follows to show whether command processing was successful. Processing continues.

**Programmer Response:** Fix the error and rerun the job.

## GIM5XX

### GIM526s THE UCLIN STATEMENT AND THE SET BDY COMMAND CONTAIN DIFFERENT {ZONE NAMES | ZONE TYPES}

**Explanation:**

The UCLIN statement points to a zone type or zone name that is not the same as the zone type or zone name in the SET command.

**System Action:** SMP/E processing for this statement stops. Processing continues for the next statement.

**Programmer Response:** Do one of the following:

- Fix the UCLIN statement and rerun it.
- Fix the zone specified for SET BOUNDARY and rerun this UCLIN statement.

### GIM527s THE DSNAME CONTAINS MORE THAN 44 CHARACTERS

**Explanation:** This message can occur in two different situations:

- If SMP/E issues this message during UCLIN processing, the UCLIN statement specifies a data set name that has more than 44 characters.
- If SMP/E issues this message during SMP/E initialization, the CSI EXEC parameter specifies a data set name that has more than 44 characters.

**System Action:** The action differs with each situation.

- If you see this message during UCLIN processing, processing stops for the statement in error. Processing continues with the next UCLIN statement.
- If you see this message during SMP/E initialization, SMP/E processing stops.

**Programmer Response:** Do one of the following:

- Fix the UCLIN statement and rerun the job.
- Correct the SMP/E EXEC parameter and rerun the job.

### GIM528s THE LOW LEVEL QUALIFIER OF THE DSNAME MUST BE - CSI

**Explanation:** The lowest level qualifier for a CSI data set name must be CSI.

**System Action:** One of the following will occur:

- If SMP/E issues this message during UCLIN processing, processing for the statement in error stops. Processing continues with the next UCLIN statement.
- If SMP/E issues this message during SMP/E initialization, SMP/E processing stops.

**Programmer Response:** Do one of the following:

- Fix the UCLIN statement and rerun the job.
- Fix the SMP/E EXEC parameter and rerun the job.

**GIM529s ZONE DELETION CAN ONLY BE PROCESSED BY THE ZONEDELETE COMMAND**

**Explanation:** UCLIN does not support a ZONEDELETE command. (Use the ZONEDELETE command to delete DLIB and target zones.)

**System Action:** SMP/E stops processing the statement and continues processing with the next statement.

**Programmer Response:** Fix the error and rerun this statement.

**GIM530s NO SMP 1.4 DATA SET SPECIFIED**

**Explanation:** You specified the CONVERT command, but you did not show which SMP4 data set that you want to convert.

**System Action:** SMP/E does not run the CONVERT command.

**Programmer Response:** Change the CONVERT command to include one of the SMP4 data set operands (CDS, ACDS, PTS, CRQ, or ACRQ) and rerun the job.

**GIM531s REQUEST TO CONVERT *type lib* INVALID WHEN BOUNDARY SET TO {GLOBAL | TARGET | DLIB} ZONE *zonename***

**Explanation:**

- *type* - CDS, ACDS, PTS, CRQ, or ACRQ
- *lib* - name of the DD statement specified on the CDS, ACDS, PTS, CRQ, or ACRQ CONVERT operand
- *zonename* - name of the target or DLIB zone from the previous SET command. Note that if the zone type is GLOBAL there will be no *zonename*.

The CONVERT command specified that SMP/E was to convert the SMP4 data set specified by *type lib*. However, the zone type that the previous command specified was not the correct type zone to receive that data.

**System Action:** SMP/E does not run the CONVERT command.

**Programmer Response:** Do one of the following:

- Change the SMP4 data set specification to be the same zone type as in the previous SET command.
- Change the previous SET command to the correct zone type.



## GIM5XX

The following table shows the correct relationship between SMP4 data set and zone type.

SMP4 DATASET	SMP/E ZONE TYPE		
	TARGET	DLIB	GLOBAL
CDS	X		
ACDS		X	
PTS			X
CRQ	X		
ACRQ		X	

### GIM532s REQUEST NOT VALID FOR GZONE

**Explanation:** The UCLIN processing that you requested for a GLOBAL zone is invalid.

**System Action:** SMP/E stops processing this statement and continues processing with the next statement.

**Programmer Response:** Fix the UCLIN request for GLOBAL zone processing and rerun this UCLIN statement.

### GIM533s CONVERT *operand* OPERAND IS VALID ONLY WHEN *dstype* CONVERSION IS REQUESTED

**Explanation:**

- *operand* - ASSEM, DLIB, LMOD, MAC, MCS, SRC, SYSMOD, or SYSTEM
- *dstype* - CDS, ACDS, PTS, CRQ, ACRQ, or a combination of these.

The CONVERT command specified an entry type *operand* that is not applicable to the SMP4 data set specified.

**System Action:** SMP/E does not run the CONVERT command.

**Programmer Response:** Do one of the following:

- Change the CONVERT command to specify the correct entry type and rerun the job.
- Fix the SMP4 data set and rerun the job.

The following table shows the valid entry type and SMP4 data set relationships.

ENTRY TYPE	SMP4 DATA SET				
	CDS	ACDS	PTS	CRQ	ACRQ
ASSEM	X				
DLIB	X				
LMOD	X				
MAC	X	X			
MCS			X		
MOD	X	X			
SRC	X	X			
SYSMOD	X	X	X	X	X
SYSTEM			X		

**GIM534s** *status* CONVERSION OF *dstype lib type* ENTRIES

**Explanation:**

- *status* - STARTING or COMPLETED
- *dstype* - CDS, ACDS, PTS, CRQ, or ACRQ
- *lib* - the name of the DD statement specified in the CDS, ACDS, PTS, CRQ, or ACRQ CONVERT operand
- *type* - ASSEM, DLIB, LMOD, MAC, MCS, MOD, SRC, SYSMOD, or SYSTEM.

CONVERT processing started or completed converting the entry type that you specified.

**System Action:** CONVERT processing continues normally.

**Programmer Response:** None.

**GIM535s** *nn type* ENTRIES CONVERTED FROM *dstype lib*

**Explanation:**

- *nn* - the number of entries SMP/E converted
- *type* - ASSEM, DLIB, LMOD, MAC, MCS, MOD, SRC, SYSMOD, or SYSTEM
- *dstype* - CDS, ACDS, PTS, CRQ, or ACRQ
- *lib* - the name of the DD statement specified in the CDS, ACDS, PTS, CRQ, or ACRQ CONVERT operand.

Shows the number of entries that SMP/E has converted.

## GIM5XX

**System Action:** CONVERT processing continues normally.

**Programmer Response:** None.

### GIM536s NO *type* ENTRIES FOUND IN *dstype lib*

**Explanation:**

- *type* - ASSEM, DLIB, LMOD, MAC, MCS, MOD, SRC, SYSMOD, or SYSTEM
- *dstype* - CDS, ACDS, PTS, CRQ, or ACRQ
- *lib* - the name of the DD statement specified in the CDS, ACDS, PTS, CRQ, or ACRQ CONVERT operand.

SMP/E converts the SMP4 data set that has no entries of the type specified.

**System Action:** CONVERT processing continues normally.

**Programmer Response:** None.

### GIM537s CONVERT PROCESSING {SUCCESSFUL | FAILED } FOR *dstype lib* TO {GLOBAL | TARGET | DLIB} ZONE *zonename*

**Explanation:**

- *dstype* - CDS, ACDS, PTS, CRQ, or ACRQ.
- *lib* - the name of the DD statement specified in the CDS, ACDS, PTS, CRQ, or ACRQ CONVERT operand.
- *zonename* - the name of the zone specified in the previous SET command. Note that if *zonetype* is GLOBAL, *zonename* will not be in the message.

This message will tell you if the conversion was successful or not.

**System Action:** One of the following can occur:

- If CONVERT was successful, processing continues with the next command.
- If CONVERT failed, the return code determines if SMP/E will run the next command.

**Programmer Response:** Do one of the following:

- If conversion was successful, do nothing.
- If conversion failed, look at the previous error messages to find the cause of the failure. Fix the error and rerun the job from the point of failure.

**GIM538s** *type name* ONLY PARTIALLY CONVERTED**Explanation:**

- *type* - ASSEM, DLIB, MAC, MCS, MOD, SRC, SYSMOD, or SYSTEM
- *name* - the entry name.

SMP/E found an error when it was converting entry *name*. A previous message identifies the error.

**System Action:** Conversion continues with the next entry from the specified data set.

**Programmer Response:** Do the following:

1. Look at the previous messages to find the cause of the error.
2. Do one of the following:
  - Rerun CONVERT to correct the error.
  - Use SMP4 to list the SMP4 data set entry, then use SMP/E to list the SMP/E entry. Use UCLIN to fix the one entry that is in error.

**GIM539s** ZONE TYPE FOR {NOAPPLY | NOACCEPT} NOT A {TARGET | DLIB} ZONE**Explanation:**

Zone types are as follows:

- The type for LIST NOAPPLY command must be a target zone.
- The type for LIST NOACCEPT command must be a DLIB zone.

**System Action:** LIST command processing stops.

**Programmer Response:** Fix the name of NOAPPLY/NOACCEPT zone and resubmit the job.

**GIM540s** ERROR OR USER HOLD INVALID WITHIN A SYSMOD

**Explanation:** When SMP/E was receiving a SYSMOD, SMP/E encountered a ++HOLD MCS that specified a HOLD category of ERROR or USER. The only HOLD category that is valid within a SYSMOD is the HOLD category of SYSTEM.

**System Action:** SMP/E did not receive the SYSMOD.

**Programmer Response:** Fix the SYSMOD construction and RECEIVE again.

## GIM5XX

**GIM541s** *lib* DYNAMIC ALLOCATION SUCCESSFUL - DDDEF = *name*, DSN = *dsname*,  
VOLUME = *volume*, UNIT = *unit*, DISP = *status*

**Explanation:**

- *lib* - the name of the DD statement that SMP/E allocated.
- *name* - the name of the DDDEF entry that SMP/E used to allocate the DD statement.
- *dsname* - the name of the data set that SMP/E allocated.
- *volume* - the volume that the data set resides on.
- *unit* - the unit that the data set resides on. Note that this will only appear if you specified the UNIT type in the DDDEF entry.
- *status* - the status of the data set (OLD, SHR, or MOD).

This message shows that SMP/E successfully allocated the specified DD statement.

**System Action:** Processing continues.

**Programmer Response:** None.

**GIM542s** *lib* DYNAMIC ALLOCATION SUCCESSFUL - CONCATENATION = *libs*

**Explanation:**

- *lib* - the name of the DD statement that SMP/E allocated
- *libs* - the names of the DD statements that SMP/E concatenated as part of the *lib* allocation.

This message shows that SMP/E successfully allocated the specified DD statement.

**System Action:** Processing continues.

**Programmer Response:** None.

**GIM543s** *lib* DYNAMIC UNALLOCATION SUCCESSFUL

**Explanation:**

- *lib* - the name of the DD statement that SMP/E freed.

This message shows that SMP/E successfully freed the specified DD statement.

**System Action:** Processing continues.

**Programmer Response:** None.

**GIM545s** *lib* DYNAMIC ALLOCATION FAILED - NO *name* DDDEF ENTRY FOUND IN {GLOBAL | TARGET | DLIB} ZONE *zonename*

**Explanation:**

- *lib* - the name of the DD statement that SMP/E was trying to allocate.  
Note that if *lib* is in the form SMP#####, the DD statement is part of a concatenation. Message GIM546 shows the DD name for the concatenation.
- *name* - the name of the DDDEF entry that SMP/E used to allocate the DD statement
- *zonename* - the name of the target or DLIB zone that should have had the DDDEF entry. If *zonetype* is GLOBAL, SMP/E does not list *zonename* in the message.

SMP/E was trying to allocate the DD statement specified by *lib*, but SMP/E could not find DDDEF entry *name* in the specified zone.

**System Action:** System action depends on the function that SMP/E is running and the DD statement that SMP/E could not allocate. A subsequent message identifies the system action.

**Programmer Response:** Do one of the following:

- Add the required DDDEF entry to zone *zonename*.
- Include the DD statement in the JCL that the system used to run SMP/E.

**GIM546s** *lib* DYNAMIC ALLOCATION FAILED - UNABLE TO ALLOCATE *name* FOR *lib* CONCATENATION

**Explanation:**

- *lib* - the name of the DD statement that SMP/E was trying to allocate
- *name* - the name of the DDDEF entry that SMP/E was to use as part of the concatenation.

SMP/E was trying to allocate the DD statement *lib* that specified a concatenation string. However, SMP/E could not allocate DD statement *name*. The reason that SMP/E could not allocate *name* is in message GIM545s.

**System Action:** System action depends on the function that SMP/E is running and the DD statement that SMP/E could not allocate. A subsequent message identifies the system action.

**Programmer Response:** Fix the error in the allocation of the *name* DD statement and rerun the SMP/E job.

**GIM547s** *lib* DYNAMIC ALLOCATION FAILED - DYNAMIC ALLOCATION FAILED - *message***Explanation:**

- *lib* - name of the DD statement that SMP/E was trying to allocate.
- *message* - the text of the error message from the DAIRFAIL system service that shows the reason for the allocation failure. If the message text is greater than 200 characters, SMP/E prints message GIM547s again with the second part of the error text.

When SMP/E was trying to allocate the DD statement *lib*, an error occurred in SVC99 (dynamic allocation SVC).

**System Action:** System action depends on the function that SMP/E is running and the DD statement that SMP/E could not allocate. A subsequent message identifies the system action.

**Programmer Response:** Fix the error in the allocation of the *lib* DD statement and rerun the SMP/E job.

**GIM548s** *lib* DYNAMIC ALLOCATION NOT ATTEMPTED - A PRIOR ALLOCATION ATTEMPT WITHIN THIS ZONE WAS NOT SUCCESSFUL**Explanation:**

- *lib* - the name of the DD statement that SMP/E was trying to allocate.

Although SMP/E needed *lib*, it did not try to dynamically allocate the DD statement because an attempt since the last SET command failed.

During processing, SMP/E keeps track of all dynamic allocation attempts from one SET command to the next SET command. When an allocation fails, SMP/E will not try that allocation again. The next SET command, however, resets the allocation history and lets SMP/E try the allocation again.

**System Action:** System action depends on the function that SMP/E is running and the DD statement that SMP/E could not allocate. A subsequent message identifies the system action.

**Programmer Response:** Do one of the following:

- Add the required DDDEF entry to the specified zone.
- Include the DD statement in the JCL that runs SMP/E.
- If you are running under TSO, do the following:
  1. Allocate the DD statement.
  2. Issue another SMP/E SET command.
  3. Rerun the failing SMP/E command.

**GIM549s** *lib* ALLOCATED BY USER - *data***Explanation:**

- *lib* - name of the DD statement that SMP/E was trying to allocate
- *data* - some of the information that SMP/E found about the data set described by *lib*.

You allocated the DD statement *lib*, which SMP/E requires for processing, rather than SMP/E dynamically allocating it.

- For background jobs, SMP/E found a DD statement in the JCL; this JCL invokes SMP.
- For foreground processing, SMP/E found a file that was already allocated.

The value of *data* depends on the DD statements as follows:

- For DD statements that describe inline input files or SYSOUT files, *data* is "SYSIN/SYSOUT".
- For DD statements that describe files other than SYSIN or SYSOUT files, *data* will be as follows:

---

```
DA(dsname) VOLUME(volser) SPACE(pri,sec) DIR(nn) alloc status disp
```

---

**Where:**

- *dsname* is the fully qualified data set name.
- *volser* is the volume serial number.
- *pri* is the primary space allocation. This field is present only for temporary data sets.
- *sec* is the secondary space allocation. This field is present only for temporary data sets.
- *nn* is the number of directory blocks that SMP/E allocated. This field is present only for temporary partitioned data sets.
- *alloc* is the units of allocation: TRACKS, CYL, or BLOCK(blklen).
- *status* is the status of the data set: NEW, MOD, or OLD.
- *disp* is the value of the data set: either DELETE or KEEP.

**System Action:** None.

**Programmer Response:** None. SMP/E writes this message to the SMPLOG data set for record keeping purposes only. The message lets you use SMPLOG to verify that SMP/E directed updates to the correct data sets and volumes at any point in time.



# GIM5XX

## GIM551s ALL RECORDS SUCCESSFULLY DELETED FROM ZONE *zonename*

**Explanation:** ZONEDELETE processing completed. SMP/E deleted the records that were in zone *zonename* from the SMPCSI data set.

**System Action:** Processing continues with the next input in SMP\_CNTL.

**Programmer Response:** This is an informational message; do nothing.

## GIM552s ZONEINDEX {SUCCESSFULLY | NOT} DELETED FOR ZONE [UNABLE TO ACQUIRE GLOBAL ZONE]

**Explanation:**

- *zonename* - zone name.

Shows if SMP/E deleted the ZONEINDEX record.

**System Action:** Command processing continues.

**Programmer Response:** Do one of the following:

- If SMP/E deleted the ZONEINDEX; do nothing.
- If the GLOBAL zone was unavailable, use UCLIN to delete the ZONEINDEX.

## GIM553s UNABLE TO OPEN PARMLIB TO OBTAIN OPCODE LIST - PROCESSING CONTINUES IN DEFAULT MODE

**Explanation:** During JCLIN processing, SMP/E could not open the PARMLIB DD card to read the members that have a list of valid OPCODEs. SMP/E uses this list of OPCODEs when it is scanning the assembler steps to distinguish valid OPCODEs from macro invocations.

Since SMP/E could not get the OPCODE lists from PARMLIB, SMP/E continues processing, using the following method to determine an OPCODE or a macro:

- If the character string is 6 characters or greater, SMP/E treats the string as a macro; SMP/E creates a macro entry.
- If the character string is 5 characters or less, SMP/E treats the string as an OPCODE; SMP/E does not create a macro entry.

For each character string, SMP/E issues message GIM555I, showing the following:

1. The string that SMP/E found.
2. The name of the assembly where SMP/E found the string.
3. An indication of how SMP/E treated the character string.

**System Action:** JCLIN input processing continues.

**Programmer Response:** Do one of the following:

- Rerun the job with a PARMLIB DD statement.
- Find from message GIM555I if the default method caused any errors; if so, use UCLIN to add the required information to the CSI.

The following is a sample UCLIN to add a macro entry to the CSI and to connect it to the proper assembler entry:

---

```
UCLIN.
ADD MAC(XXXX) GENASM(name).
ENDUCL.
```

---

Where:

- *xxxx* is the name of the macro from GIM555I.
- *name* is the name of the assembler entry from GIM555I.

**GIM554s PARMLIB MEMBER *name1* NOT FOUND - *message text***

**Explanation:**

- *name1* - the name of the PARMLIB member that has the list of valid OPCODEs for JCLIN processing
- *message text* - one of the following:
  - PROCESSING CONTINUES IN DEFAULT MODE
  - PROCESSING CONTINUES USING ONLY THE DATA IN PARMLIB MEMBER *name2*

During JCLIN processing, SMP/E could not find one or more of the two PARMLIB members that it uses to distinguish OPCODEs from macros. SMP/E uses these PARMLIB members when it is scanning the assembly steps in the JCLIN input.

- When *message text* equals “PROCESSING CONTINUES IN DEFAULT MODE”, one of the following conditions occurred:
  - You did not specify the OPCODE(*name2*) operand on the JCLIN command or the ++JCLIN MCS, and SMP/E could not find the SMP/E provided member (GIMOPCDE).
  - You specified the OPCODE operand, and SMP/E could not find either member.

SMP/E uses the following rules to determine an OPCODE from a macro:

- If the character string is 6 characters or greater, SMP/E treats the string as a macro; SMP/E creates a macro entry.
- If the character string is 5 characters or less, SMP/E treats the string as an OPCODE; SMP/E does not create a macro entry.

For each character string, SMP/E issues message GIM555I, showing the following:

1. The string that SMP/E found.
  2. The name of the assembly where SMP/E found the string.
  3. An indication of how SMP/E treated the character string.
- When *message text* equals “PROCESSING CONTINUES USING ONLY THE DATA IN PARMLIB MEMBER *name2*”, the following condition occurred:
    - SMP/E found only one of the two PARMLIB members. SMP/E will use only the data in the *name2* PARMLIB member for determining macros from OPCODEs.

**System Action:** JCLIN input processing continues.

**Programmer Response:** Do one of the following:

- Rerun the job with a PARMLIB member present.
- Find out from message GIM555I if the default method caused any errors; if so, use UCLIN to add the required information to the CSI.

The following is a sample UCLIN to add a macro entry to the CSI and to connect it to the proper assembler entry:

---

```
UCLIN.
ADD MAC(name1) GENASM(name2).
ENDUCL.
```

---

Where:

- *name1* is the name of the macro from GIM555I.
- *name2* is the name of the assembler entry from GIM555I.

**GIM555s** *xxxx* FOUND IN ASSEMBLY *name1* INPUT IS CONSIDERED *message text*

**Explanation:**

- *xxxx* - the character string that SMP/E found
- *name1* - the assembly where SMP/E found the characters string
- *message text* - one of the following:
  - A MACRO BECAUSE IT IS GREATER THAN 5 CHARACTERS
  - AN OPCODE BECAUSE IT IS LESS THAN 6 CHARACTERS
  - A MACRO BECAUSE IT WAS NOT FOUND IN PARMLIB MEMBER *name2*

During JCLIN processing, SMP/E found the character string *xxxx* in assembly *name1*. SMP/E determined that it should treat the character string as either an OPCODE or macro; *message text* explains why.

**System Action:** JCLIN input processing continues.

**Programmer Response:** Do one of the following:

- Fix the error as shown in GIM553 or GIM554 and rerun the job.
- Find out from message GIM555I if the default method caused any errors; if so, use UCLIN to add the required information to the CSI.

The following is a sample UCLIN to add a macro entry to the CSI and to connect it to the proper assembler entry:

---

```
UCLIN.
ADD MAC(name) GENASM(value).
ENDUCL.
```

---

**GIM556s** UNABLE TO ALLOCATE *lib1* DD STATEMENT - CONTINUING USING *lib2* RATHER THAN *lib1*

**Explanation:**

- *lib1* - name of DD statement
- *lib2* - name of the DD statement that SMP/E will use.

During the invocation of a utility program, the UTILITY entry specified that DD statement *lib1* was required. However, SMP/E could not find *lib1* in the JCL that runs or as a DDDEF entry in the current zone. Rather than stopping the utility, SMP/E used the *lib2* DD statement, which it allocated.

SMP/E does this normally for the optional SYSPRINT DD name that you may specify in the UTILITY entry.

**System Action:** Processing continues.

## GIM5XX

**Programmer Response:** You do not need to do anything for the current SMP/E run. However, fix the problem for subsequent executions by doing one of the following:

- Add a DD card to the SMP/E cataloged procedure.
- Add a DDDEF entry to the applicable zones.

### GIM557s ERROR ENCOUNTERED {READING | STORING} *type name* - CONVERT PROCESSING {CONTINUES | TERMINATES}

**Explanation:**

- *type* - entry type (MOD, MAC, SYSMOD, etc.)
- *name* - entry name.

During CONVERT, SMP/E found one of the following:

- An error reading an entry from the SMP4 data sets.
- An error while storing the converted entry on the SMP/E data set.

Depending on the severity of the error, SMP/E will do one of the following:

- SMP/E will not convert that one entry.
- SMP/E will stop the total CONVERT command.

**System Action:** If *status* is "CONTINUES", the CONVERT command will continue with the next entry; otherwise, processing will stop.

**Programmer Response:** Do one of the following according to the condition:

- If CONVERT processing stopped, look at the previous messages to find the error and fix the problem. Rerun the CONVERT.
- If CONVERT continues and the number of these messages is excessive, fix the problems shown in previous messages and rerun the CONVERT.
- If CONVERT continues and only a few of these messages appear, use the SMP4 UNLOAD command to create UCLIN. Use that UCLIN to build the entry in the SMP/E data set.

### GIM558s ZONE {NAME | TYPE} *value* FROM *command* COMMAND CONFLICTS WITH ZONE {NAME | TYPE} FROM PROCESSING

**Explanation:**

- *value* - zone name if *zone* is NAME, TARGET or DLIB if *zone* is TYPE
- *command* - command entered.

**System Action:** Processing stops for this command.

**Programmer Response:** Check that the type and name combinations that GLOBAL ZONEINDEX defines are the same values as those you specified on the command. Fix the error and rerun the job.

**GIM559s THE CONCAT KEYWORD IS MUTUALLY EXCLUSIVE WITH ANY OTHER DDDEF KEYWORD**

**Explanation:** Do not use the CONCAT keyword with any other DDDEF keywords for UCLIN processing.

**System Action:** SMP/E sets the return code to 8. Statement processing stops, and processing continues with the next statement.

**Programmer Response:** Fix the error and resubmit the statement.

**GIM560s THE CONCAT KEYWORD CONTAINS MORE THAN 16 DDDEF NAMES**

**Explanation:** The CONCAT keyword cannot have more than 16 DDDEF names as input.

**System Action:** SMP/E sets the return code to 8. Statement processing stops, and processing continues with the next statement.

**Programmer Response:** Fix the error and rerun the statement.

**GIM561s THE ZONE NAME USED FOR THE PURGE OPERAND MUST BE A DLIB TYPE ZONE - *zonename* IS A TARGET TYPE**

**Explanation:** The PURGE operand value on the REJECT command must be the name of a DLIB type zone. However, the *zonename* is not a DLIB type zone.

**System Action:** Command processing stops.

**Programmer Response:** Specify a valid DLIB type zone name for the PURGE option of the REJECT command and rerun the job.

**GIM562s *jobname* REQUIRES BUSY SMP/E CSI DATA SET '*dsname*', REPLY 'WAIT' OR 'END'**

**Explanation:**

- *jobname* - name of the task that needs SMP/E CSI data set
- *dsname* - data set name of the SMP/E CSI data set.

Task *jobname* is trying to open SMP/E CSI data set *dsname* for potential update operations, but another task already has the SMP/E CSI data set *dsname* open for potential update operations.

Only one task can update an SMP/E CSI data set at a time. The current task must wait for the resource or end. SMP/E issues this message as a WTOR, and SMP/E's action depends on your reply.

## GIM5XX

**System Action:** Your reply affects SMP/E the following ways:

- If you reply "WAIT", SMP/E waits until the resource is available.
- If you reply "END", SMP/E does not wait for the resource, and SMP/E stops the process requiring the data set.

**Programmer Response:** Do one of the following:

- If you replied "WAIT", do nothing, since eventually the resource will become available.
- If you replied "END", resubmit the portion of the SMP/E job that stopped because the resource was not available.

### GIM563s ZONE TYPE FOR THE RELATED ZONE *zonename* MUST BE *zonetype* FOR *command* PROCESSING

**Explanation:**

- *zonename* - name of the related zone for this function
- *zonetype* - type of the related zone from the GLOBAL zone ZONEINDEX
- *command* - name of the SMP/E command that you are requesting.

**System Action:** Function processing stops with a return code of 12.

**Programmer Response:** Check that the related zone type is the proper type for the requested SMP/E command and do one of the following:

- If *command* is ACCEPT, the related zone type must be target.
- If *command* is RESTORE, the related type must be DLIB.

Fix the inconsistent data, and rerun the SMP/E command.

### GIM564s SYSMOD *sysmod* HAS ALREADY BEEN INPUT AS A {SELECT | EXCLUDE | TO} VALUE

**Explanation:**

- *sysmod* - SYSMOD-ID.

**System Action:** Function processing stops with a return code of 12.

**Programmer Response:**

Fix the command so that the SYSMOD named is only entered one time.

- Do not enter the SYSMOD more than once in a SELECT list or EXCLUDE list, or both a SELECT and an EXCLUDE list.
- Do not enter the SYSMOD more than once on a single ++ASSIGN MCS statement.

Then rerun the SMP/E command.

**GIM565s** THE *type* SUB-ENTRY DOES NOT EXIST - ADD ASSUMED**Explanation:**

- *type* - subentry type.

SMP/E could not find the subentry specified on the UCLIN REP command. It therefore assumes you wanted to add the subentry.

**System Action:** SMP/E adds the subentry with a return code of 4.

**Programmer Response:** None.

**GIM566s** *name* CAN NOT BE ADDED - ENTRY ALREADY EXISTS**Explanation:**

- *name* - entry name.

The entry already exists, and SMP/E cannot add it.

**System Action:** Processing continues with the next statement, with a return code of 8.

**Programmer Response:** If the entry name is in error, fix it and rerun the statement.

**GIM567s** ERROR OCCURRED IN SMPHOLD DATA FILE MEMBER *mcs* (*sysmod*)**Explanation:**

- *mcs* - ++HOLD, ++RELEASE, ++NULL, or UNKNOWN
- *sysmod* - SYSMOD-ID or UNKNOWN.

Message GIM203I immediately precedes this message and shows that the line immediately before GIM203I has a syntax error.

**System Action:** Processing stops for the SMPHOLD FILE member that is in error.

**Programmer Response:** Check the format of the keyword on the specified command. Fix the syntax error and rerun the job.

**GIM568s** *type* INPUT CONTAINS DUPLICATE VALUES**Explanation:**

- *type* - subentry type.

Subentry type *type* has duplicate values on the UCLIN statement.

**System Action:** Processing continues with the next statement with a return code of 8.

**Programmer Response:** Remove the duplicate values and rerun the statement.



## GIM5XX

### GIM569s ZONEINDEX ERROR DETECTED - THE REMAINING UCLIN STATEMENTS WILL BE FLUSHED UNTIL ENDUCL IS PROCESSED

**Explanation:** If SMP/E finds an error while it is processing a ZONEINDEX keyword, SMP/E considers it a critical error that can cause unpredictable results.

**System Action:** SMP/E flushes the remaining UCLIN statements until it processes the ENDUCL statement. Processing continues in a syntax checking mode with a return code of 12.

**Programmer Response:** Fix the statement in error and rerun the statements that SMP/E did not process.

### GIM570s THE PARM SUB-ENTRY EXCEEDS THE 100 CHARACTER MAXIMUM

**Explanation:** The maximum number of characters that you can specify in the PARM keyword is 100.

**System Action:** Processing for this statement stops and continues with the next statement.

**Programmer Response:** Fix the statement in error and rerun the statements that SMP/E did not process.

### GIM571s CONCAT IS INVALID WHEN OTHER DDDEF SUB-ENTRIES ALREADY EXIST IN THE ENTRY

**Explanation:** When you specify the CONCAT keyword, there can be no other subentries present in the DDDEF entry that you want to update.

**System Action:** Processing for this statement stops and continues with the next statement.

**Programmer Response:** Fix the statement in error and rerun the statements that SMP/E did not process.

### GIM572s CONCAT IS PRESENT IN ENTRY - CAN NOT ADD DDDEF SUB-ENTRIES

**Explanation:** When you specify DDDEF keywords other than CONCAT for input, there cannot be a CONCAT subentry present in the entry.

**System Action:** Processing for this statement stops and continues with the next statement.

**Programmer Response:** Fix the statement in error and rerun the statements that SMP/E did not process.

**GIM573s** *type* UCLIN STATEMENT MUST CONTAIN AT LEAST ONE KEYWORD**Explanation:**

- *type* - entry type.

The specified statement needs a minimum amount of data before UCLIN can consider it for processing. Any valid keyword that you specify for this entry type will make this UCLIN statement valid.

**System Action:** Processing continues with the next statement with a return code of 8.

**Programmer Response:** Add a valid keyword and rerun the statement.

**GIM574s** SMPPTS IS NOT A VALID DDNAME FOR THE PTS OPERAND ON THE CONVERT COMMAND

**Explanation:** The ddname SMPPTS is reserved for the SMP/E data set. You must assign a different ddname for the SMP4 data set.

**System Action:** CONVERT processing stops.

**Programmer Response:** Do the following:

- Change the ddname for the SMP4 data set. (SMPPTS must be the name of the SMP/E data set.)
- Check that the JCL also shows this change.
- Resubmit the CONVERT command for processing.

**GIM575s** LIST BACKUP CAN ONLY BE CODED ALONE OR WITH THE SYSMOD KEYWORD

**Explanation:** SMP/E requires that you list the BACKUP data set (SMPSCDS) separately.

**System Action:** SMP/E ignores the LIST statement and processing continues with next statement.

**Programmer Response:** Fix the LIST statement.

**GIM576s** *ddname1* DYNAMIC ALLOCATION FAILED - *name* IS A CONCAT DDDEF ENTRY SPECIFIED AS PART OF *ddname2* CONCATENATION**Explanation:**

- *ddname1* - ddname of first data set
- *name* - DDDEF entry name
- *ddname2* - ddname of second data set.

When SMP/E was processing the dynamic allocation for the ddname that you specified, SMP/E tried to concatenate a concatenation DDDEF entry.

## GIM5XX

**System Action:** DDDEF processing stops.

**Programmer Response:** Fix the DDDEF entry for the ddname so that it only concatenates real data set DDDEF entries, *not* other concatenation DDDEF entries.

**GIM577s I/O ERROR OCCURRED PERFORMING A *action* OPERATION - DDNAME = *lib*, MEMBER = *name*, APPROXIMATE TTR = *ttr***

**Explanation:**

- *action* - the type of operation that SMP/E was doing when it detected the I/O error (READ, WRITE, STOW, BLDL, OPEN, CLOSE)
- *lib* - the name of the DD statement where SMP/E detected the error
- *name* - the partitioned data set member name where SMP/E detected the error.

*Note: If the data set was sequential, or if the data set was operating on the directory portion of a partitioned data set, this part of the message (“,MEMBER = name”) does not appear.*

- *ttr* - the approximate TTR where SMP/E detected the error
  - For READ operations, the TTR is from the first record for the indicated member.
  - For WRITE operations, the TTR is from the DCB where SMP/E did the last write.

*Note: If the data set was sequential, or if the data set was operating on the directory portion of a partitioned data set, this part of the message (“,APPROXIMATE TTR = ttr”) does not appear.*

SMP/E produced this message when it met an I/O error condition for a non-VSAM data set. SMP/E may have detected the I/O error condition with one of the following:

- The return code from a STOW or BLDL operation
- The SYNAD exit for READ and WRITE operations.

**System Action:** SMP/E’s action depends on the following:

- The data set where SMP/E detected the I/O error condition
- The processing that SMP/E had already done.

Subsequent messages show SMP/E’s action. The action may range from stopping the processing of a SYSMOD to stopping SMP/E.

**Programmer Response:** Find the cause of the error from the message. Fix the error and rerun the SMP/E job.

**GIM578s ZONE DEFINITION CANNOT BE UNLOADED****Explanation:**

SMP/E cannot unload the zone definition records.

**System Action:** SMP/E ignores the UNLOAD statement and processing continues with next statement.

**Programmer Response:** Remove the TZONE or DZONE from the UNLOAD command and resubmit the statement.

**GIM579s VSAM DATA SET '*dsname*' IS OUT OF SPACE****Explanation:**

- *dsname* - VSAM CSI data set name.

The VSAM CSI *dsname* is full.

**System Action:** Processing stops for the SMP/E command that is causing SMP/E to make updates to the VSAM CSI *dsname*.

**Programmer Response:** Do the following:

1. Use IDCAMS to EXPORT the data set.
2. DELETE the data set.
3. DEFINE a new data set with the same name and more space.
4. IMPORT the information that is in the old data set and saved with EXPORT.

**GIM580s ZONE OPEN ERROR - *error type*****Explanation:**

- *error type* - one of the following values:
  1. DYNAMIC ALLOCATION FAILED FOR *zonename* ZONE DATA SET *name*
  2. SMP/E CSI DATA SET '*name*' IS BUSY
    - *zonename* - name of zone trying to be opened
    - *name* - SMP/E CSI data set name.

## GIM5XX

If the error type is number 1, allocation failed for the indicated VSAM CSI data set. This may happen due to one of the following:

- The data set name is incorrect in the ZONEINDEX record for the indicated zone in the GLOBAL zone definition.
- The volume that has the indicated VSAM CSI data set could not be mounted.
- The catalog for the indicated VSAM CSI data set was not available.
- The indicated VSAM CSI data set does not exist.

If the error type is number 2, another task is currently using the the indicated VSAM CSI data set, and the operator replied 'END' to message GIM562.

**System Action:** Open processing for the indicated zone and VSAM CSI data set stops.

**Programmer Response:** Do one of the following:

- If the error type is 1, fix the error and resubmit the job.
- If the error type is 2, do one of the following:
  - Resubmit the job when the indicated VSAM CSI data set is not in use.
  - Ask the operator to reply 'WAIT' to message GIM562.

**GIM581s** *program* **TO BE USED FOR THE** *utility* **UTILITY IS NOT VALID FOR SMP/E USAGE AS DEFINED BY GIMUTTBL**

**Explanation:**

- *program* - program name
- *utility* - utility type.

**System Action:** Processing stops and continues with the next command.

**Programmer Response:** If the program is to be a valid program for the utility function, add the program to the GIMUTTBL with the GIMDFUT macro. Refer to Appendix A in *Installing SMP/E*.

**GIM582s** **ZONE** *zonename* **NOT FOUND IN ZONE INDEX**

**Explanation:**

- *zonename* - name of the zone you want to rename.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Do one of the following:

- Check that you have not misspelled the name of the zone.
- If the spelling is correct, check that a GLOBAL zone ZONEINDEX entry exists for the zone.

**GIM583s ZONE *zonename* FOUND IN ZONE INDEX****Explanation:**

- *zonename* - new name of the zone you want to rename.

SMP/E found a GLOBAL zone ZONEINDEX entry for the new zone name. However, the new zone name should not be in the GLOBAL zone ZONEINDEX entries.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Do one of the following:

- Check that you spelled the name of the new zone name correctly and rerun the ZONERENAME operation.
- If the spelling is right, check that the new zone name was not defined by someone else.
  - If someone else is currently using the new zone name, choose a different zone name for the ZONERENAME operation and rerun the ZONERENAME operation.
  - If the zone name is not currently in use, use UCLIN to delete the GLOBAL zone ZONEINDEX entry for the new zone name and rerun the ZONERENAME operation.

**GIM584s ZONE *zonename* FOUND IN NEWDATASET****Explanation:**

- *zonename* - new name of the zone to be renamed.

SMP/E found an internal use record, known as the zone encode record, in the data set that has the zone you want to rename. This record shows that the new zone name is already in use. SMP/E found remnants of a zone with the same name as the new zone name in the new data set.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** To get rid of the remnants, do the following:

1. Use UCLIN to add a GLOBAL zone ZONEINDEX entry for the new zone name that uses the new data set value for the data set.
2. Do a ZONEDELETE for the new zone name.

*Note:* Since you remove all data for the zone from the subject data set when you do these steps, be absolutely sure SMP/E does not need the data.

3. After you use ZONEDELETE, rerun the ZONERENAME command.

## GIM5XX

**GIM585s** ZONE *oldzonename* CANNOT BE RENAMED TO *newzonename*

**Explanation:**

- *oldzonename* - name of the zone you want to rename
- *newzonename* - new name of the zone you want to rename.

ZONERENAME could not rename the zone. Previously issued messages show why the failure occurred.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Correct the errors shown by previous error messages and rerun the ZONERENAME command.

**GIM586s** ZONE *oldzonename* RENAMED TO *newzonename*

**Explanation:**

- *oldzonename* - name of the zone you want to rename
- *newzonename* - new name of the zone you want to rename.

ZONERENAME has successfully renamed the zone.

**System Action:** ZONERENAME processing completes. Processing continues with the next command.

**Programmer Response:** Do not respond; this is an informational message.

**GIM587s** ZONE *oldzonename* RENAMED TO *newzonename* IN NEWDATASET

**Explanation:**

- *oldzonename* - name of the zone you want to rename
- *newzonename* - new name of the zone you want to rename.

ZONERENAME has successfully renamed the zone in a NEWDATASET (a copy of another CSI data set). ZONERENAME has also connected the renamed zone to the master CSI with a GLOBAL zone ZONEINDEX entry.

**System Action:** ZONERENAME processing completes. Processing continues with the next command.

**Programmer Response:** Do not respond; this is an informational message.

**GIM588s ZONE *zonename* FOUND IN ZONE INDEX BUT HAS NOT BEEN DEFINED****Explanation:**

- *zonename* - name of the zone you want to rename.

ZONERENAME found a GLOBAL zone ZONEINDEX entry for the old zone name. However, SMP/E searched through the CSI data set and did *not* find an internal processing record, known as the zone encode record, for the old zone name.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Check that the CSI data set has the zone you want to rename. Check the following:

1. The data set you defined on the DD statement with the new zone name as the DD statement name
2. The data set for the old zone, if you did not provide a DD statement and you specified the SAMEDATASET operand on the ZONERENAME command
3. The data set you specified with the NEWDATASET operand on the ZONERENAME command.

**GIM590s ZONERENAME CANNOT BE DONE-INPUT AND OUTPUT ZONE NAMES ARE EQUAL- *zonename*****Explanation:**

- *zonename* - name of the zone you want to rename.

The ZONERENAME command found that the name of the zone to be renamed is the same as the new name you specified for the zone.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Check that the name of the zone you are renaming is not the new name of the zone and resubmit the job.

**GIM591s KEYWORDS *keyword1* AND *keyword2* ARE MUTUALLY EXCLUSIVE****Explanation:**

- *keyword1* - keyword
- *keyword2* - keyword.

The operands for the ZONERENAME command are mutually exclusive; do not specify them together.



## GIM5XX

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Do not use the mutually exclusive operands together. Correct the ZONERENAME command and resubmit the job.

### GIM592s THE ZONE NAME OF GLOBAL IS RESTRICTED FOR SMP USAGE

**Explanation:** You specified one of the following:

- The zone you want to rename as GLOBAL
- The new zone name on the ZONERENAME command as GLOBAL.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** GLOBAL is a reserved zone name for SMP/E usage. Choose a different zone name and resubmit the job.

### GIM593s TOTYPE OPTION SPECIFIED. OLDZONENAME MUST BE A DLIB TYPE {ZONEINDEX | DEFINITION}

**Explanation:** You used the TOTYPE(TARGET) operand on the ZONERENAME command. This means that the zone you want to rename is currently a distribution zone. However, ZONERENAME processing found one of the following:

- The GLOBAL zone ZONEINDEX entry for the zone you want to rename is target type.
- The zone definition for the zone you want to rename is target type.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Do one of the following:

- If the zone you want to rename is a distribution zone, fix the GLOBAL zone ZONEINDEX entry and the zone definition for the zone to show that the zone is distribution type.
- If the zone you want to rename is a target zone, remove the TOTYPE(TARGET) operand.

After you fix the error, resubmit the job.

**GIM594s NEWDATASET CSI NAME VALUE CAN NOT BE THE SAME AS OLDZONENAME  
CSI NAME VALUE**

**Explanation:** The value of the NEWDATASET operand specified on the ZONERENAME command is equal to the value of the data set name in the GLOBAL zone ZONEINDEX entry for the zone you want to rename.

**System Action:** ZONERENAME processing stops. Processing continues with the next command.

**Programmer Response:** Do one of the following:

- If the zone you want to rename is to be renamed in its original data set, use the SAMEDATASET operand, *not* the NEWDATASET operand.
- If the zone you want to rename exists in a copy of the CSI data set, check that the data set name of the new data set is different than the original data set name for the zone.

After you fix the error, resubmit the job.

**GIM595s ZONE DEFINITION {OPTIONS | RELATED | TYPE} RECORD VALUE *oldvalue*  
{ADDED | CHANGED} TO *newvalue***

**Explanation:**

- *oldvalue* - value added to or changed in rename zone's definition
- *newvalue* - the value after it has been added to or changed.

**System Action:** None.

**Programmer Response:** None.

**GIM596s {ENQ | DEQ} FOR {EXCLUSIVE | SHARED} USE OF *dsname* FOR *command*  
{INITIATED | WAS SUCCESSFUL | WAS NOT SUCCESSFUL}**

**Explanation:**

- *data* - name of the CSI data set or SMPPTS data set
- *command* - SMP/E command.

SMP/E writes this message to the SMPLOG data set each time it enqueues or dequeues a CSI data set.

**System Action:**

When the enqueue is not successful, the current phase of the command that SMP/E is executing stops. Command processing may stop.

**Programmer Response:** When the enqueue is not successful, check that the required data sets are not being used by another user.

## GIM5XX

### GIM597s GLOBAL ZONE UPDATES FROM *zonename* ZONE WERE SUCCESSFUL

**Explanation:**

- *zonename* - name of the zone that has updates.

The command updated the GLOBAL zone.

**System Action:** None.

**Programmer Response:** None.

### GIM598s GLOBAL ZONE UPDATES FROM *zonename* ZONE NOT DONE

**Explanation:**

- *zonename* - name of the zone that has updates.

Command processing finished, but the GLOBAL zone was not available for update. SMP/E saved the updates in the target or DLIB zone shown by the *zonename* value in the message.

**System Action:** None.

**Programmer Response:** The next time a SET, APPLY, ACCEPT or RESTORE command accesses the *zonename* zone, SMP/E will try again to update the GLOBAL zone.

If the next command will not run successfully until SMP/E updates the GLOBAL zone, do the following:

1. Run SMP/E.
2. SET the boundary to the *zonename* zone.

### GIM599s JOB *jobname* HAS BEEN WAITING FOR *nn* MINUTES FOR DATA SET *dsname*

**Explanation:**

- *jobname* - name of the task that is waiting
- *nn* - time the task has been waiting, a multiple of 30 minutes
- *dsname* - name of the CSI data set that has the zone or the SMPPTS data set.

You specified the PROCESS= WAIT parameter on the EXEC statement of the SMP/E job.

The task needs one or more CSI data sets and/or the SMPPTS data set to run the job. Another task, or someone using the administration dialogs, is using one or more of the data sets that the task needs. The task will continue to wait until all the data sets are available, or until you cancel the task.

**System Action:** None.

**Programmer Response:** None.

**GIM600s** *command* COMMAND IS BEING TERMINATED BECAUSE A CSI OR THE SMPPTS DATA SET WAS NOT AVAILABLE**Explanation:**

- *command* - SMP/E command.

You specified the PROCESS=END parameter on the EXEC statement of the SMP/E job.

The command has waited 10 minutes for one or more CSI data sets or the SMPPTS data set to become available. Another task, or someone using the administration dialogs, is using a CSI data set that the task needs.

**System Action:** Command processing stops.

**Programmer Response:** Do one of the following:

- Resubmit the job when the required CSI data sets are available.
- Put the PROCESS=WAIT parameter on the EXEC statement of the SMP/E step.

**GIM601s** THE PURGE OPTION IS NOT VALID FOR AN EXPORT OF THE GLOBAL ZONE

**Explanation:** You specified the PURGE option on a ZONEEXPORT command for the GLOBAL zone. However, PURGE is not allowed for the GLOBAL zone.

**System Action:** Command processing stops.

**Programmer Response:** Do one of the following:

- If you want to export the GLOBAL zone, remove the PURGE operand from the command and run it again.
- If you want to delete the GLOBAL zone, you must delete the CSI data set that contains it.

**GIM602s** ZONE *zonename* WAS SUCCESSFULLY WRITTEN TO OUTFILE *lib***Explanation:**

- *zonename* - zone that was exported
- *lib* - ddname of the output data set.

SMP/E successfully wrote *zonename* to the OUTFILE data set. All processing was successful.

The EXPORT command can fail during zone deletion processing after SMP/E has sent this message. However, the output tape that SMP/E built is valid input for a ZONEIMPORT command.

**System Action:** None.

**Programmer Response:** None.

# GIM6XX

**GIM603s THE ZONE TYPES ARE INCOMPATIBLE, *name1* IS *type1* AND *name2* IS *type2***

**Explanation:**

- *name1* - SET to zone
- *type1* - type of the SET to zone (DLIB, GLOBAL, or TARGET)
- *name2* - INPUT zone
- *type2* - type of the INPUT zone (DLIB, GLOBAL, or TARGET)

The INPUT and SET to zone types are not the same and are not a valid combination (DLIB to TARGET).

**System Action:** Command processing stops with a return code of 12. SMP/E does not move any data.

**Programmer Response:** Specify different zones to fix the mismatch in zone types and submit the job again.

**GIM604s THE SET BOUNDARY ZONE *zonename1* AND THE {INTO ZONE|INPUT ZONE} *zonename2* HAVE DIFFERENT NAMES**

**Explanation:**

- *zonename1* - SET BOUNDARY zone name
- *zonename2* - INTO|INPUT zone name.

The SET BOUNDARY and the INTO|INPUT zone names are different.

**System Action:** Command processing stops.

**Programmer Response:** Match the SET to and INTO|INPUT zone names and run the command again.

**GIM605s THE INTO ZONE *zonename* ALREADY EXISTS IN THE *dsname***

**Explanation:**

- *zonename* - name of the INTO zone
- *dsname* - name of the receiving CSI data set.

The receiving zone cannot exist in the CSI data set.

**System Action:** Command processing stops with a return code of 12. SMP/E does not move any data.

**Programmer Response:** Do one of the following:

- Change the name of the receiving zone.
- Delete the current zone before you rerun the command.

**GIM606s THE GLOBAL ZONE IS BEING IMPORTED TO *dsname* THAT ALREADY CONTAINS A ZONE****Explanation:**

- *dsname* - CSI data set receiving the zone data.

SMP/E has to import the GLOBAL zone to a CSI that only has a ZPOOL record. When you see this message, the CSI already has another zone.

**System Action:** Command processing stops with a return code of 12. SMP/E does not move any data.

**Programmer Response:** Do one of the following:

- Change the name of the receiving CSI.
- Delete the existing zones.

**GIM607s THE IMPORT OF ZONE *zonename* FROM INFILE *lib* IS INCOMPLETE****Explanation:**

- *zonename* - receiving zone name
- *lib* - name of the DD statement definition for the INFILE data set.

An end-of-file on the INFILE data set occurred before SMP/E read the special terminating data record. This could be due to either an I/O error or an EXPORT that stopped before it completed.

**System Action:** Command processing stops with a return code of 12. SMP/E deleted all data that it imported.

**Programmer Response:** Do one of the following:

- If a previous error shows that an I/O error on the INFILE data set caused the problem, use a different input device to run the job again.
- If SMP/E produced an I/O error, check the EXPORT that created the INFILE data set. If the EXPORT did not complete successfully, rerun the EXPORT to create a new INFILE data set.
- If you cannot determine the problem, export the original zone to create a new INFILE data set.

**GIM608s ZONE *zonename* TYPE WAS CHANGED FROM DLIB TO TARGET****Explanation:**

- *zonename* - name of the output zone.

The input zone is a DLIB type, which SMP/E is changing to a TARGET type when it is importing or copying this zone.

# GIM6XX

**System Action:** None.

**Programmer Response:** None.

**GIM609s** THERE IS NO ZONE DEFINITION RECORD FOR ZONE *zonename*

**Explanation:**

- *zonename* - name of the input zone.

The target or DLIB zone that SMP/E is processing does not have the zone type definition. The system assumes that the zone does not have valid data.

**System Action:** Command processing stops with a return code of 12.

**Programmer Response:** Do the following:

1. Check that *zonename* is the zone you want SMP/E to process.
2. If *zonename* is the correct zone, use UCLIN to define the zone type in the zone.
3. Rerun the command.

**GIM610s** AN ENTRY TYPE OF *type* IS NOT VALID, THE ENTRY TYPE MUST BE DDDEF OR UTILITY

**Explanation:**

- *type* - entry type specified on the ZONEEDIT command.

The valid entry types for the ZONEEDIT command are DDDEF and UTILITY. The type specified was neither of these.

**System Action:** Command processing stops.

**Programmer Response:**

- If you want to change a DDDEF or UTILITY entry, re-enter the ZONEEDIT command with the correct entry type.
- If you want to change any other type of entry, use the UCLIN command.

**GIM611s** A FIELD NAME OF *field* IS NOT VALID FOR THE ENTRY TYPE {DDDEF | UTILITY}

**Explanation:**

- *field* - field name.

The ZONEEDIT command did not specify a valid field name for the entry type chosen.

For DDDEF, the valid field names are:

- DATASET or DA
- SYSOUT
- UNIT
- VOLUME
- WAIT

For UTILITY, the valid field names are:

- NAME
- PRINT

**System Action:** The change is not made and command processing stops.

**Programmer Response:** Re-enter the ZONEEDIT command with a valid field name for the entry type chosen.

**GIM612s** **DUPLICATE ++MOVE STATEMENTS FOR** *element* **TO LIBRARY** *ddname* -  
**SYSMOD** *sysmod*

**Explanation:**

- *element* - name of the element to be moved
- *ddname* - name of the receiving library
- *sysmod* - SYSMOD-ID of the sysmod being processed.

The same element cannot be moved from two different system libraries into the same system library.

**System Action:** SYSMOD processing stops with a return code of 8.

**Programmer Response:**

- If there are two copies of the element, move one. Then delete the other by doing the following:
  - Use a ++DELETE MCS statement for LMODs.
  - Use a ++MOD, ++MAC, or ++SRC MCS statement for other elements.
- If there is only one copy of the element, take out the extra ++MOVE MCS statement.

Then rerun the job.



# GIM6XX

## GIM613s THE NAME OF THE ZONE TO BE IMPORTED (*zonename1*) AND THE ZONE TO IMPORT (*zonename2*) ARE DIFFERENT

### Explanation:

- *zonename1* - name of the zone specified on the ZONEIMPORT command
- *zonename2* - name of the zone in the INFILE data set.

The name of the zone you want to import is not the same as the name of the zone in the INFILE data set. You specified any of the following:

- The wrong zone name
- The wrong ddname in the INFILE parameter.

**System Action:** ZONEIMPORT processing stops.

**Programmer Response:** Do one of the following:

- If you specified the wrong name on the ZONEIMPORT command, correct the zone name and submit the job again.
- If you specified the wrong data set, correct the data set on the INFILE PARAMETER and submit the job again.

## GIM614s INFILE DATA SET IS NOT A VALID FORMAT FOR A ZONEIMPORT

**Explanation:** The format of INFILE data set is wrong for ZONEIMPORT.

**System Action:** ZONEIMPORT processing stops.

**Programmer Response:** Do one of the following:

- If the data set that the INFILE operand specified is not the output from ZONEEXPORT, specify the correct input data set and submit the job again.
- If the data set is correct, use ZONEEXPORT to recreate the data set. Submit the job using the new output from ZONEEXPORT.

## GIM615s THE NAME OF THE IMPORTED ZONE WAS CHANGED FROM *zonename2* TO *zonename1*

### Explanation:

- *zonename1* - name of the receiving zone
- *zonename2* - name of the zone that SMP/E is importing, the name of the zone that the ZONEEXPORT command exported.

The imported and receiving zones have different names. SMP/E received the zone and changed the name of the zone to *zonename1*.

**System Action:** SMP/E processes the command.

**Programmer Response:** None.

**GIM616s THE INPUT AND RECEIVING ZONES HAVE THE SAME NAME**

**Explanation:** The input and receiving zones have the same name; this is invalid for ZONECOPY.

**System Action:** ZONECOPY processing stops with a return code of 12.

**Programmer Response:** Change one of the zone names and run the command again.

**GIM617s *type name* TO BE {MOVED | RENAMED | DELETED } DOES NOT EXIST IN LIBRARY *lib* - SYSMOD *sysmod***

**Explanation:**

- *type* - member entry (MAC, MOD, LMOD, or SRC) or ALIAS name
- *name* - name of the member or alias to be processed
- *lib* - ddname of the library that contains the member or alias
- *sysmod* - SYSMOD-ID.

Member *name* has an entry in the target or distribution zone but does not exist in *lib*. It may have been moved, renamed, or deleted not using an SMP/E command, or its entry may have been created using UCLIN or JCLIN.

**System Action:** SYSMOD processing stops.

**Programmer Response:** First determine why the member does not exist in the library.

- If it was moved, deleted, or renamed not using an SMP/E command, reconstruct it in *lib*.
- If the entry was created using UCLIN or JCLIN, decide whether to create the member in the library or to delete the entry from the zone.

Then rerun the job.

**GIM618s THE ++MOVE FOR *name* CONTAINS THE SAME {SYSLIB | DISTLIB} AND {TOSYSLIB | TODISTLIB} *lib* - SYSMOD *sysmod***

**Explanation:**

- *name* - member name
- *lib* - ddname of the library from which to move the member
- *sysmod* - SYSMOD-ID of the sysmod in error.

SMP/E cannot move a member from one library to another library with the same name.

## GIM6XX

**System Action:** SYSMOD processing stops with a return code of 8.

**Programmer Response:** If the SYSMOD is supposed to move the member, make sure the ddnames of the current and new libraries are different. Otherwise, delete the ++MOVE MCS statement. Then rerun the job.

**GIM619s** *type name* **TO BE {MOVED | RENAMED | DELETED} DOES NOT EXIST IN THE**  
*zonename* **ZONE - SYSMOD** *sysmod*

**Explanation:**

- *type* - element type (MAC, MOD, LMOD, or SRC)
- *name* - element name
- *process* - MOVED, RENAMED, or DELETED
- *zonename* - target or distribution zone name
- *sysmod* - SYSMOD-ID.

SMP/E cannot find an entry for element *name* in zone *zonename*.

**System Action:** The system action depends on the processing being done:

- If you are restoring an entry for a renamed LMOD, processing stops.
- If you are accepting a MAC, MOD, or SRC entry with DISTLIB changes, processing stops.
- Otherwise, processing continues.

**Programmer Response:** If processing stopped, make the missing entry available to SMP/E and rerun the job. Otherwise, no action is needed.

**GIM620s** *type name* **WAS MOVED FROM {SYSLIB | DISTLIB} *oldlib* TO *newlib* - SYSMOD**  
*sysmod*

**Explanation:**

- *type* - member type (MAC, MOD, LMOD, or SRC) or ALIAS name
- *name* - member or alias name
- *oldlib* - ddname of the library that contained the member before the move
- *newlib* - ddname of the library that contains the member after the move
- *sysmod* - SYSMOD-ID.

The member was successfully moved from the old library to the new one.

**System Action:** None.

**Programmer Response:** None.

**GIM621s** **KEYWORD** *type* **IS {NOT VALID | REQUIRED} FOR A {SYSTEM | DLIB} TYPE MOVE - SYSMOD** *sysmod*

**Explanation:**

- *type* - type of the incorrect or missing keyword

If the message says the keyword was NOT VALID, the possible types are DLIB, LMOD, MOD, and SYSTEM.

If the message says the keyword was REQUIRED, the possible types are TOSYSLIB and SYSLIB for a SYSTEM type MOVE, and TODISTLIB and DISTLIB for a DLIB type MOVE.

- *sysmod* - SYSMOD-ID.

A keyword for a ++MOVE MCS statement is either incorrect or missing.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Do one of the following:

- If the keyword was incorrect, remove it or replace it with the correct keyword.
- If the keyword was missing, add it to the SYSMOD.

Then rebuild the SYSMOD and rerun the job.

**GIM622s** *mcs* **MCS IS OUT OF SEQUENCE IN SYSMOD** *sysmod*

**Explanation:**

- *mcs* - MCS type (++MOVE, ++RENAME, or ++DELETE)
- *sysmod* - SYSMOD-ID.

The specified MCS statement is out of order. It must be after the ++VER and ++IF MCS statements but before any ++JCLIN or element MCS statements.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Put the MCS statement where it belongs, then rerun the job.

**GIM623s** ++DELETE OF COPIED LMOD *name* **IS NOT SUPPORTED - SYSMOD** *sysmod*

**Explanation:**

- *name* - name of the load module
- *sysmod* - SYSMOD-ID.

The SYSMOD contained a ++DELETE MCS statement for a copied load module. This is not allowed.

## GIM6XX

**System Action:** The ++DELETE MCS statement is skipped, and SYSMOD processing stops with a return code of 8.

**Programmer Response:** To delete the load module, add a ++MOD DELETE MCS statement.

Then rerun the job.

**GIM624s** *operand* **OPERAND IS REQUIRED TO { ++MOVE | ++RENAME }**  
*name* - SYSMOD *sysmod*

**Explanation:**

- *operand* - the missing operand:
  - *TONAME* - the new name for the LMOD
  - *TYPE* - the element type (MOD, MAC, SRC, or LMOD)
  - *LIBRARY* - the library (SYSLIB and TOSYSLIB, or DISTLIB and TODISTLIB)
- *name* - name of the element to be moved, deleted, or renamed
- *sysmod* - SYSMOD-ID of the SYSMOD containing the element.

The ++MOVE or ++RENAME MCS statement did not specify a required keyword or value.

**System Action:** SYSMOD processing stops. The SYSMOD is not received.

**Programmer Response:** Add the required keyword, rebuild the SYSMOD and rerun the job.

**GIM625s** {DUMP ID | MSG ID} *value* **NOT SUPPORTED BY DEBUG**

**Explanation:**

- *value* - the value of the dump ID or message ID that is not supported.

If the message shows DUMP ID, the dump ID specified on the DEBUG command is not a defined dump point.

If the message shows MSG ID, the message ID specified is not an SMP/E message (it does not begin with the prefix 'GIM').

**System Action:** Command processing stops.

**Programmer Response:** Re-enter the DEBUG command with the correct dump ID or message ID.

**GIM626s** THE DATA SET NAME *oldname* IS NOT CHANGED SINCE THE NEW NAME *newname* IS LONGER THAN 44 CHARACTERS

**Explanation:**

- *oldname* - current data set name
- *newname* - new data set name.

A data set name can have up to 44 characters. The new name would have been too long, so the data set name was not changed.

**System Action:** Command processing stops with a return code of 8.

**Programmer Response:** Re-enter the ZONEEDIT command so the new name is 44 characters or less.

**GIM627s** {++MOVE | ++DELETE} WILL NOT BE DONE FOR *type name* IN SYSMOD *sysmod* - *operand* VALUES UNEQUAL

**Explanation:**

- *type* - entry type (MAC, MOD, SRC, LMOD, ALIAS)
- *name* - entry name
- *sysmod* - SYSMOD-ID of the SYSMOD that contains the ++MOVE or ++DELETE MCS statement
- *operand* - SYSLIB, DISTLIB, ALIAS, or FMID.

SMP/E detected that the values specified on the indicated MCS statement did not match the information that is currently in the indicated entry.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Correct the discrepancies and rerun the job.

**GIM629s** LMOD *oldlmod* WAS RENAMED TO *newlmod* IN LIBRARY *lib* - SYSMOD *sysmod*

**Explanation:**

- *oldlmod* - old LMOD name
- *newlmod* - new LMOD name
- *lib* - ddname of the library that contains the LMOD
- *sysmod* - SYSMOD-ID.

The LMOD was successfully renamed.

**System Action:** None.

**Programmer Response:** None.

# GIM6XX

**GIM630s** SYSLIB OPERAND IS REQUIRED TO ++DELETE *lmod* - SYSMOD *sysmod*

**Explanation:**

- *lmod* - LMOD name
- *sysmod* - SYSMOD-ID.

The ++DELETE MCS statement is missing the SYSLIB operand. SMP/E cannot delete a load module from a SYSLIB if that SYSLIB is not specified.

**System Action:** SYSMOD processing stops.

**Programmer Response:** Add the missing SYSLIB operand, rebuild the SYSMOD, and rerun the job.

**GIM632s** *type name* WAS DELETED IN LIBRARY *lib* - SYSMOD *sysmod*

**Explanation:**

- *type* - entry type (LMOD or ALIAS)
- *name* - entry name
- *lib* - ddname of the library that contained the LMOD
- *sysmod* - SYSMOD-ID.

The LMOD or ALIAS was successfully deleted.

**System Action:** None.

**Programmer Response:** None.

**GIM633s** SYSMOD *sysmod* CONTAINS ++DELETE FOR LOAD MODULE(S) AND CANNOT BE RESTORED

**Explanation:**

- *sysmod* - SYSMOD-ID that contained the ++DELETE MCS statement.

When SYSMOD *sysmod* was installed, SMP/E deleted one or more load modules. Since load modules cannot be rebuilt, they cannot be restored.

**System Action:** RESTORE processing stops.

**Programmer Response:** None.

**GIM634s ZONESET *name* IS NOT DEFINED IN THE GLOBAL ZONE****Explanation:**

- *name* - name of the ZONESET specified in the REPORT command.

The GLOBAL zone does not contain the ZONESET entry specified on the REPORT command.

**System Action:** REPORT processing stops with a return code of 12.

**Programmer Response:** Make sure the ZONESET name you specified is correct.

- If the name is incorrect, enter a different name or change the spelling and process the REPORT command again.
- If the name is correct, use the UCLIN to define the ZONESET in the GLOBAL zone, then process the REPORT command again.

**GIM635s THE INPUT AND RECEIVING CSI ARE THE SAME**

**Explanation:** The input and receiving zones must be in different CSIs.

**System Action:** ZONECOPY processing stops with a return code of 12.

**Programmer Response:** Do one of the following:

- Create the receiving zone in another CSI.
- Use ZONEMERGE or ZONEEXPORT and ZONEIMPORT to copy the zone in the same CSI.

**GIM639s REQUIRED OPERAND *operand* WAS NOT SPECIFIED ON THE ZONE *utility* COMMAND****Explanation:**

- *operand* - name of the missing operand
- *utility* - utility name.

SMP/E needs *operand* to process the specified zone utility command.

**System Action:** Command processing stops with a return code of 12.

**Programmer Response:** Add the missing operand, *operand*, to the command and rerun the command.



## GIM6XX

### GIM640s JOBCARD MEMBER *member* WAS NOT FOUND IN *lib* - A JCL COMMENT WILL BE GENERATED EACH TIME A JOB CARD IS REQUIRED

#### Explanation:

- *member* - member that the JOBCARD operand specifies. The default *member* name is "JOBCARD"
- *lib* - ddname in the JOBCARD operand.

Using the JOBCARD (ddname, member) operand of GENERATE, you specified that the job card that SMP/E will use exists as *member* in the data set, *lib*. SMP/E could not find *member* in that data set.

**System Action:** GENERATE processing continues. SMP/E generates a JCL comment in place of the JOB card.

**Programmer Response:** Do one of the following:

- Add the JOBCARD member to the data set that you specify in the JOBCARD operand. Rerun the GENERATE command.
- Edit the GENERATE output to find the JCL comment that the system generated. (The format of the JCL comment is defined under "The GENERATE Command" in the *SMP/E Reference*). Change the JCL comment so it is a valid job card.

### GIM641s MOD *mod* MUST BE ASSEMBLED BUT NO ASSEM OR SRC ENTRY FOUND - NO ASSEMBLY WILL BE SCHEDULED

#### Explanation:

- *mod* - name of the module that SMP/E must assemble.

SMP/E found a MOD entry with a DISTLIB of SYSPUNCH, showing that the module is not in any distribution library. Instead:

1. The module is in source format, which SMP/E must assemble each time it processes the module.
2. SMP/E normally gets the source text either from one of the following:
  - A target zone ASSEM entry
  - An SRC entry with the same name as the MOD entry.

For this module, SMP/E could not find an ASSEM or SRC entry. Therefore, SMP/E could not do the assembly.

**System Action:** GENERATE processing continues; SMP/E does not schedule an assembly.

**Programmer Response:** Do one of the following:

- If SMP/E does not need the MOD entry anymore, delete it.
- Run the appropriate JCLIN to add an ASSEM entry or to add an SRC entry to the target zone.

**GIM642s MOD *mod* WHICH COULD NOT BE ASSEMBLED IS PART OF LMOD *lmod* - IT IS NOT INCLUDED IN THE LMOD LINK EDIT**

**Explanation:**

- *mod* - name of a module that message GIM641 identifies
- *lmod* - name of the load module where the module is link-edited.

Since SMP/E could not assemble the identified module, there is no object deck to link-edit into the identified load module.

**System Action:** SMP/E will link-edit the load module without the module.

**Programmer Response:** Find out why the module was not assembled (message GIM641) and do one of the following:

- Correct the problem and rerun the GENERATE command.
- Add an assembly step to the GENERATE output and link-edit the INCLUDE card in the load module link step.

**GIM643s {MAC | SRC} *name* HAS NO FMID - IT IS SELECTED FOR COPY FROM *dlib* TO *tglib* BUT AN ERROR MAY RESULT IF IT IS NOT FOUND**

**Explanation:**

- *name* - macro or source name
- *dlib* - distribution library ddname for the macro or source
- *tglib* - target library ddname for the macro or source.

According to SMP/E, every element (macro or source) in the target zone was part of a function that SMP/E installed into the target zone. Each macro or source entry identifies the function (FMID) that installed it.

In this case, the identified macro or source entry does not have an FMID subentry.

**System Action:** SMP/E will copy the macro or source entry from the distribution library data set to the system library data set. However, you may get an error if the macro or source entry is not in the distribution library.

**Programmer Response:** Do one of the following:

- Update the macro or source entry to reflect the current owner.
- Delete the macro or source entry if SMP/E does not need it anymore.

## GIM6XX

**GIM644s MOD *mod* HAS NO FMID - IT IS SELECTED FOR {COPY | LINK} TO LMOD *lmod* IN LIBRARY *ddname* BUT AN ERROR MAY RESULT IF IT IS NOT FOUND**

**Explanation:**

- *mod* - module name
- *lmod* - load module name
- *ddname* - ddname of the library where the load module resides.

According to SMP/E, every element (module) in the target zone was part of a function that SMP/E installed into the target zone. Each module identifies the function (FMID) that installed it.

In this case, the identified module entry does not have an FMID subentry.

**System Action:** SMP/E will copy or link the module into the load module, *lmod*. However, you may get an error if the module is not in the distribution library.

**Programmer Response:** Do one of the following:

- If the function that SMP/E installed into this target zone owns the module, update the module entry to include that FMID.
- If a function in another target zone owns the module, do nothing. This condition occurs when one function includes a module from another function. During JCLIN processing, SMP/E creates a MOD entry for the module; however, since SMP/E never installs the module in this zone, SMP/E never adds the FMID subentry.
- If SMP/E does not need the module; delete it.

**GIM645s LMOD *lmod* WAS FOUND BUT NO MOD ENTRIES WERE FOUND THAT SPECIFIED *lmod* IN THE MOD LMOD LIST**

**Explanation:**

- *lmod* - load module name.

An LMOD entry is in the target zone, but there are not any MOD entries to show that the module was part of that load module.

**System Action:** Since SMP/E does not know what modules make up the load module, SMP/E cannot schedule a link-edit. GENERATE processing continues.

**Programmer Response:** Do one of the following:

- Add LMOD subentries to the appropriate MOD entries and rerun the GENERATE command.
- If SMP/E does not need the load module anymore; delete it.

**GIM646s UNABLE TO GENERATE *statement* JCL STATEMENT IN STEP *stepname* OF JOB *jobname* - A JCL COMMENT HAS BEEN GENERATED TO INDICATE THE ERROR**

**Explanation:**

- *statement* - JCL statement that SMP/E did not generate
- *stepname* - name of the step that required the statement
- *jobname* - name of the job that required the statement.

SMP/E could not get the information to generate the specified JCL statement.

**System Action:** SMP/E generated a JCL comment to replace the JCL statement it needed. (The JCL comment format is defined in “The GENERATE Command” in the *SMP/E Reference*). Processing continues, but the job will not run unless you fix the error.

**Programmer Response:** Do one of the following:

- Add the appropriate DDDEF entries and rerun the GENERATE command.
- Edit the GENERATE output to add the appropriate JCL statements in place of the generate comments.

**GIM647s INCOMPLETE *command* COMMAND - THE *operand* OPERAND IS REQUIRED**

**Explanation:**

- *command* - SMP/E command
- *operand* - name of the missing operand.

You did not specify a required operand, *operand*, on the *command* command.

**System Action:** Command processing stops with a return code of 12.

**Programmer Response:** Rerun the command with the required operand.

**GIM648s {MOD | MAC | SRC} *name* HAS NO DISTLIB SUBENTRY - IT CANNOT BE PROCESSED**

**Explanation:**

- *name* - name of the MOD, MAC, or SRC entry.

SMP/E assumes that all elements (modules, macros, and source entries) exist in a distribution library. SMP/E found an entry that did not have a DISTLIB subentry. Since the GENERATE command processes all elements from the distribution library to the appropriate target library, SMP/E could not process this element.

**System Action:** SMP/E did not install the module, macro, or source entry into any target library.

**Programmer Response:** Use UCLIN to add the DISTLIB subentry to the MOD, MAC, or SRC entry and rerun the GENERATE command.

# GIM6XX

## GIM649s NO ELEMENTS SELECTED FOR INSTALLATION - NO JOBS GENERATED

**Explanation:** SMP/E looked at the target zone, but did not find any modules, macros, or source entries that the system had to install into any system library. SMP/E did not generate an install job stream.

**System Action:** Processing completes normally.

**Programmer Response:** Do one of the following:

- If you set the FORFMID operand, check that the FMID is valid and rerun the GENERATE command.
- If you did not set the FORFMID operand, check that you have primed the target zone. You should have run the JCLIN function to add the information that lets SMP choose how to install the various elements. Rerun the GENERATE command.

## GIM650s {PROGRAM | PROCEDURE} *program* IN JOB *jobname* STEP *stepname* IS NOT BEING PROCESSED AS JCLIN [- SYSMOD *sysmod*]

**Explanation:**

- *program* - program or procedure name specified on the JCLIN EXEC statement
- *jobname* - job name
- *stepname* - step name
- *sysmod* - SYSMOD-ID that contains a ++JCLIN MCS statement.

SMP/E did not recognize the program or procedure name as a valid copy, assembler, linkage editor, or update utility or procedure.

*Note: If the JCLIN command is being processed, SYSMOD sysmod does not appear in the message. If inline JCLIN is being processed, the SYSMOD-ID does appear.*

**System Action:** The step is not processed, and SMP/E goes on to the next step. You will get a return code of 4 if SMP/E does not recognize a specified program.

**Programmer Response:**

Check if it is necessary for SMP/E to recognize the program at this point. If so, specify a recognized program or utility on the JCLIN command or ++JCLIN MCS statement. Then rerun the job.

Then rerun the job.

## GIM651s ZONE *zonename* IS AN EMPTY ZONE

**Explanation:**

- *zonename* - name of the zone.

You issued the ZONEEXPORT command for a zone that has no data for SMP/E to process.

**System Action:** Command processing stops with a return code of 12.

**Programmer Response:** Check the zone name and do one of the following:

- If the zone name is correct, build the zone and rerun the command.
- If the zone name is incorrect, change the zone name and rerun the command.

**GIM652s ZPOOL RECORD MISSING IN *dsname* CSI**

**Explanation:**

- *dsname* - name of the CSI data set.

You did not initialize the CSI data set, *dsname*, with a ZPOOL record.

**System Action:** Command processing stops with a return code of 12.

**Programmer Response:** Check the *dsname* name and do one of the following:

- If you specified the correct name, do the following:
  1. Initialize the *dsname* with a ZPOOL record.
  2. Create the zones and rerun the job.
- If you specified the incorrect name, change the *dsname* name and rerun the job.

**GIM653s THE FIELD NAME *field* HAS BEEN SPECIFIED MORE THAN ONCE ON A ZONEEDIT CHANGE STATEMENT**

**Explanation:**

- *field* - field name.

A single ZONEEDIT CHANGE statement specified the same field name more than once. This is not allowed.

**System Action:** Command processing stops.

**Programmer Response:** Fix the CHANGE statement to specify the field only once. If there is more than one change for the same field, code a CHANGE statement for each change. Then rerun the job.

**GIM654s ZONEDESCRIPTION EXCEEDS 500 CHARACTERS - THE FIRST 500 CHARACTERS ARE SAVED**

**Explanation:** SMP/E allows up to 500 characters in a zone description.

**System Action:** SMP/E saves the first 500 characters of the zone description and processing continues.

**Programmer Response:** If the first 500 characters are enough, no action is required. Otherwise, use the UCLIN REP command to replace the saved description with a new, shorter one.

# GIM6XX

**GIM655s** ZONE *zonename* IN THE FORZONE ENTRY IS NOT DEFINED IN THE *name* ZONESET

**Explanation:**

- *zonename* - name of the zone
- *name* - name of the ZONESET entry.

The zone *zonename* in the FORZONE operand of the REPORT command is not defined in the *name* ZONESET. All zones in the FORZONE operand must be defined in the ZONESET.

**System Action:** REPORT processing stops with a return code of 12.

**Programmer Response:**

1. Do either of the following:
  - a. Remove the zone name from the FORZONE operand.
  - b. Add the zone name to the ZONESET using UCLIN.
2. Then rerun the REPORT command.

**GIM656s** ALL THE ZONES IN THE ZONESET MUST BE OF THE *zonetype* TYPE. ZONE *zonename* IS A *zonetype* TYPE

**Explanation:**

- *zonename* - name of the zone
- *zonetype* - TARGET or DLIB.

All the zones contained in a ZONESET must be the same type, either all target zones or all distribution zones. *zonename* is different from the other zones in the ZONESET.

**System Action:** REPORT processing stops with a return code of 12.

**Programmer Response:** Remove the zones that are not of the same type from the ZONESET, then rerun the REPORT command.

**GIM657s** DISTRIBUTION ZONE *zonename* IS NOT AVAILABLE - LOAD MODULE BUILD PROCESSING CANNOT CHECK THE DISTRIBUTION ZONE

**Explanation:**

- *zonename* - name of the distribution zone.

SMP/E is trying to build a load module in a system library using modules in the related distribution library. However, the distribution zone for that library is not available.

**System Action:** SMP/E stops building the load modules for which it needs the distribution zone. SMP/E also issues the following messages:

- GIM674, which lists the load modules SMP/E terminated
- GIM675, which lists the modules SMP/E requires to complete the load modules
- GIM226, which lists the SYSMOD(s) SMP/E terminated because of load module termination.

**Programmer Response:** Make the required distribution zone available and rerun the APPLY.

**GIM658s MODULE *mod* WAS NOT FOUND IN DISTRIBUTION ZONE *zonename***

**Explanation:**

- *mod* - name of the module not found in the distribution zone
- *zonename* - name of the distribution zone.

SMP/E could not find the indicated module in the distribution zone during load module build processing. Message GIM673, which follows this message, identifies the affected load modules and target libraries.

**System Action:** The load module is link-edited without the missing module and processing continues.

**Programmer Response:** Link-edit the module into the load module outside SMP/E.

**GIM659s MODULE *mod* WAS FOUND IN DISTRIBUTION ZONE *zonename* AT A DIFFERENT LEVEL THAN THAT OF THE TARGET ZONE MODULE**

**Explanation:**

- *mod* - module name
- *zonename* - name of the distribution zone.

SMP/E required the specified module to build a load module. It found the module in the distribution zone, but the module was at the wrong service level. Message GIM673, which follows this message, lists the affected load modules.

**System Action:** The load module is link-edited without the module and processing continues.

**Programmer Response:** Link-edit the module into the load module outside SMP/E.



## GIM6XX

### GIM660s UNABLE TO OBTAIN STORAGE FOR JCLIN CROSS REFERENCE REPORT - ALTERNATE REPORT FORMAT USED

**Explanation:** SMP/E does not have enough storage to format the JCLIN Cross-Reference report in two columns. Single-column format will be used.

**System Action:** None.

**Programmer Response:** None.

### GIM661s UNABLE TO OBTAIN STORAGE FOR JCLIN WORK AREAS; THE JCLIN CROSS REFERENCE REPORT WILL NOT BE PRODUCED

**Explanation:** SMP/E could not obtain enough storage to produce the JCLIN Cross Reference report.

**System Action:** JCLIN processing continues, but the Cross Reference report will not be produced.

**Programmer Response:** None.

### GIM662s THERE ARE NO {DDDEF | UTILITY} ENTRIES FOR ZONE *zonename*

**Explanation:**

- *zonename* - zone name.

The DDDEF or UTILITY entry to be changed does not exist in zone *zonename*.

**System Action:** Command processing stops.

**Programmer Response:**

1. Check to see if the zone specified is the one you want to change.
2. If the zone is correct, check to see if the job or command to define the entries was successful.

### GIM663s ACCJCLIN IS ONLY VALID IN A DLIB ZONE DEFINITION

**Explanation:**

ACCJCLIN was specified on a UCLIN command to control whether inline JCLIN is saved at ACCEPT time. However, the boundary was not set to a distribution zone.

**System Action:** SMP/E does not change the ACCJCLIN value. Processing continues with the next command.

**Programmer Response:** Set the boundary to a distribution zone and run the UCLIN again.

**GIM665s** {SYSLIB | ALIAS | FMID} OPERAND ON {++MOVE | ++DELETE} FOR *name*  
HAS DUPLICATE VALUE *value* - SYSMOD *sysmod*

**Explanation:**

- *name* - name of the element being deleted
- *value* - the value that is duplicated
- *sysmod* - SYSMOD-ID of the SYSMOD containing the element.

The ++MOVE or ++DELETE MCS statement specifies the same operand value more than once.

**System Action:** The SYSMOD is not received.

**Programmer Response:** Remove the duplicated value and process the SYSMOD again.

**GIM666s** {TONAME | ALIAS} VALUE *value* SAME AS ELEMENT NAME *name* ON  
{++DELETE | ++RENAME} - SYSMOD *sysmod*

**Explanation:**

- *value* - the value of the conflicting operand
- *name* - name of the conflicting element
- *sysmod* - the SYSMOD-ID of the SYSMOD containing MCS statement.

The TONAME or ALIAS name on a ++DELETE or ++RENAME MCS statement must be different from the element name.

**System Action:** The SYSMOD is not received.

**Programmer Response:** Change the TONAME, ALIAS name, or element name on the indicated MCS statement and process the SYSMOD again.

**GIM667s** SNAP IS NOT VALID WITHOUT DUMP ID

**Explanation:**

On the DEBUG command, you must enter DUMPON, a dump-id, and SNAP to get a SNAP dump. These other operands were missing from the command you entered.

**System Action:** DEBUG processing stops with a return code of 16.

**Programmer Response:** Correct the DEBUG command by adding DUMPON and a dump-id, then rerun the job.

## GIM6XX

**GIM668s** {MACRO | SOURCE} ENTRY *name* NOT DELETED FROM THE {SMPMTS | SMPSTS} BECAUSE NO {MACRO | SOURCE} WAS FOUND IN THE *zonename* TARGET ZONE

**Explanation:**

- *name* - entry name
- *zonename* - target zone name.

During CLEANUP processing, SMP/E could not delete the specified entry from the SMPMTS or SMPSTS data set because no corresponding entry existed in the target zone.

**System Action:** CLEANUP processing continues with a return code of 8.

**Programmer Response:** None.

**GIM669s** {ALIAS | FMID | SYSLIB} VALUE COUNT EXCEEDS MAXIMUM ALLOWED OF {16 | 10 | 2} ON {++MOVE | ++DELETE} FOR *element* - SYSMOD (*sysmod-id*)

**Explanation:**

- ALIAS | FMID | SYSLIB - operand specified on ++MOVE or ++DELETE
- 16 | 10 | 2 - maximum values allowed for the operands:
  - ALIAS can have up to 16 values.
  - FMID can have up to 10 values.
  - SYSLIB can have up to 2 values.
- ++MOVE | ++DELETE - MCS type
- *element* - the name of the element being moved or deleted
- *sysmod-id* - ID of the SYSMOD that contains the ++MOVE or ++DELETE.

The number of values entered on the specified MCS statement exceeds the maximum allowed.

**System Action:** SYSMOD processing stops with a return code of 8.

**Programmer Response:** Fix the MCS statement so that it specifies an allowable number of values for the indicated operand. Then rereceive SYSMOD *sysmod-id*.

**GIM670s** THE {RELATED | OUTFILE | INFILE} NAME *value* CANNOT BE THE SAME AS THE SET TO ZONE *zone name*

**Explanation:**

- *value* - one of the following:
  - For RELATED NAME, this is the name of the RELATED zone.
  - For OUTFILE NAME, this is the ddname of the OUTFILE data set.
  - For INFILE NAME, this is the ddname of the INFILE data set.
- *zone name* - the name of the zone specified on the SET command.

One of the following errors occurred:

- On the UCLIN command used to define a TZONE or DZONE entry, the RELATED zone name was the same as the zone name specified on the SET command.
- On the ZONEEXPORT command, the OUTFILE ddname was the same as the zone name specified on the SET command.
- On the ZONEIMPORT command, the INFILE ddname was the same as the zone name specified on the SET command.

**System Action:** For UCLIN errors, command processing stops with a return code of 8. Processing continues with the next command.

For ZONEEXPORT and ZONEIMPORT errors, command processing stops with a return code of 12.

**Programmer Response:** Correct the values so they are not the same. Then rerun the command.

#### GIM671s SKIPPING TO THE ENDZONEEDIT COMMAND

**Explanation:**

A severe error occurred during ZONEEDIT processing (the return code was 12 or higher).

**System Action:** SMP/E skips to the ENDZONEEDIT command without processing any more ZONEEDIT CHANGE statements.

**Programmer Response:** See Chapter 1 of this book for a description of ZONEEDIT return codes. Fix the indicated problem and rerun the job.

#### GIM672s NO RELATED DISTRIBUTION ZONE IS DEFINED FOR TARGET ZONE *zonename* - LOAD MODULE BUILD PROCESSING CANNOT CHECK THE DISTRIBUTION ZONE

**Explanation:**

- *zonename* - name of the target zone.

While building a load module in a target library, SMP/E needed to check the related distribution zone for a required module. However, no related distribution zone is defined for the target zone.

**System Action:** SMP/E stops building load modules that need the missing distribution zone. It also issues the following messages:

- GIM674, which tells which load modules have been terminated
- GIM675, which tells which modules are required to complete the load modules

## GIM6XX

- GIM226, which tells which SYSMODs were terminated because of the load module termination.

**Programmer Response:** Define a related distribution zone in the target zone definition entry and rerun the job.

### GIM673s LOAD MODULE *lmod* IN SYSLIB(S) *lib1* [AND *lib2*] WILL NOT INCLUDE MODULE *mod*

**Explanation:**

- *lmod* - name of the load module
- *lib1* - the first system library to contain the load module
- *lib2* - the second system library to contain the load module
- *mod* - name of the module that will be missing from the load module.

SMP/E could not find *mod*, which it needs to build load module *lmod*. The load module is link-edited without the missing module.

**System Action:** Processing continues.

**Programmer Response:** Link-edit the module into the load module outside SMP/E.

### GIM674s PROCESSING TERMINATED FOR LOAD MODULE *lmod* IN SYSLIB(S) *lib1* [AND *lib2*]

**Explanation:**

- *lmod* - name of the load module
- *lib1* - one system library that is to contain the load module
- *lib2* - a second system library that is to contain the load module.

SMP/E stopped building load module *lmod* because of previous error. Check for messages (such as GIM672) that explain this error.

**System Action:** Load module processing ends. SMP/E issues the following related messages:

- GIM675, which lists the modules SMP/E requires to complete the load module
- GIM226, which lists the SYSMOD(s) SMP/E stopped processing because of the load module error.

**Programmer Response:** Correct the error indicated in the other messages and rerun the job.

**GIM675s MODULE *mod* IS REQUIRED FOR LOAD MODULE *lmod*****Explanation:**

- *mod* - name of the module needed to complete the load module
- *lmod* - name of the load module.

Because of a previous error, SMP/E cannot get the modules it needs to build load module *lmod*. This message lists these modules. A previous message (such as GIM657 or GIM672) tells what error is preventing SMP/E from getting the modules.

**System Action:** Processing for the specified load module stops. Processing continues for other load modules and SYSMODs.

**Programmer Response:** Correct the error indicated in the other messages and rerun the job.

**GIM676s ATTEMPT TO {CATALOG | UNCATALOG} TLIB DATASET *lib*  
{SUCCESSFUL | FAILED} - CATALOG RC=*rc*****Explanation:**

- *lib* - data set name
- *rc* - return code.

SMP/E automatically catalogs SMPTLIB data sets when it allocates them and automatically uncatalogs those data sets when it deallocates them. This message indicates whether SMP/E successfully cataloged or uncataloged SMPTLIB data set *lib*.

**System Action:** Processing continues with a return code of 4.

**Programmer Response:**

- If SMP/E was successful, you need not take any action.
- If SMP/E failed, but cataloging and uncataloging these data sets is not important to your installation, you need not take any action.
- If SMP/E failed, but cataloging and uncataloging these data sets is important to your installation, you must catalog the data sets yourself.

**GIM677s SMPLOGA DOES NOT HAVE THE SAME BLKSIZE AS SMPLOG - LOG  
RECORDING SWITCHED TO SMPOUT**

**Explanation:** Because the SMPLOG data set is full, SMP/E tried to record log data in SMPLOGA. However, SMPLOGA does not have the same blocksize as SMPLOG. SMP/E is therefore using the SMPOUT data set.

**System Action:** Processing continues.

**Programmer Response:** Make sure the SMPLOG and SMPLOGA data sets have the same blocksize.

# GIM6XX

**GIM678s** {*value* | ALL} {DUMP ID | MSG IDS} ALREADY OFF

**Explanation:**

- *value* - the value of the dump ID.

The DEBUG command specified DUMPOFF for a dump ID or for all message IDs. However, these dump points were already turned off.

**System Action:** DEBUG processing ends with a return code of 16.

**Programmer Response:** Run the job again without turning off the specified dump points.

**GIM679s** MOD *mod* has *name* AS BOTH A TALIAS AND LMOD SUBENTRY - THE TALIAS IS BEING IGNORED AND SHOULD BE DELETED

**Explanation:**

- *mod* - module name
- *name* - alias or load module name

The MOD entry for *mod* had *name* as both a TALIAS subentry and a LMOD subentry. This is redundant; there is no need for the TALIAS subentry.

**System Action:** GENERATE ignores the TALIAS subentry and builds a step to copy module *mod* to LMOD *name*.

**Programmer Response:** Use the UCLIN DEL command to delete the TALIAS subentry from the MOD entry.

**GIM680s** THE DUMP ID FOR *message* IS *dump-id*

**Explanation:**

- *message* - SMP/E message number
- *dump-id* - DEBUG dump ID.

This is an informational message issued for the DEBUG DUMPMSG command. SMP/E will associate the indicated dump-ID with the dump you requested for the SMP/E message.

**System Action:** Processing continues.

**Programmer Response:** None.

**GIM681s BOTH A MOD AND {MAC | SRC} ENTRY FOR *mod* WERE FOUND WITH A DISTLIB OF *distlib* - THE MOD ENTRY IS BEING IGNORED AND SHOULD BE DELETED**

**Explanation:**

- *mod* - module name
- *distlib* - DISTLIB name.

The MOD entry for module *mod* has no FMID subentry and has the same DISTLIB value as the one in the indicated MAC or SRC entry. This MOD entry was probably built during JCLIN processing. If an IEBCOPY SELECT statement for a macro or source does not include the TYPE=MAC or TYPE=SRC comment, SMP/E assumes the TYPE is MOD. It then builds a MOD entry and an LMOD entry, which are not needed for macros and source. Later, when it reads the ++MAC or ++SRC MCS statement, it builds a MAC or SRC entry.

**System Action:** GENERATE ignores the MOD entry.

- If the LMOD was copied, the LMOD entry is also ignored.
- If the LMOD was link-edited, GENERATE will build the link-edit step for the LMOD but will not include the module in error.

**Programmer Response:** Use UCLIN to delete the MOD entry. Then list the LMOD and MAC or SRC entry and do one of the following:

- If the LMOD was copied to a single SYSLIB and the SYSLIB name is the same as the MAC or SRC SYSLIB, use UCLIN to delete the LMOD entry.
- If the LMOD was linked or copied to multiple SYSLIBs and one of them is the same as the MAC or SRC SYSLIB, use UCLIN to delete that SYSLIB subentry from the LMOD entry.





**System Modification  
Program Extended**

**READER'S  
COMMENT  
FORM**

**Messages and Codes**

**Order No. GC28-1108-3**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

**Note:** *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

- As an introduction  As a text (student)  
 As a reference manual  As a text (instructor)  
 For another purpose (explain) \_\_\_\_\_

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

What is your occupation? \_\_\_\_\_

Newsletter number of latest Technical Newsletter (if any) concerning this publication: \_\_\_\_\_

If you wish a reply, give your name and address: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

IBM branch office serving you \_\_\_\_\_

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automatic mail-sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

**Reader's Comment Form**

Fold and Tape

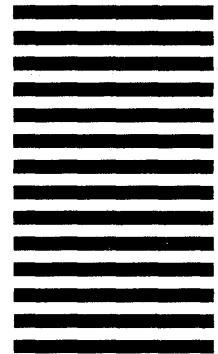
Please Do Not Staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS      PERMIT NO. 40      ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE

**International Business Machines Corporation  
Department 52Q  
Neighborhood Road  
Kingston, New York 12401**



Fold and Tape

Please Do Not Staple

Fold and Tape



**System Modification  
Program Extended**

**READER'S  
COMMENT  
FORM**

**Messages and Codes**

**Order No. GC28-1108-3**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

**Note:** *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

How did you use this publication?

- |  |   |
|--|---|
| <input type="checkbox"/> As an introduction                  | <input type="checkbox"/> As a text (student)    |
| <input type="checkbox"/> As a reference manual               | <input type="checkbox"/> As a text (instructor) |
| <input type="checkbox"/> For another purpose (explain) _____ |   |

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: \_\_\_\_\_

Comment: \_\_\_\_\_

What is your occupation? \_\_\_\_\_

Newsletter number of latest Technical Newsletter (if any) concerning this publication: \_\_\_\_\_

If you wish a reply, give your name and address: \_\_\_\_\_

IBM branch office serving you \_\_\_\_\_

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automatic mail-sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

**Reader's Comment Form**

Fold and Tape

Please Do Not Staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE

**International Business Machines Corporation  
Department 52Q  
Neighborhood Road  
Kingston, New York 12401**



Fold and Tape

Please Do Not Staple

Fold and Tape



GC28-1108-3



PRINTED IN U.S.A.

GC28-1108-3