z/OS
3.2

z/OS SMP/E User's Guide

SA23-2277-70

Note

Before using this information and the product it supports, read the information in “Notices” on page 219.

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2026-01-23

© Copyright International Business Machines Corporation 1986, 2026.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures xi

Tables xiii

About this document xv
- Who should read this publication xv
- SMP/E publications xv

How to provide feedback to IBM xvii

Summary of changes xix
- Summary of changes for z/OS 3.2 xix
- Summary of changes for SMP/E Version 3 Release 7 in z/OS 3.1 xix
- Summary of changes for z/OS SMP/E User's Guide Version 3 Release 7 in z/OS Version 2 Release 5 (V2R5) xx
- Summary of changes for z/OS SMP/E User's Guide Version 3 Release 7 in z/OS Version 2 Release 4 (V2R4) xxi

Chapter 1. SMP/E primer 1
- What is SMP/E, and why should I use it? 1
  - Understanding your system 1
  - Changing the elements of the system 2
  - Keeping track of the elements of the system 8
- How does SMP/E work? 11
  - The distribution and target libraries 11
  - The consolidated software inventory (CSI) 13
- What are the basic SMP/E commands I need to know? 15
  - Setting the zone you want to work on 15
  - Receiving the SYSMOD into SMP/E's data sets 15
  - Applying the SYSMOD to the target libraries 15
  - Restoring the target libraries to the previous level 15
  - Accepting the SYSMOD and updating the distribution libraries 15
  - Displaying SMP/E data 15
  - Flow of SMP/E SYSMOD processing 16
- Receiving the SYSMOD into SMP/E's data sets 18
  - What happens during RECEIVE processing 18
  - What happens during internet service retrieval 18
  - How SMP/E keeps track of RECEIVE processing 19
  - Using the RECEIVE command 20
  - Summary of the RECEIVE command 22
- Applying the SYSMOD to the target libraries 22
  - What happens during APPLY processing 22
  - How SMP/E keeps track of APPLY processing 23
  - Using the APPLY command 24
  - Summary 26
- Restoring the target libraries to the previous level 26
  - What happens during RESTORE processing 27
  - How SMP/E keeps track of RESTORE processing 27
  - Using the RESTORE command 28

Summary ... 30
Accepting the SYSMOD into the distribution libraries ... 30
What happens during ACCEPT processing ... 30
How SMP/E keeps track of ACCEPT processing ... 31
Using the ACCEPT command ... 32
Summary ... 34
Displaying SMP/E data ... 34
Using the query dialogs ... 35
Using the LIST command ... 37
Using the REPORT commands ... 38
SMP/E CSI application programming interface ... 38
Summary ... 38

Chapter 2. SMP/E concepts ... 41
What is SMP/E? ... 41
What are SYSMODs? ... 41
Data sets used by SMP/E ... 43
How SMP/E can help you install and maintain products ... 46
Where to begin ... 46
Installing SYSMODs ... 46
Monitoring your system ... 48
Managing the SMP/E database ... 49
Managing zones ... 50
Linking and relinking modules ... 51
General SMP/E processing ... 51

Chapter 3. Preparing to use SMP/E ... 53
Authorizing use of SMP/E commands and services ... 53
Authorizing use of unsigned GIMZIP packages ... 54
Allocating and initializing data sets in the SMP/E database ... 55
CSI data sets ... 55
PTS data sets ... 69
SCDS data sets ... 69
How to dynamically allocate data sets to be used during SMP/E processing ... 69
Sources of information for dynamic allocation ... 70
How dynamic allocation works ... 71
Defining utility programs and associated parameters to SMP/E ... 71
Using default values for utility programs ... 72
Defining values for utility programs ... 73
Example: How to request the desired utility processing ... 74
Recovering after errors from utility processing ... 75
Overview of your input to retry processing ... 75
Example: How to request the desired retry processing ... 76
Connecting SMP/E dialogs to ISPF ... 77
Check for required programs ... 77
Add dialog modules to the PCF command table ... 78
Concatenate the dialog libraries ... 78
Connect the dialogs to ISPF ... 80
Customize the SMP/E dialogs ... 80
Setting up SMP/E for easier operation ... 81
Recommended values for OPTIONS entry ... 82
Recommended DDDEF entries for link-edit utility output ... 83
Specifying automatic cross-zone requisite checking ... 83
Defining the information needed to invoke SMP/E ... 85
Required JCL statements ... 86
Sample cataloged procedure for SMP/E ... 87
Checking that you have the appropriate access ... 91

Defining exit routines....91

Chapter 4. Preparing to use Internet Service Retrieval....93
Identity and authentication overview....93
Obtaining a user certificate....94
Uploading the user certificate to z/OS....94
Setting up RACF....94
Access to the RACDCERT command....94
Creating key rings....95
Displaying certificate authority certificates....95
Adding certificate authority certificates....96
Adding the user certificate to your RACF data base....96
Connecting the certificates to the key ring....97
Sharing a user certificate among multiple user IDs....97
Debugging key ring and certificate issues....98
Replacing a user certificate that expired....98
Refreshing RACF classes....98
Setting up alternate security products....99
Defining the ORDERSERVER input for RECEIVE ORDER....99
Defining the CLIENT input for RECEIVE ORDER....100
Options that affect Java....100
Options that affect HTTPS operations....101
Options that affect download operations....101
Network configuration notes....102
Summary....102
Example....103

Chapter 5. Preparing for secure Internet delivery....105
Secure Sockets Layer overview....105
Enabling certificate authority certificates....106
Access to certificate authority certificates....106
Displaying certificate authority certificates....106
Adding certificate authority certificates....107
Access for SMP/E to use certificate authority certificates....107
Refreshing RACF classes....108
Define CLIENT input for RECEIVE and GIMGTPKG....108
Options that affect HTTPS operations....108
Options that affect FTPS operations....109
Example....112

Chapter 6. Preparing to verify signatures for GIMZIP packages....113
Digital signature overview....113
Enabling certificate authority certificates....113
Access to certificate authority certificates....113
Displaying certificate authority certificates....114
Create a key ring....114
Access for SMP/E to use certificate authority certificates....115
Refreshing RACF classes....115
Define CLIENT input for RECEIVE, GIMGTPKG and GIMUNZIP....115

Chapter 7. Installing a new function....117
Introduction....117
RECEIVE-APPLY-ACCEPT method....117
The standard RECEIVE-APPLY-ACCEPT method....117
Preparing your system....118
Staging the SYSMODs: The RECEIVE process....119
Updating the target libraries: The APPLY process....119

Testing the new function....122
Updating the distribution libraries: The ACCEPT process....122
Checking other zones for requisites: REPORT CROSSZONE....123

Chapter 8. Installing preventive service.... 125
Introduction....125
A RECEIVE ORDER request....125
Preventive service process: Summary....125
Preparing your system....126
Staging the SYSMODs: The RECEIVE process....126
Updating the target libraries: The APPLY process....126
Checking the update (APPLY CHECK)....127
Updating the target library (APPLY)....130
Installing PTFs that need special processing....130
Testing the new service level....130
Updating the distribution libraries: The ACCEPT process....131
Checking the update (ACCEPT CHECK)....131
Updating the distribution library (ACCEPT)....132
Installing PTFs that need special processing....132

Chapter 9. Installing corrective service.... 133
Introduction....133
Building or checking the fix....133
Preparing your system....135
Staging the SYSMODs: the RECEIVE process....135
Generating a service request using the RECEIVE ORDER Command....135
Updating the target libraries: the APPLY process....136
Checking the update (APPLY CHECK)....136
Updating the target library (APPLY)....137
Testing the corrective service....137
Updating the distribution libraries: the ACCEPT process....137
Checking the update (ACCEPT CHECK)....137
Updating the distribution library (ACCEPT)....138

Chapter 10. Installing a user modification.... 139
Introduction....139
Preparing your system....139
Staging the SYSMODs: The RECEIVE process....139
Updating the target libraries: The APPLY process....140
Checking the update (APPLY CHECK)....140
Updating the target library (APPLY)....140
Testing the USERMOD....141
Updating the distribution libraries: The ACCEPT process....141

Chapter 11. Managing exception SYSMODs.... 143
Introduction....143
What SMP/E does with the HOLDDATA....144
Initial entry into staging data sets: RECEIVE....144
Updating target libraries: APPLY....145
Updating distribution libraries: ACCEPT....146
Removing HOLDDATA from SMP/E data sets....146
Sources of HOLDDATA....146
CBPDO packages....146
Automated service delivery package....147
How to process HOLDDATA....147

Chapter 12. Creating cross-product, cross-zone load modules: The LINK MODULE command....149
When to use LINK MODULE....149
How to use LINK MODULE....149

Chapter 13. Displaying the data managed by SMP/E: The LIST command....151
Introduction....151
Listing all the SMP/E data....151
Listing by specific entry type....152
Listing specific entries....153
Listing by FMID or FMIDSET....154
Listing to compare two zones....154
Summary....155

Chapter 14. Changing the data SMP/E manages: The UCLIN command.... 157
Introduction....157
When to use UCLIN....157
How to use UCLIN....158

Chapter 15. Identifying cross-zone requisites: The REPORT CROSSZONE command.... 161
Introduction....161
Identifying zones to be processed....161
Running the REPORT CROSSZONE command....162
Installing the SYSMODs....163

Chapter 16. Identifying installed SYSMODs affected by error holds: The REPORT ERRSYSMODS command.... 165
Introduction....165
Running the REPORT ERRSYSMODS command....165
Installing the SYSMODs....166

Chapter 17. Listing the source IDs in a zone: The REPORT SOURCEID command.. 167
Introduction....167
Running the REPORT SOURCEID command....167
Listing the SYSMODs....167

Chapter 18. Comparing the SYSMODs installed in two zones: The REPORT SYSMODS command....169
Introduction....169
Running the REPORT SYSMODS command.... 169
Installing the SYSMODs....169

Chapter 19. Building a user modification.... 171
Choosing between a USERMOD and a function SYSMOD....171
Creating the MCSs.... 171
The ++USERMOD MCS.... 172
The ++VER MCS....172
The ++JCLIN MCS.... 173
++MOD and ++ZAP MCSs....174
++MAC and ++MACUPD MCSs....174
++SRC and ++SRCUPD MCSs....175
The ++PROGRAM MCS....175
Data element MCSs....175

Hierarchical file system element MCSs....175
Examples of USERMODs....176
Example 1: Updating a module....176
Example 2: Replacing a module....176
Example 3: Adding new modules....177
Example 4: Replacing a macro or source code....177
Example 5: Updating a macro or source code....178
Example 6: Adding new source code....178
Example 7: Adding new source code that uses an IBM-supplied macro....180
Example 8: Adding a new module that uses an IBM-Supplied macro....181

Chapter 20. Determining which SYSMODs led others to fail: The causer SYSMOD summary report....183
Introduction....183
Using causer SYSMOD information....183
Resolving errors for all SYSMODs that failed....183
Resolving errors for a single SYSMOD that failed....184
Example....184

Chapter 21. Java archive update exploiter's guide....185
JAR replacements in FMIDs....185
JAR updates in PTFs....185
JAR replacements in PTFs....186

Appendix A. Migration....187
Migration overview....187
Terms you need to know....187
SMP/E release levels....187
Developing a migration strategy....188
SMP/E V3R7 overview....189
Deprecate SHA-1 hash validation (IO29474)....189
HTTPS download support (IO20858)....190
Alternate software inventory format (IO22234)....190
GIMUNZIP extensions (IO23270)....190
ZONEMERGE command extensions (IO23466)....191
8-character userids (IO24768)....191
Binder RMODE=64 and LONGPARM options (IO25475)....191
SMP/E V3R6 overview....192
Multitasking using GIMDDALC SYSPRINT allocation....192
Adding SAF checks to SMP/E processing (IO11698)....192
Cross Global Zone Reporting....192
SYSMOD Comparison HOLDDATA Report....192
Retention of HOLDDATA (IO13643)....192
SMP/E V3R5 overview....192
Enhanced utility input....193
Long SOURCEID support....193
ZONEMERGE command....193
HTTPS and FTP enhancements....194
HOLDDATA report changes....194
BYPASS(HOLDSYS) message severity changes....194
ZONEEDIT enhancement....195
RECEIVE ORDER processing enhancements....195
Fix Category HOLDDATA....195
SMP/E V3R4 overview....195
Enhancement to the RECEIVE command....195
Impacts to SMP/E zone entries....196
ICSF not required for GIMZIP and RECEIVE FROMNETWORK....197

Improved load module build processing....197
SMP/E order management dialog....197
SMP/E query dialog....197
SMP/E V3R3 overview....197
GIMGTPKG service routine....197
Enhancements to GIMZIP and GIMUNZIP service routines....198
RECEIVE FROMNETWORK FTP interface enhancements....198
REJECT CHECK command....198
Extended RECEIVE SOURCEID processing....198
SPCLCMOD and CMWA....199
SMP/E V3R2 overview....199
LINK LMODS command....199
REPORT CALLLIBS command removal....199
UPGRADE command....199
GIMXSID service routine....199
GIMZIP: Archive segmentation....200
GIMZIP: User defined subdirectories....200
Java archive files....200
Smaller SMPLTS data set....201
DUMMY data set for SYSDEFSD....201
SMP/E dialog customization....202
GIMUTTBL removal....202
SMP/E V3R1 overview....202
Defining exit routines using SMPPARM member GIMEXITS....203
Dynamic allocation using SMPPARM member GIMDDALC....203
Enhanced link name values....203
Removal of function to create backup IEANUC01 load modules....204
Conditional JCLIN processing....204
Network delivery of SMP/E input....204
AMODE=64 and COMPAT=PM4 link edit parameters....205
Selected SMP/E data sets may now reside in a UNIX file system....205
HFS data set identification....205
SMPPTS spill data sets....205
HOLDDATA summary reports....205
SMP/E load modules and service routines moved to SYS1.MIGLIB....205
GIMXTRX service routine....205
OS/390 version 2 release 7 SMP/E overview....206
SMP/E planning and migration assistant....206
Data element reformatting....206
Description for a SYSMOD....206
Improved protection for UNIX file system files....206
Pre-built load module support....206
Product data....207
Sequential data set support....207
Shell script support....207
Symbolic link support....207
OS/390 version 2 release 5 SMP/E overview....207
CBIPO dialogs....207
Client code installation....207
Global zone merge....208
Library change interface....208
Improved load module build processing....208
Load module return code....208
Performance improvements....208
PTF compaction in SMPPTS data set....209
Enhanced RECEIVE command processing....209
Reduced SMP/E message output....209
GIMAPI: All entries and subentries support....209

GIMAPI: Version support 209
OS/390 version 1 release 3 SMP/E overview 210
API for user access to the CSI 210
Enhanced cross-zone requisite checking 210
Enhanced exception SYSMOD report 210
Enhanced ++IF FMID processing 211
Enhanced internal HOLD SYS processing 211
Enhanced ZONEEDIT command 211
Enhancements to the binder utility in DFSMS/MVS 211
System/390 service update facility 212
OS/390 version 1 release 2 SMP/E overview 212
BLOCKSIZE=8800 for SMP/E data sets 212
BUILDMCS command 212
Bypassing system holds for specific SYSMODs 212
FMIDSET selection 213
Receiving relative file data sets created from PDSEs 213
SMP/E dialogs: FIND command 213
SMP/E GIMOPCDE member moved from PARMLIB 213

Appendix B. Recommended service upgrade (RSU) 215

Appendix C. Accessibility 217

Notices 219
Terms and conditions for product documentation 220
IBM Online Privacy Statement 221
Policy for unsupported hardware 221
Minimum supported hardware 221
Trademarks 222
Glossary 223

Index 245

Figures

1. Creating load modules.... 2

2. Introducing an element.... 4

3. Preventing problems with an element.... 5

4. Fixing problems with an element.... 6

5. Customizing an element.... 7

6. PTF replacement.... 8

7. PTF prerequisite.... 9

8. Load module constructions.... 10

9. The public library.... 12

10. The distribution and target libraries.... 13

11. z/OS system with SMP/E.... 14

12. Flow of SMP/E SYSMOD processing.... 17

13. Results of RECEIVE processing.... 20

14. Results of APPLY processing.... 24

15. Results of RESTORE processing.... 28

16. Results of ACCEPT processing.... 32

17. Query selection menu.... 35

18. CSI query panel.... 36

19. CSI query - Select entry panel.... 36

20. CSI query - SYSMOD entry panel.... 36

21. Example of a SYSMOD hierarchy.... 43

22. Summary of zone relationships.... 45

23. A single-CSI structure.... 56

24. A multiple-CSI structure.....58

25. Using a separate global zone for each subsystem.....59

26. Using one CSI for the whole system.....60

27. Using a master CSI.....61

28. Using a master CSI and a separate CSI for each zone.....62

29. Using a master CSI and one CSI per SREL.....63

30. Relationships between zone definition entries.....66

31. Relationships of OPTIONS, UTILITY, zone definition entries and the SET command.....73

32. Sample logon procedure that concatenates SMP/E and ISPF libraries.....80

33. Sample SMP/E cataloged procedure.....89

34. APPLY SYSLIB concatenation: APPLY different from ACCEPT.....90

35. ACCEPT SYSLIB concatenation: APPLY different from ACCEPT.....91

36. SYSMOD Status Report: Sample Report for APPLY.....184

37. Causer SYSMOD summary report: sample report for APPLY.....184

Tables

1. Publications for IBM SMP/E for z/OS, V3R6 xv

2. Functions and resource names that must be carefully controlled 53

3. Entries controlling SMP/E processing 67

4. Entries describing the status and structure of the target and distribution libraries 67

5. Default values for UTILITY entries 72

6. How to request the utility processing 74

7. How to request the retry processing 76

8. ISPF libraries and related SMP/E target libraries 78

9. SMPTABL data set allocations 79

10. Sources for functions and their installation information 117

11. Alternatives to UCLIN 157

12. Comparison of USERMODs and function SYSMODs 171

13. Information needed to add new source code 178

About this document

This publication documents a new and enhanced version of SMP/E. New or changed information is identified by revision bars (|) to the left of the addition or change.

Who should read this publication

Anyone who uses SMP/E, or who wants to understand SMP/E processes, should read this publication.

After reading this publication, you should be able to do most SMP/E processes. You may have to refer to *z/OS SMP/E Commands* for details on commands.

SMP/E publications

Table 1. Publications for IBM SMP/E for z/OS, V3R6

Title	Description
z/OS SMP/E Messages, Codes, and Diagnosis, GA32-0883	Explains SMP/E messages and return codes and the actions to take for each; and how to handle suspected SMP/E problems.
z/OS SMP/E Commands, SA23-2275	Explains SMP/E commands and processing in detail.
z/OS SMP/E Reference, SA23-2276	Explains SMP/E modification control statements, data sets, exit routines, and programming interfaces in detail and provides additional SMP/E reference material.
z/OS SMP/E User's Guide, SA23-2277	Describes how to use SMP/E to install programs and service.

For information about z/OS publications, see *z/OS Information Roadmap*.

© Copyright IBM Corp. 1986, 2026

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see How to send feedback to IBM.

© Copyright IBM Corp. 1986, 2026

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy).

Summary of changes for z/OS 3.2

The following content is new, changed, or no longer included in z/OS 3.2.

New

The following content is new.

October 2025 refresh

- In support of APAR IO29474 (www.ibm.com/support/pages/apar/IO29474), "Authorizing use of unsigned GIMZIP packages" on page 54 and "Deprecate SHA-1 hash validation (IO29474)" on page 189 are added. (APAR IO29474, which applies to z/OS 3.2, 3.1, and 2.5)

September 2025 release

- None.

Changed

The following content is changed.

September 2025 release

- None.

Deleted

The following content is deleted.

September 2025 release

- None.

Summary of changes for SMP/E Version 3 Release 7 in z/OS 3.1

The following content is new, changed, or no longer included in z/OS 3.1.

New

The following content is new.

September 2023 release

- None.

Changed

The following content is changed.

© Copyright IBM Corp. 1986, 2026

May 2025 refresh

- “Adding certificate authority certificates” on page 96 is updated.

January 2025 refresh

- These sections are updated to reflect the new root CA certificate for the IBM Automated Delivery Request server.
  - “Adding certificate authority certificates” on page 96
  - “Displaying certificate authority certificates” on page 95
  - “Connecting the certificates to the key ring” on page 97
  - “Debugging key ring and certificate issues” on page 98
  - “Defining the CLIENT input for RECEIVE ORDER” on page 100
  - “Options that affect HTTPS operations” on page 108
- These sections are updated to reflect the current releases of Java™.
  - “Options that affect Java” on page 100
  - “Summary” on page 102
  - “HTTPS Fast Path!” on page 105
  - “Options that affect HTTPS operations” on page 108
  - “Example” on page 112

September 2023 release

- None.

Deleted

The following content is deleted.

May 2025 refresh

- Information about preventive service planning (PSP) buckets is deleted because PSP buckets for many IBM products, including z/OS 2.5 and 3.1, are no longer updated. For more information, see the following IBM Support document: PSP bucket information for IBM Z products (www.ibm.com/support/pages/node/7127792).

September 2023 release

- None.

Summary of changes for z/OS SMP/E User's Guide Version 3 Release 7 in z/OS Version 2 Release 5 (V2R5)

New

The following content is new.

June 2023

Updates were made to reflect IBM's software download servers' new CA root certificate. For more information, refer to “Adding certificate authority certificates” on page 107, “Adding certificate authority certificates” on page 96 and “Enabling certificate authority certificates” on page 106.

March 2023

Updates were made to the following topic: Chapter 6, “Preparing to verify signatures for GIMZIP packages,” on page 113

February 2023

For APAR IO28360, a super topic was added on how to prepare in order to verify signatures for GIMZIP packages. For more information, refer to Chapter 6, “Preparing to verify signatures for GIMZIP packages,” on page 113.

April 2022

The description for ++JCLIN MCS was updated, refer to “The ++JCLIN MCS” on page 173 for more information.

September 2021

IBM's download servers are updated for TLS 1.2 support. For more information on the changes and how to enable TLS 1.2 in the z/OS Communications Server FTP client program, refer to Chapter 5, “Preparing for secure Internet delivery,” on page 105

Changed

The following content is changed.

- None.

Deleted

The following content is deleted.

September 2021

- Removed Note from topic “Installing SYSMODs in the distribution libraries: ACCEPT” on page 48.

Summary of changes for z/OS SMP/E User's Guide Version 3 Release 7 in z/OS Version 2 Release 4 (V2R4)

New

The following content is new.

April 2021 refresh

As of May 1, IBM's download servers are updated for TLS 1.2 support. For more information on the changes and how to enable TLS 1.2 in the z/OS Communications Server FTP client program, refer to Chapter 5, “Preparing for secure Internet delivery,” on page 105.

Prior to April 2021 refresh

- In the "Debugging key ring and certificate issues" topic, the serial number in the sample RACDCERT command was updated to match that of the DigiCert Global Root CA certificate. For more information, refer to “Debugging key ring and certificate issues” on page 98.
- In the topic "Preparing to use Internet service retrieval", “Summary” on page 102 was updated with new Java level information and a correction for the DigiCert Global Root CA certificate.
- “SMP/E V3R7 overview” on page 189

Changed

The following content is changed.

Prior to April 2021 refresh

- The Recommended Service Upgrades topic was updated, refer to Appendix B, “Recommended service upgrade (RSU),” on page 215.
- In the topic "Preparing to use Internet service retrieval", the "Enabling certificate authority certficates" topic was changed to "Displaying certificate authority certificates", refer to “Displaying certificate authority certificates” on page 95 for the updated information.

-

Deleted

The following content is deleted.

- None.

Chapter 1. SMP/E primer

This chapter provides an introduction to SMP/E to new SMP/E users. If you are already familiar with SMP/E, you can skip this chapter.

What is SMP/E, and why should I use it?

SMP/E is a tool designed to manage the installation of software products on your z/OS system and to track the modifications you make to those products. Usually, it is the system programmer's responsibility to ensure that all software products and their modifications are properly installed on the system. The system programmer also has to ensure that all products are installed at the proper level so all elements of the system can work together. At first, that might not sound too difficult, but as the complexity of the software configuration increases, so does the task of monitoring all the elements of the system. To better understand this, let's take a closer look at your z/OS system and see how SMP/E can help you maintain it.

Understanding your system

Your z/OS system might appear to be one large block of code that drives your CPU. Actually, z/OS is a complex system comprising many different smaller blocks of code. Each of those smaller blocks of code perform a specific function within the system.

For example, some of the functions that can appear in a z/OS system include:

- Base Control Program (BCP)
- C/C++ IBM Open Class® Library
- z/OS Communications Server
- Cryptographic Services
- DFSMSdfp
- DFSORT
- Distributed File Service
- Hardware Configuration Definition (HCD)
- High Level Assembler (HLASM)
- IBM HTTP Server
- Infoprint Server
- ISPF
- JES2 or JES3
- z/OS Language Environment®
- Network File System
- Open Systems Adapter/Support Facility (OSA/SF)
- Resource Measurement Facility (RMF)
- System Display and Search Facility (SDSF)
- SMP/E
- Time Sharing Option/Extensions (TSO/E)
- z/OS UNIX System Services (z/OS UNIX)

Each system function is composed of one or more load modules. In a z/OS environment, a load module represents the basic unit of machine-readable, executable code. Load modules are created by combining one or more object modules and processing them with a link-edit utility. The link-editing of modules is

© Copyright IBM Corp. 1986, 2026

a process that resolves external references and addresses. The functions on your system, therefore, are one or more object modules that were combined and link-edited.

To see where the object modules come from, llook at the example in Figure 1 on page 2.

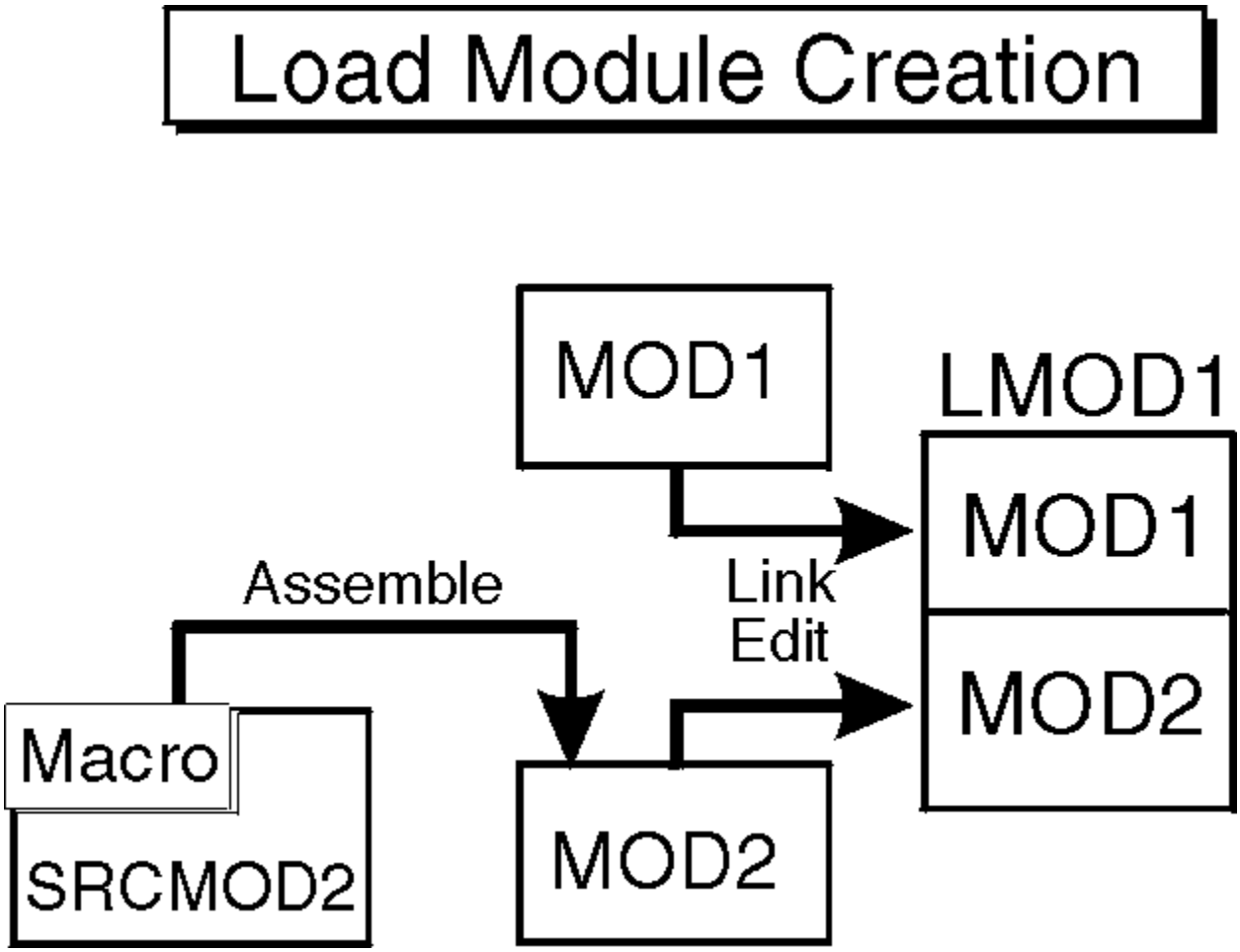


Figure 1. Creating load modules

Most of the time, object modules are sent to you as part of a product. In this example, the object module MOD1 was sent as part of the product. Other times, you might need to assemble source code sent to you by product packagers to create the object module. You can modify the source code and then assemble it to produce an object module. In the example, SRCMOD2 is source code that you assemble to create object module MOD2. When assembled, you link-edit object module MOD2 with object module MOD1 to form the load module LMOD1.

In addition to object modules and source code, most products distribute many additional parts, such as macros, help panels, dialog elements, and other z/OS library members. These modules, macros, and other types of data and code are the basic building blocks of your system. All of these building blocks are called *elements*.

Changing the elements of the system

Over time, you may need to change some of the elements of your system. These changes may be necessary to improve the usability or reliability of a product. You may want to add some new functions to your system, upgrade some of the elements of your system, or modify some elements for various reasons. In all cases, you are making system modifications. In SMP/E, we refer to these system modifications as SYSMODs.

A SYSMOD is the actual package containing information SMP/E needs to install and track system modifications. SYSMODs are composed of two parts:

- Modification control statements (MCS), designated by ++ as the first two characters, that tell SMP/E:
  - What elements are being updated or replaced
  - How the SYSMOD relates to product software and other SYSMODs
  - Other specific installation information
- Modification text, which is the object modules, macros, and other elements supplied by the SYSMOD

There are four different categories of SYSMODs, each supporting a task you might want to perform:

Function SYSMODs
Introduce the elements for a product.

PTF (program temporary fix) SYSMODs
Prevent or fix problems with an element, or introduce new element s.

APAR (authorized program analysis reports) SYSMODs
Fix problems with an element.

USERMOD (user modifications) SYSMODs
Customize an element.

Introducing an element—the function SYSMOD

One way you can modify your system is to introduce new elements into that system. To accomplish this with SMP/E, you can install a function SYSMOD. The function SYSMOD introduces a new product, a new version or release of a product, or updated functions for an existing product into the system. All other types of SYSMODs are dependent upon the function SYSMOD, because they are all modifications of the elements originally introduced by the function SYSMOD.

When we refer to installing a function SYSMOD, we are referring to the placing of all the product's elements in the system data sets, or *libraries*. Examples of these libraries are SYS1.LPALIB, SYS1.MIGLIB, and SYS1.SVCLIB. Figure 2 on page 4 depicts the process of creating executable code in the production system libraries.

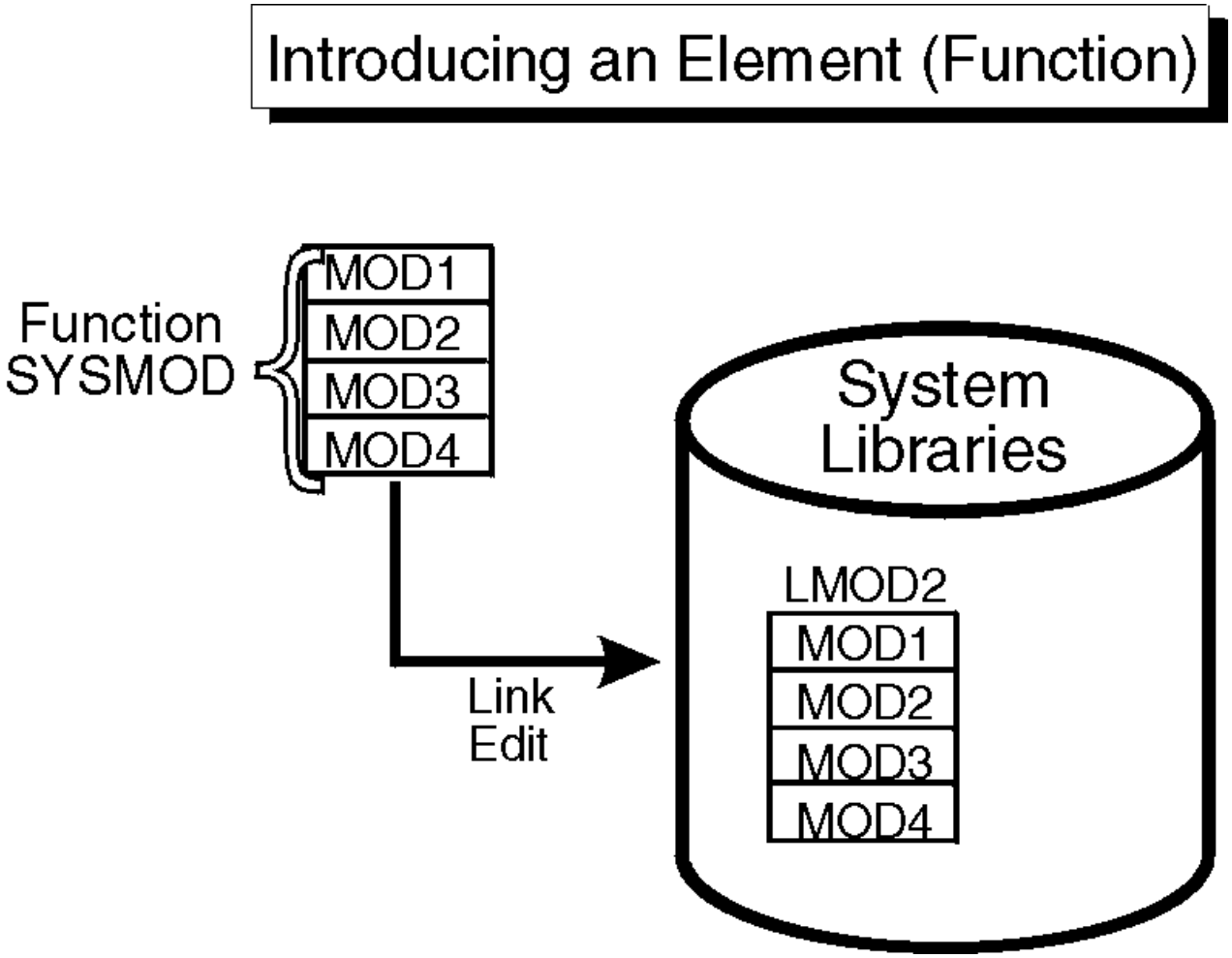


Figure 2. Introducing an element

In this figure, the installation of a function SYSMOD link-edits object modules MOD1, MOD2, MOD3, and MOD4 to create load module LMOD2. The executable code created in load module LMOD2 is installed in the system libraries through the installation of the function SYSMOD.

There are two types of function SYSMODs:

- A base function SYSMOD adds or replaces an entire system function. Examples of base functions are SMP/E and JES2.
- A dependent function SYSMOD provides an addition to an existing system function. It is called dependent because its installation depends upon a base function already being installed. Examples of dependent functions are the language features for SMP/E.

Both base function SYSMODs and dependent function SYSMODs are used to introduce new elements into the system.

Here's an example of a simple function SYSMOD that introduces four elements:

```
++FUNCTION(FUN0001)         /* SYSMOD type and identifier.   */.
++VER(Z038)                 /* For MVS SREL                  */.
++MOD(MOD1)  RELFILE(1)     /* Introduce this module         */
            DISTLIB(AOSFB)  /* in this distribution library. */.
++MOD(MOD2)  RELFILE(1)     /* Introduce this module         */
            DISTLIB(AOSFB)  /* in this distribution library. */.
++MOD(MOD3)  RELFILE(1)     /* Introduce this module         */
            DISTLIB(AOSFB)  /* in this distribution library. */.
++MOD(MOD4)  RELFILE(1)     /* Introduce this module         */
            DISTLIB(AOSFB)  /* in this distribution library. */.
```

Preventing or fixing problems with an element—the PTF SYSMOD

When a problem with a software element is discovered, IBM supplies its customers with a tested fix for that problem. This fix comes in the form of a program temporary fix (PTF). Although you may not have experienced the problem the PTF is intended to prevent, it is wise to install the PTF on your system. The PTF SYSMOD is used to install the PTF, thereby preventing the occurrence of that problem on your system.

Typically, PTFs are designed to replace or update one or more complete elements of a system function. Let's look at Figure 3 on page 5.

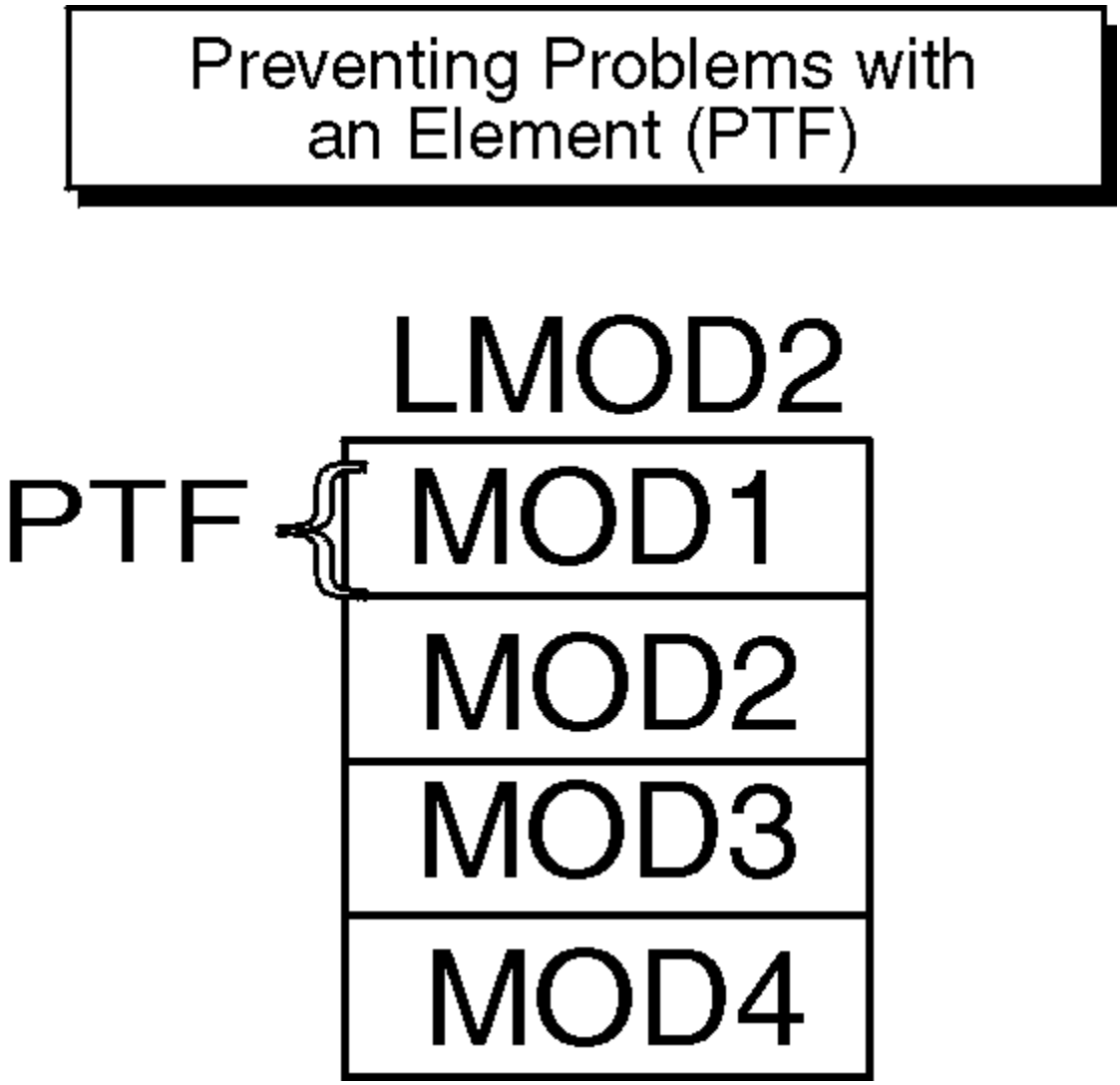


Figure 3. Preventing problems with an element

In Figure 3 on page 5, we see a previously installed load module, LMOD2. If we want to replace the element MOD1, we should install a PTF SYSMOD that contains the module MOD1. That PTF SYSMOD replaces the element in error with the corrected element. As part of the installation of the PTF SYSMOD, SMP/E relinks LMOD2 to include the new and corrected version of MOD1.

Here is an example of a simple PTF SYSMOD:

```
++PTF(PTF0001)              /* SYSMOD type and identifier.   */.
++VER(Z038) FMID(FUN0001)   /* Apply to this product.        */.
++MOD(MOD1)                 /* Replace this module           */
            DISTLIB(AOSFB)  /* in this distribution library. */.
...
... object code for module
...
```

PTF SYSMODs are always dependent upon the installation of a function SYSMOD. In some cases, some PTF SYSMODs may also be dependent upon the installation of other PTF SYSMODs. These dependencies are called prerequisites. We will look at a typical PTF prerequisite when we discuss the complexity of keeping track of the elements of the system.

Fixing problems with an element—the APAR SYSMOD

You may sometimes find it is necessary to correct a serious problem that occurs on your system before a PTF is ready for distribution. In this situation, IBM supplies you with an authorized program analysis report (APAR). An APAR is a fix designed to quickly correct a specific area of an element or replace an element in error. You install an APAR SYSMOD to implement a fix, thereby updating the incorrect element.

In Figure 4 on page 6, the shaded section shows an area of MOD2 containing an error.

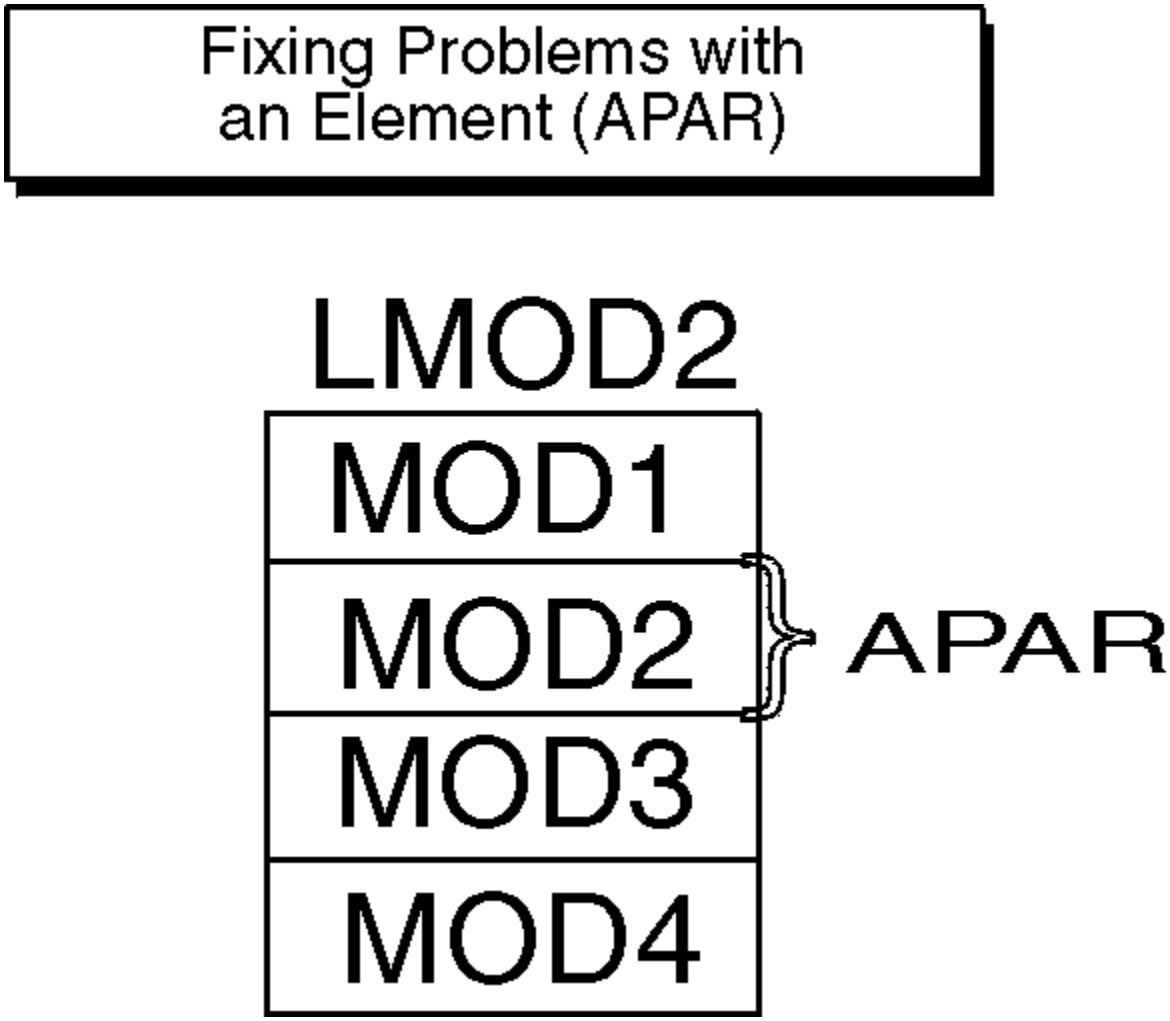


Figure 4. Fixing problems with an element

The processing of the APAR SYSMOD provides a modification for object module MOD2. During the installation of the APAR SYSMOD, MOD2 is updated (and corrected) in load module LMOD2.

Here is an example of a simple APAR SYSMOD:

```
++APAR(APAR001)             /* SYSMOD type and identifier.   */.
++VER(Z038) FMID(FUN0001)   /* Apply to this product         */
            PRE(UZ00004)    /* at this service level.        */.
++ZAP(MOD2)                 /* Update this module            */
            DISTLIB(AOSFB)  /* in this distribution library. */.
...
... zap control statements
...
```

The APAR SYSMOD always has the installation of a function SYSMOD as a prerequisite, and can also be dependent upon the installation of other PTF or APAR SYSMODs.

Customizing an element—the USERMOD SYSMOD

If you had a requirement for a product to perform differently from the way it was designed, you might want to customize that element of your system. IBM provides you with certain modules that allow you to tailor IBM code to meet your specific needs. After making the desired changes, you add these modules to your system by installing a USERMOD SYSMOD. This SYSMOD can be used to replace or update an element, or to introduce a totally new user-written element into the system. In either case, the USERMOD SYSMOD is built by you either to change IBM code or to add your own code to the system.

In Figure 5 on page 7, MOD3 was updated through the installation of a USERMOD SYSMOD.

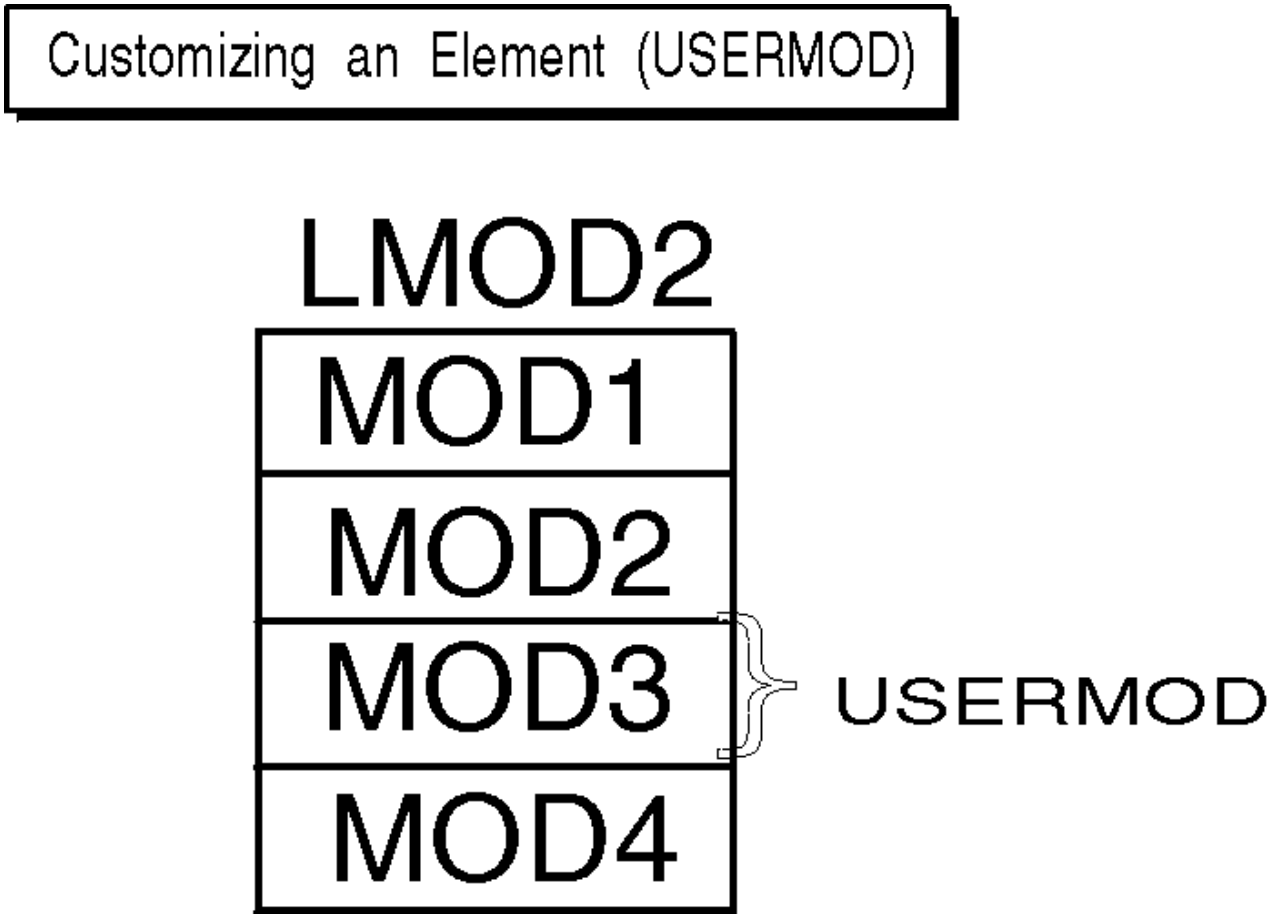


Figure 5. Customizing an element

Here is an example of a simple USERMOD SYSMOD:

```
++USERMOD(USRMOD1)             /* SYSMOD type and identifier.   */.
++VER(Z038) FMID(FUN0001)      /* Apply to this product         */
            PRE(UZ00004)       /* at this service level.        */.
++SRCUPD(JESMOD3)              /* Update this source module     */
            DISTLIB(AOSFB)     /* in this distribution library. */.
...
... update control statements
...
```

Prerequisites for USERMOD SYSMODs are the installation of a function SYSMOD, and possibly the installation of other PTF, APAR, or USERMOD SYSMODs.

SYSMOD prerequisites

As you have learned, PTF, APAR, and USERMOD SYSMODs all have the function SYSMOD as a prerequisite. In addition to their dependencies on the function SYSMOD:

- PTF SYSMODs may be dependent upon other PTF SYSMODs.
- APAR SYSMODs may be dependent upon PTF SYSMODs and other APAR SYSMODs.
- USERMOD SYSMODs may be dependent upon PTF SYSMODs, APAR SYSMODs, and other USERMOD SYSMODs.

Consider the complexity of these dependencies. When you multiply that complexity by hundreds of load modules in dozens of libraries, the need for a tool like SMP/E becomes apparent.

Let's examine the impact of these dependencies on the maintenance of software in a z/OS environment.

Keeping track of the elements of the system

The importance of keeping track of system elements and their modifications becomes readily apparent when we examine the z/OS maintenance process. Often, a PTF contains multiple element replacements. In the example in Figure 6 on page 8, PTF1 contains replacements for two modules, MOD1 and MOD2. Although load module LMOD2 contains four modules, only two of those modules are being replaced.

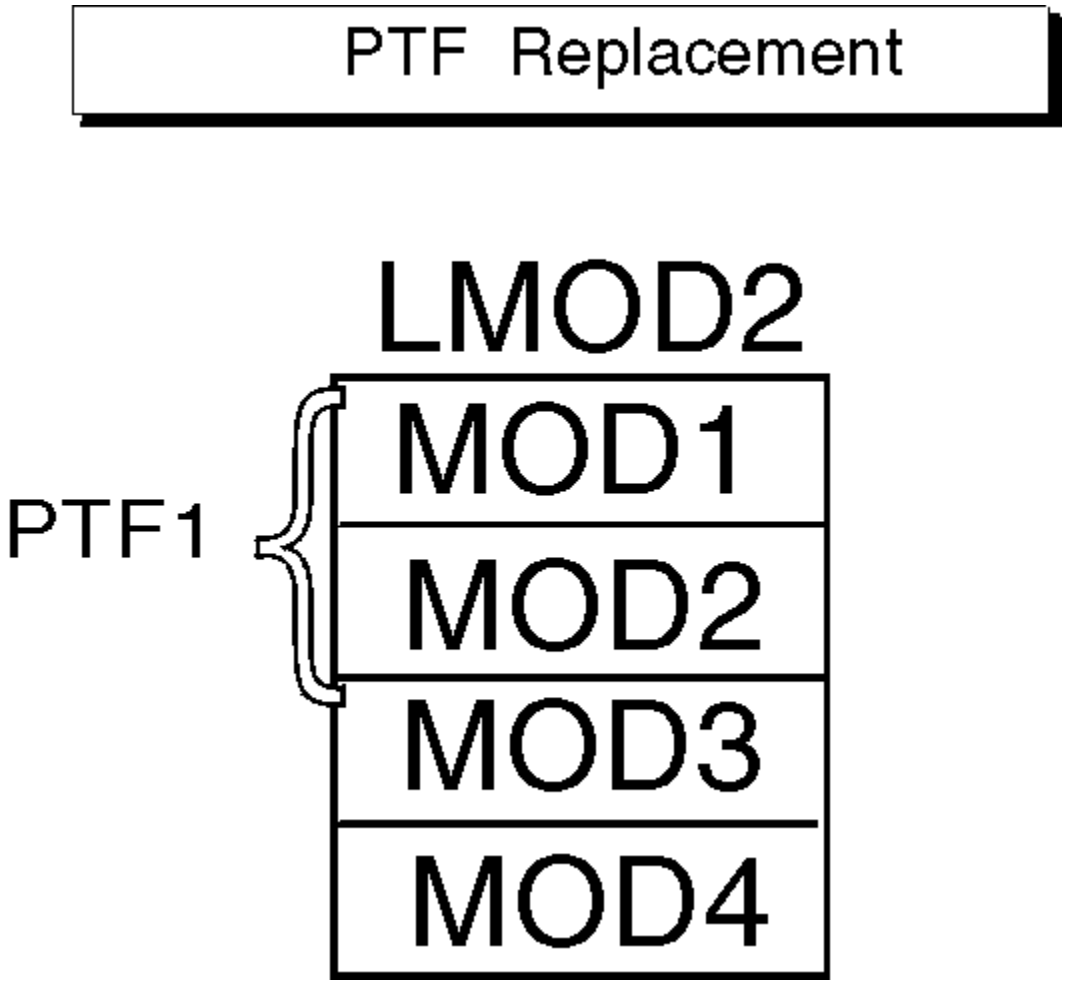


Figure 6. PTF replacement

But what happens if a second PTF replaces some of the code in a module that was replaced by PTF1? Let's look at Figure 7 on page 9.

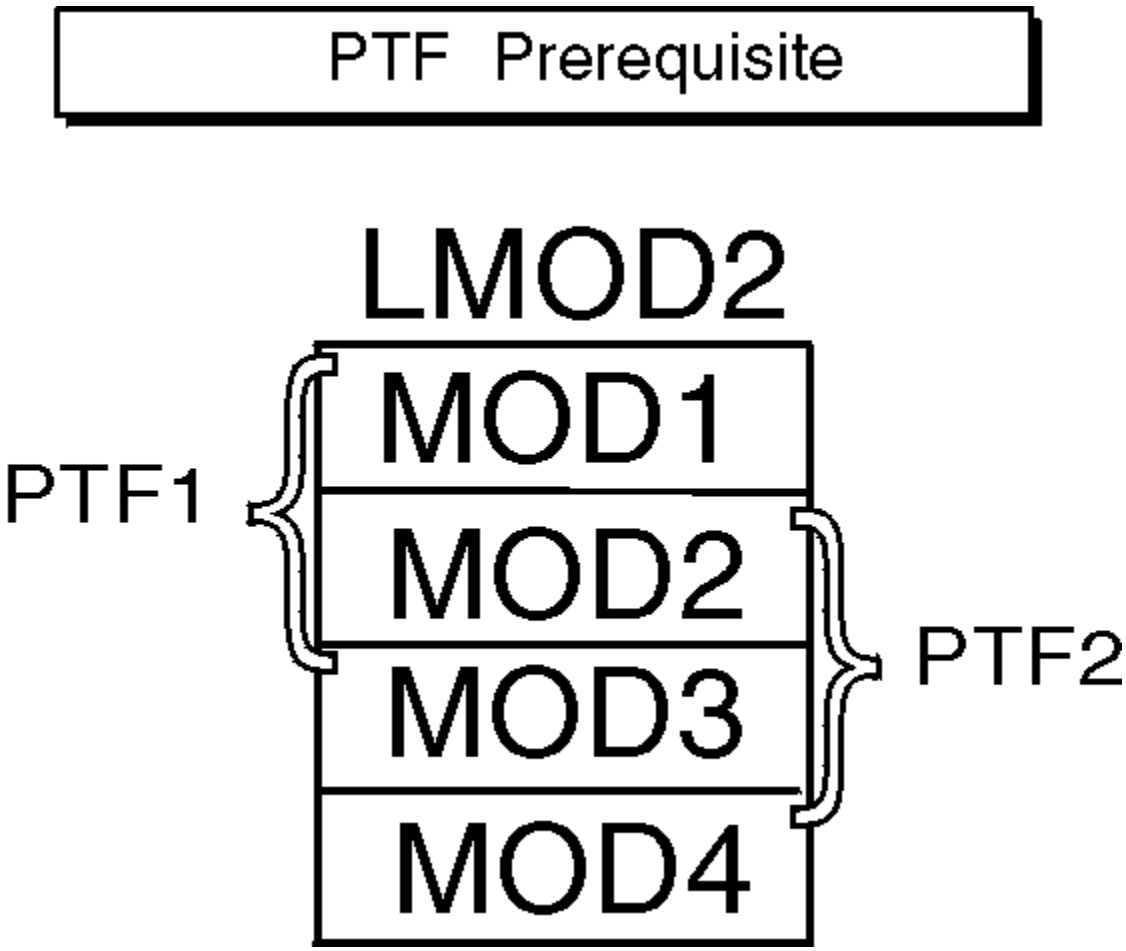


Figure 7. PTF prerequisite

In this example, PTF2 contains replacements for MOD2 and MOD3. For MOD1, MOD2, and MOD3 to interface successfully, PTF1 must be installed before PTF2. That's because MOD3 supplied in PTF2 may depend on the PTF1 version of MOD1 to be present. It is this dependency that constitutes a prerequisite. SYSMOD prerequisites are identified in the modification control statements (MCS) part of the SYSMOD package we discussed in "Changing the elements of the system" on page 2.

In addition to tracking prerequisites, there is another important reason to track system elements. The same module is often part of many different load modules. Let's take a look at the example in Figure 8 on page 10.

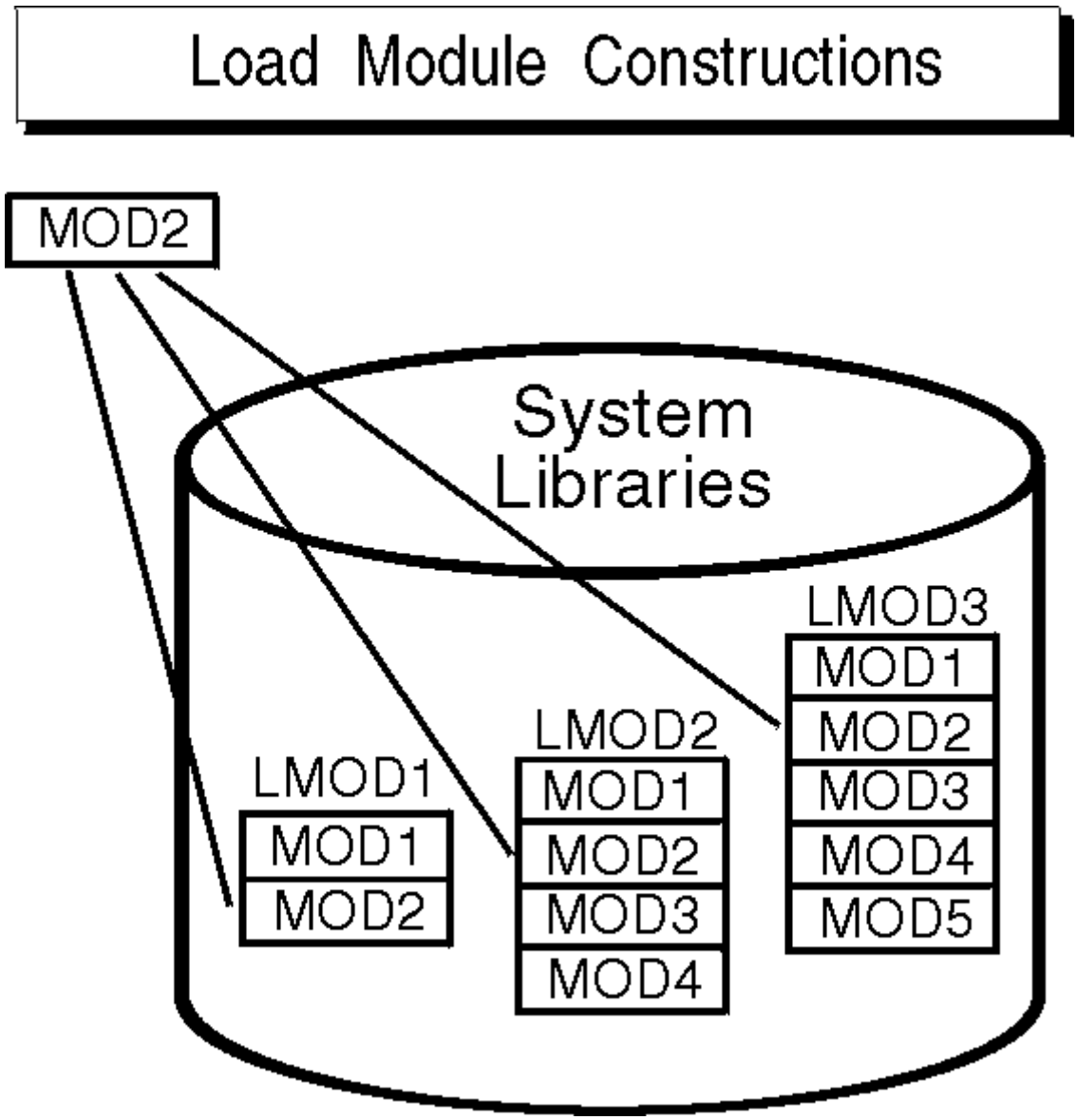


Figure 8. Load module constructions

In Figure 8 on page 10, the same MOD2 module is present in LMOD1, LMOD2, and LMOD3. When a PTF is introduced that replaces the element MOD2, that module must be replaced in all the load modules in which it exists. Therefore, it is imperative that we keep track of all load modules and the modules they contain.

You can now appreciate how complicated the tracking of system elements and their modification levels can become. Let's take a brief look at how we implement the tracking capabilities of SMP/E.

Tracking and controlling requisites

To track and control elements successfully, all elements and their modifications and updates must be clearly identified to SMP/E. SMP/E relies on modification identifiers to accomplish this. There are three modification identifiers associated with each element:

- Function modification identifiers (FMIDs) that identify the function SYSMOD that introduced the element into the system.
- Replacement modification identifiers (RMIDs) that identify the last SYSMOD (usually a PTF SYSMOD) to replace the element.

- Update modification identifiers (UMIDs) that identify the SYSMODs that have updated an element since it was last replaced.

SMP/E uses these modification identifiers to track all SYSMODs installed on your system. This ensures that they are installed in the proper sequence. Now that we realize the need for element tracking and know the types of things SMP/E tracks, let's look at how SMP/E performs its tracking function.

How does SMP/E work?

Let's review our discussion of how functions are installed into the system. We begin with elements, such as modules, macros, and source code. These elements are then processed by utilities, such as an assembler or link-editor, to create load modules. The load modules contain the machine-readable, executable code.

Your production system in a z/OS environment consists of the z/OS operating system and all the code needed to do your everyday work. That's fine, but where is all that stuff kept, and how is it organized? Let's find out.

The distribution and target libraries

To properly perform its processing, SMP/E must maintain a great deal of information about the structure, content, and modification status of the software it manages. Think of all the information SMP/E has to maintain as if it were all the information contained in the public library. To follow this analogy, let's refer to Figure 9 on page 12.

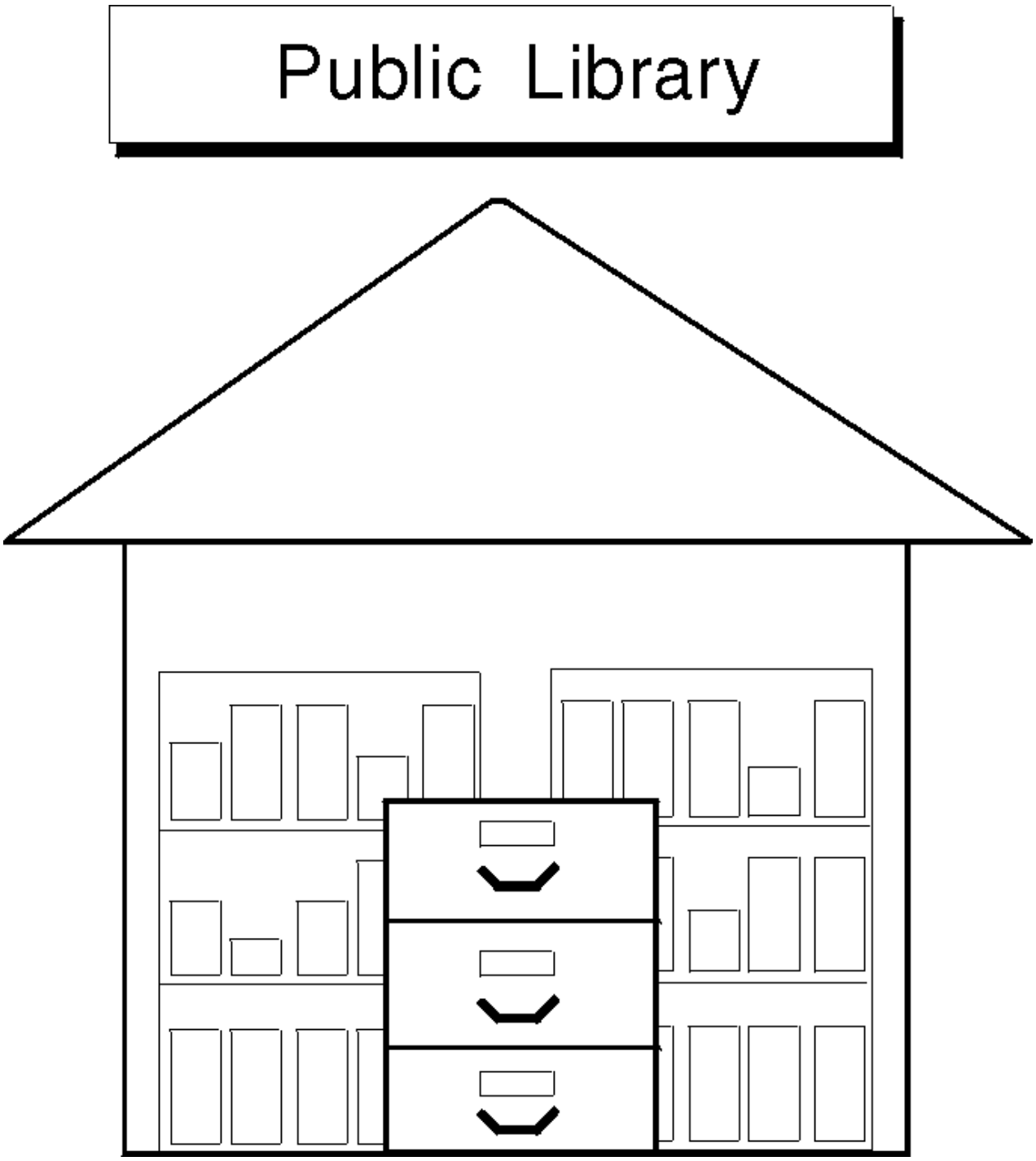


Figure 9. The public library

If you look at this figure depicting the public library, you see bookshelves filled with books and a card catalog with drawers containing a card for each book in the library. These cards contain information, such as the title, author, publishing dates, type of book, and a pointer to the actual book on the shelf.

In the SMP/E environment, there are two distinct types of bookshelves. They are referred to as the *distribution libraries* and the *target libraries*. Figure 10 on page 13 depicts these two types of SMP/E libraries.

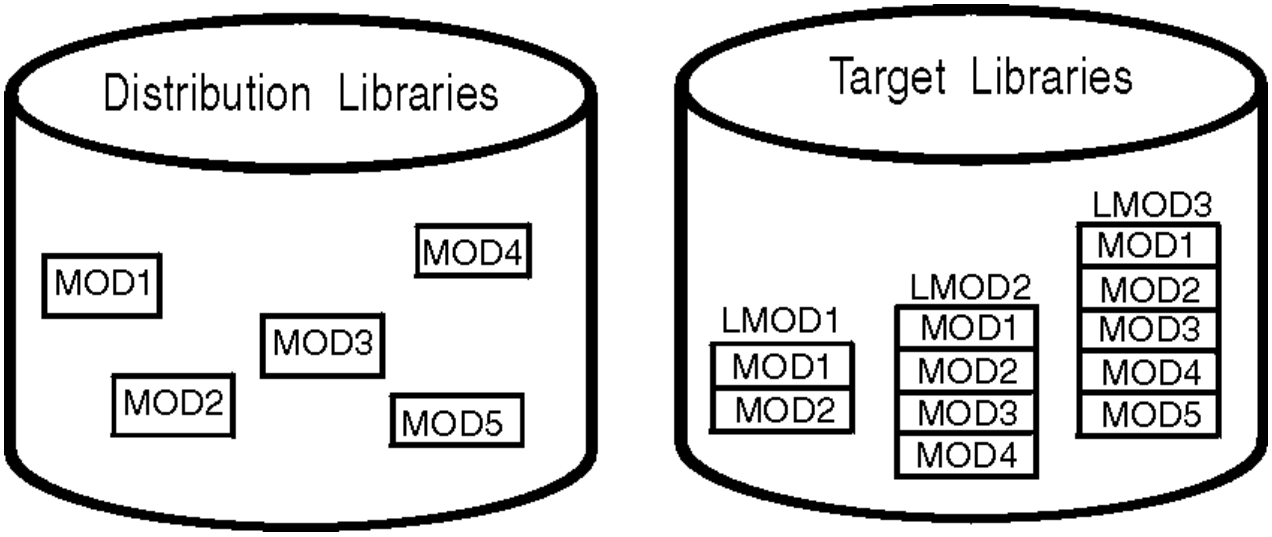


Figure 10. The distribution and target libraries

In much the same way the bookshelves in the public library hold the library books, the distribution and target libraries hold the elements of the system.

Distribution libraries contain all the elements, such as modules and macros, that are used as input for running your system. One very important use of the distribution libraries is for backup. Should a serious error occur with an element on the production system, the element can be replaced by a stable level found in the distribution libraries.

Target libraries contain all the executable code needed to run the system.

The consolidated software inventory (CSI)

As you refer to the analogy of the public library, you can see that there is one important piece of Figure 9 on page 12 that we have not yet considered. In the public library, there is a card catalog to help you find the book or piece of information you are looking for. SMP/E provides the same type of tracking mechanism in the form of the consolidated software inventory (CSI).

The CSI data sets contain all the information SMP/E needs to track the distribution and target libraries. As the card catalog contains a card for each book in the library, the CSI contains an entry for each element in its libraries. The CSI entries contain the element name, type, history, how the element was introduced into the system, and a pointer to the element in the distribution and target libraries. The CSI does not contain the element itself, but rather a description of the element it represents.

Let's see exactly how these entries are arranged in the CSI.

The SMP/E zones

The cards in the public library card catalog are arranged alphabetically by the author's last name, and by the topic and title of the book. In the CSI, entries for the elements in the distribution and target libraries are grouped according to their installation status. That is, entries representing elements found in the distribution libraries are contained in the distribution zone. Entries representing elements found in the target libraries are contained in the target zone. Both of these zones serve the same purpose as the drawers of the public library card catalog.

In addition to the distribution and target zones, the SMP/E CSI also contains a global zone. The global zone contains:

- Entries needed to identify and describe each target and distribution zone to SMP/E
- Information about SMP/E processing options
- Status information for all SYSMODs SMP/E has begun to process
- Exception data for SYSMODs requiring special handling or that are in error

In SMP/E, when we speak of exception data, we are usually referring to HOLDDATA. HOLDDATA is often supplied for a product to indicate a specified SYSMOD should be held from installation. Reasons for holding a SYSMOD can be:

- A PTF is in error and should not be installed until the error is corrected (ERROR HOLD).
- Certain system actions may be required before SYSMOD installation (SYSTEM HOLD).
- The user may want to perform some actions before installing the SYSMOD (USER HOLD).

All the information found in the global zone, combined with the information found in the distribution and target zones, represents the data SMP/E needs to install and track your system software.

Remember the picture of the public library in Figure 9 on page 12? Now look at Figure 11 on page 14.

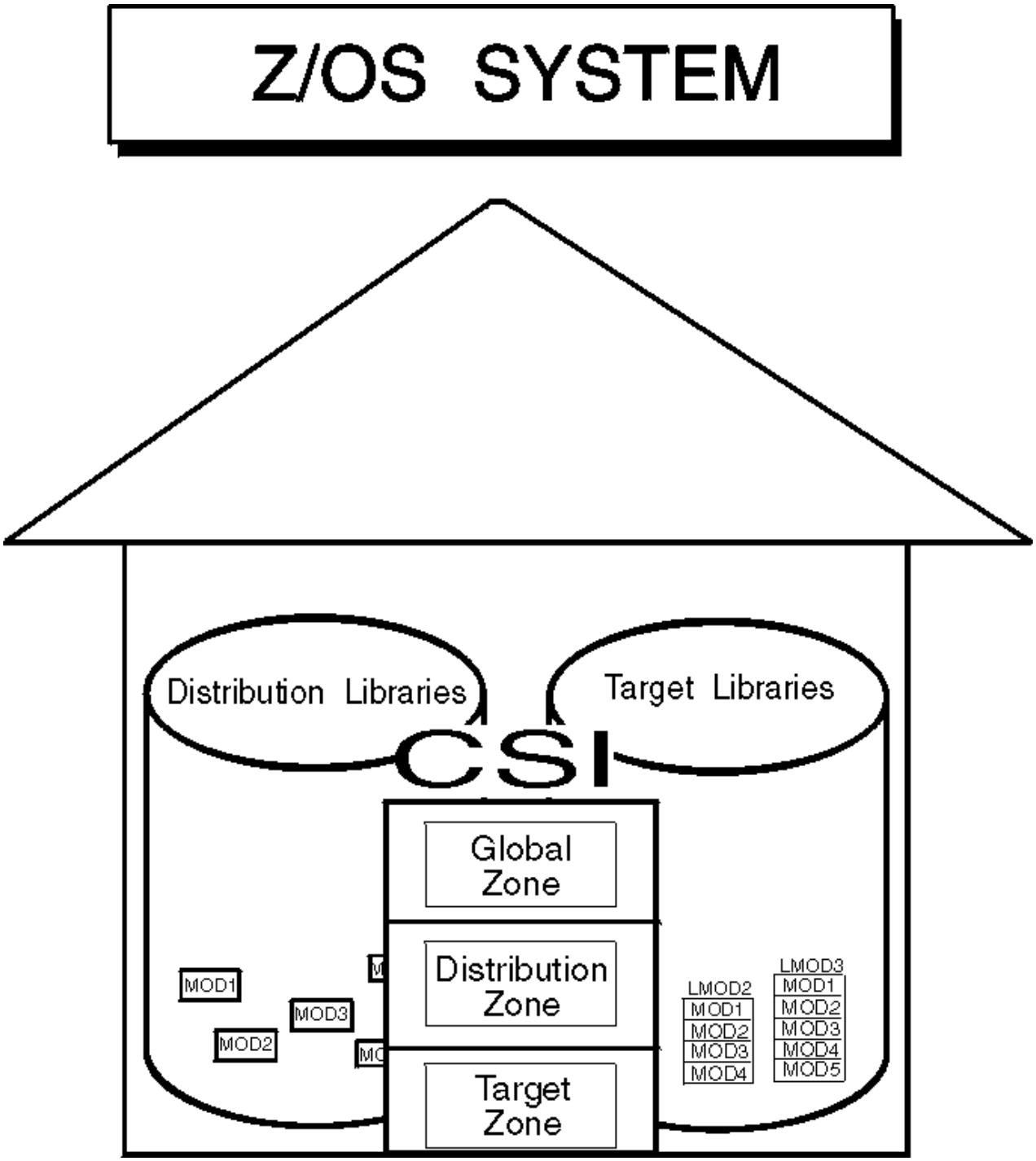


Figure 11. z/OS system with SMP/E

Now you can see how all the elements of the system fit together, and how they can be installed, modified, and tracked using SMP/E.

What are the basic SMP/E commands I need to know?

Now that you are familiar with SMP/E and what it can do, you are probably wondering what you need to know to get started using SMP/E. Let's take a look at the basic processing commands you need to know to use SMP/E.

Setting the zone you want to work on

Before processing SMP/E commands, you must first set the zone on which you want SMP/E to work (global, target, or distribution). You do this by issuing the SET command. The SET command identifies the zone and, therefore, the libraries, upon which subsequent SMP/E commands are to act.

The SET command can also be used to request a particular set of predefined processing options. For more information about the SET command, see The SET command in *z/OS SMP/E Commands*.

Receiving the SYSMOD into SMP/E's data sets

For SMP/E to install a SYSMOD, the SYSMOD must be “received” into data sets that can be used by SMP/E. The SMP/E RECEIVE command performs the task of copying the SYSMOD from the distribution medium from which it was sent into the data sets used by SMP/E.

For more information about the RECEIVE command, refer to “Receiving the SYSMOD into SMP/E's data sets” on page 18.

Applying the SYSMOD to the target libraries

Once a SYSMOD has been received, you want to “apply” the SYSMOD to the appropriate target libraries. The SMP/E APPLY command invokes various system utilities to install the SYSMOD's elements into the target libraries.

For more information about the APPLY command, refer to “Applying the SYSMOD to the target libraries” on page 22.

Restoring the target libraries to the previous level

Should you experience problems after applying a SYSMOD, you may want to “restore” its elements in error to a previous and stable level. The SMP/E RESTORE command replaces a failing element with a copy from the distribution libraries.

For more information about the RESTORE command, refer to “Restoring the target libraries to the previous level” on page 26.

Accepting the SYSMOD and updating the distribution libraries

After you have performed a SYSMOD RECEIVE and APPLY, you want to “accept” the elements into the distribution libraries for backup. However, this should be done only after you are satisfied with the performance and stability of the elements of the SYSMOD. Once you ACCEPT a SYSMOD, you cannot RESTORE its element to a previous level. The SMP/E ACCEPT command updates the distribution libraries so they are available for backup of any future SYSMODs.

For more information about the ACCEPT command, refer to “Accepting the SYSMOD into the distribution libraries” on page 30.

Displaying SMP/E data

The SMP/E CSI and other primary data sets contain a great deal of information you may find useful when installing new elements or functions, preparing user modifications, or debugging problems. You can

display that information, as well as information about modules, macros, and other elements, in several different ways.

- Query dialogs display specific information you request through interactive dialogs with SMP/E.
- The LIST command generates a hardcopy listing of information about your system.
- REPORT commands check, compare, and generate hardcopy information about the contents of zones on your system.
- The SMP/E CSI application programming interface can be used to write application programs to query the contents of your system's CSI data sets.

For more information about displaying SMP/E data, refer to “Displaying SMP/E data” on page 34.

Flow of SMP/E SYSMOD processing

To see the flow of SMP/E SYSMOD processing for the RECEIVE, APPLY, RESTORE, and ACCEPT commands, let's look at Figure 12 on page 17.

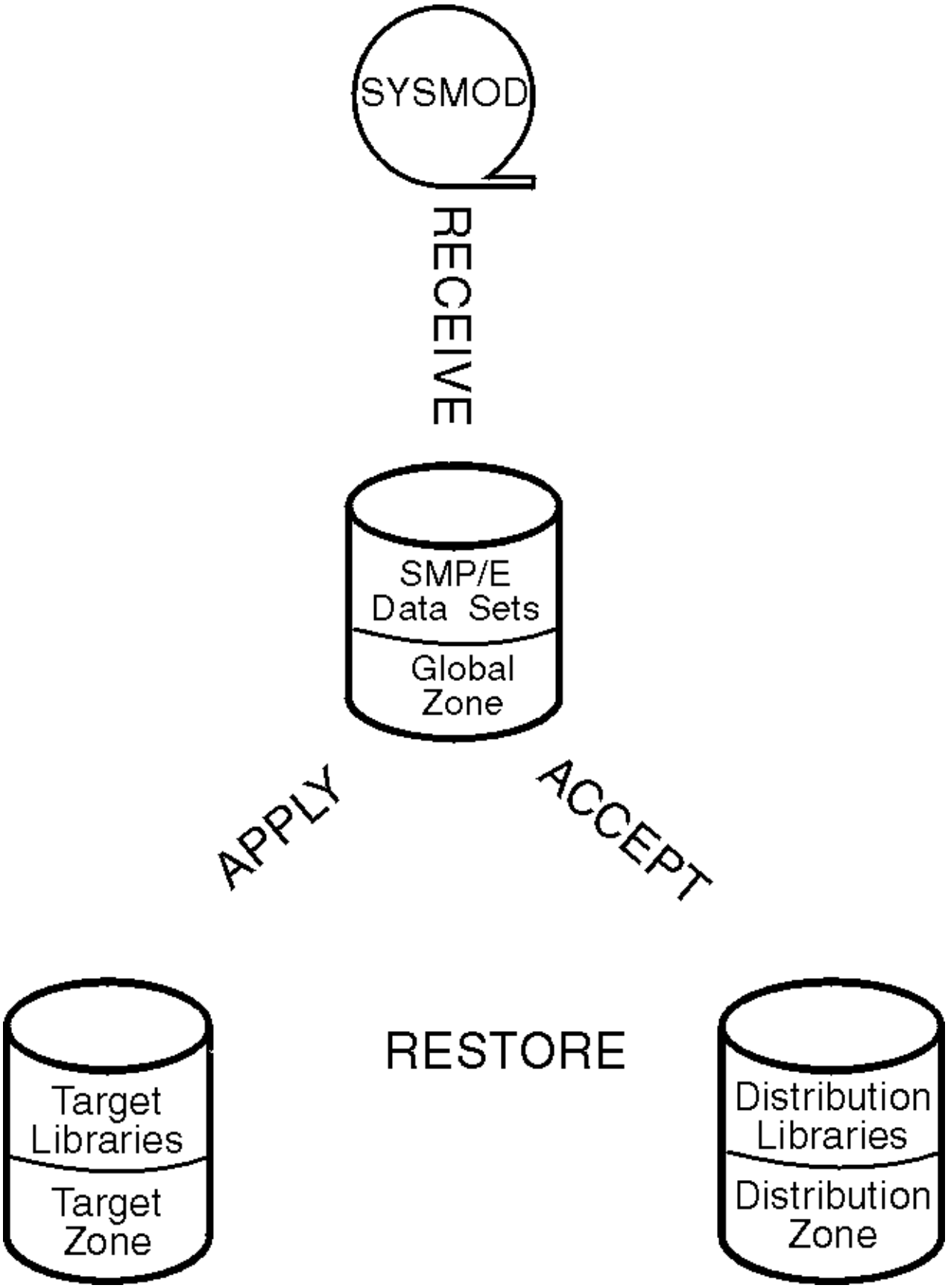


Figure 12. Flow of SMP/E SYSMOD processing

Receiving the SYSMOD into SMP/E's data sets

To initiate SMP/E processing, you must first install the software into SMP/E data sets. You can use the RECEIVE command to load the SYSMOD information from the distribution medium into the SMPPTS and SMPTLIB data sets for later installation of the SYSMODs.

In this chapter, you will learn about those data sets and the following topics:

- "What happens during RECEIVE processing" on page 18
- "What happens during internet service retrieval" on page 18
- "How SMP/E keeps track of RECEIVE processing" on page 19
- "Using the RECEIVE command" on page 20
- "Summary of the RECEIVE command" on page 22

What happens during RECEIVE processing

SMP/E knows software in terms of SYSMODs. Each SYSMOD processed by SMP/E contains two types of information:

- Instructions telling SMP/E what elements are in the SYSMOD and how to install them
- The actual element replacements or updates contained in the SYSMOD

The instructions are made up of a series of control statements called *modification control statements* (MCSs). The element replacements or updates can be packaged in several ways:

- The **RELFILE** method packages the elements in relative files that are separate from the MCSs. This method is used mostly for function SYSMODs. (The examples in the remainder of this book assume that function SYSMODs are packaged in RELFILE format.)
- The **inline** method packages the elements immediately following the associated MCSs.
- The **indirect library** method packages elements in DASD data sets that are separate from the MCSs.

For more details about packaging, see Standard Packaging Rules for z/OS-based Products (publibz.boulder.ibm.com/epubs/pdf/gimpkg80.pdf).

During RECEIVE processing, the MCS for each SYSMOD is copied to an SMP/E temporary storage area called the *SMPPTS data set*. The MCS entry contains the MCS and any inline element replacements or updates for the SYSMOD. Relative files, however, are stored in another temporary storage area called the SMPTLIB data sets.

We briefly mentioned HOLDDATA in "The SMP/E zones" on page 13. HOLDDATA is processed by the RECEIVE command and is stored for use later during installation of the affected SYSMODs.

What happens during internet service retrieval

Internet Service Retrieval uses the SMP/E RECEIVE ORDER command to order software service directly from IBM. With RECEIVE ORDER processing, you run an SMP/E job that places the service order directly with the IBM Automated Delivery Request server, waits for the order to be fulfilled, and then downloads and processes the service package contents.

You can use the RECEIVE ORDER command to request HOLDDATA or PTF service orders based on criteria you specify. When you request HOLDDATA only, you receive the last two years of Enhanced HOLDDATA for the entire z/OS platform. You can request two types of PTF service orders:

Corrective
: You can order PTFs that resolve specific APARS. The package resulting from such an order is tailored to your SMP/E environment and contains the PTFs you requested plus any requisite PTFs not already present in your environment.

Preventive
: You can order all currently available PTFs that meet your selection criteria. The package resulting from such an order is tailored to your SMP/E environment and contains the PTFs that match your selection

criteria plus any requisite PTFs not already present in your environment. There are three selection criteria:

Critical
Includes all available PTFs that resolve high impact pervasive (HIPER) problems or PTFs in error (PE).

Recommended
Includes all available PTFs identified with Recommended Service Update SOURCEID (RSUyymm) and all available PTFs that resolve a critical problem (HIPER or PE). Recommended service includes PTFs through the most recent RSU level.

All
Includes all available PTFs.

All PTF packages also contain the last two years of Enhanced HOLDDATA for the entire z/OS software platform.

Using the RECEIVE ORDER command, you can automate the service update process. For example, you can have an SMP/E job run every night at 1:00 AM to order and download the latest HOLDDATA and critical service, so the information is available locally and ready for use every morning.

How SMP/E keeps track of RECEIVE processing

SMP/E updates the global zone with information about the SYSMODs that have been received:

SYSMOD entries
SYSMOD entries are created in the global zone for each SYSMOD that has been received.

HOLDDATA entries
HOLDDATA entries are created in the global zone for each ++HOLD statement that has been received. HOLDDATA entries identify SYSMODs that should be held back from being installed because they require special handling or are in error.

Figure 13 on page 20 shows what you have learned about RECEIVE processing.

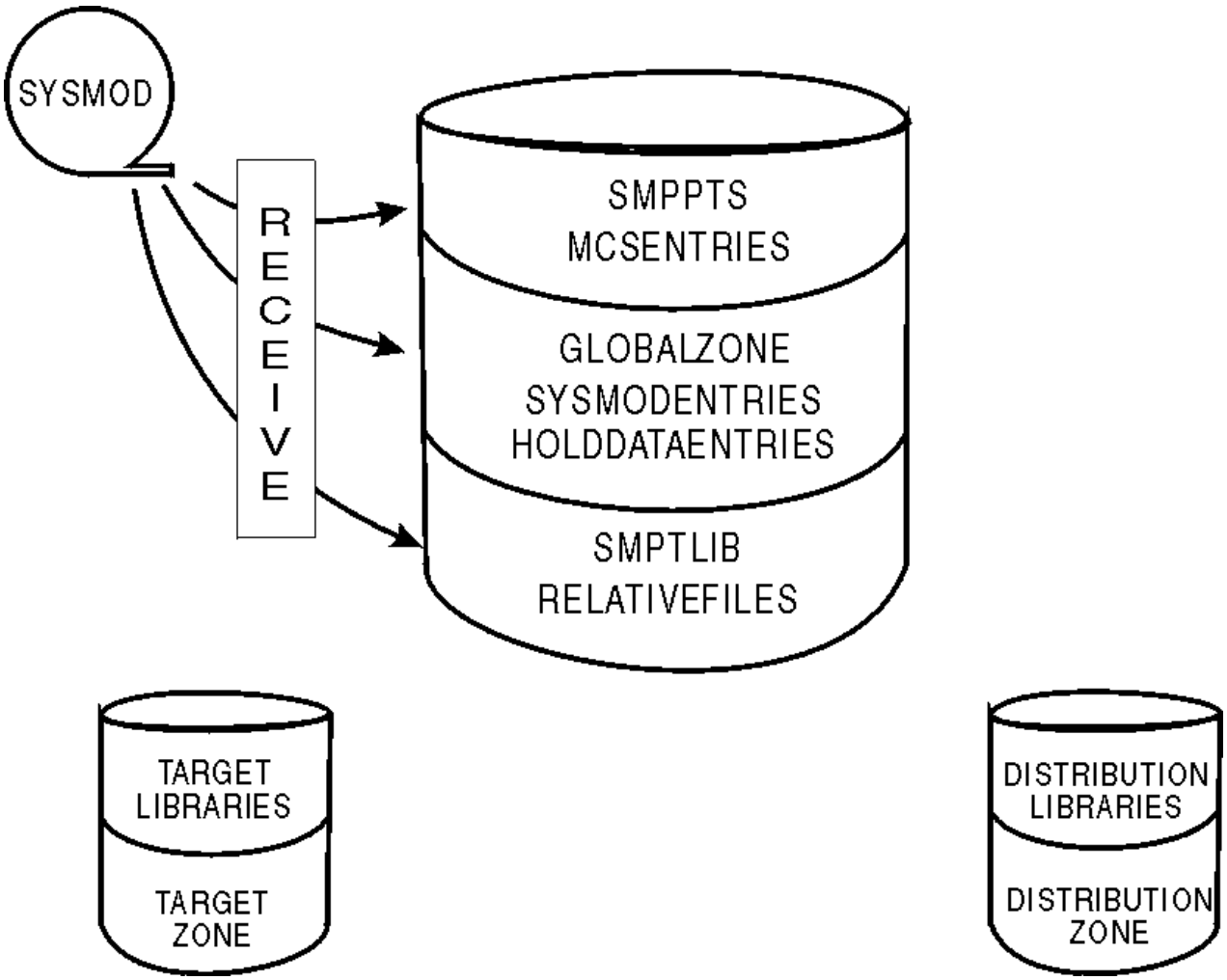


Figure 13. Results of RECEIVE processing

Using the RECEIVE command

In this section, you will see some basic examples of how you might use the RECEIVE command.

Examples

Let's look at a few of these examples.

Receiving SYSMODs and HOLDDATA

In the course of maintaining your system, you need to install service and process the related HOLDDATA. Assume IBM has supplied you with a service tape (such as a CBPDO package or an ESO tape), and you want to install it on your system. The first step is to receive the SYSMODs and HOLDDATA that are contained on the tape. You can accomplish this by specifying the following commands:

```
SET      BDY(GLOBAL).
RECEIVE.
```

When you issue these commands, SMP/E receives all the SYSMODs and HOLDDATA on the service tape into the global zone.

Receiving only HOLDDATA

There may be times when you do not want to receive the SYSMODs from a service tape, but you do want to receive the HOLDDATA. Because the HOLDDATA provides information about SYSMODs requiring

special handling or that are in error, it is important for you to receive the HOLDDATA into SMP/E's storage repository as soon as possible. The following commands process only the HOLDDATA:

```
SET      BDY(GLOBAL).
RECEIVE  HOLDDATA.
```

By issuing these commands, you direct SMP/E to receive only the HOLDDATA from the service tape into the global zone.

Receiving only SYSMODs

Assume you have previously received only the HOLDDATA from a service tape and are now ready to install the SYSMODs. To install these SYSMODs (using the APPLY and ACCEPT commands), you must first receive them. This can be done by specifying the following commands:

```
SET      BDY(GLOBAL).
RECEIVE  SYSMODS.
```

When you issue these commands, you direct SMP/E to receive only the SYSMODs from the service tape into the global zone.

Receiving SYSMODs and HOLDDATA for a specific product

You may want to receive SYSMODs and HOLDDATA for a particular product from the service tape. You can accomplish this task by specifying the following commands:

```
SET      BDY(GLOBAL).
RECEIVE  FORFMID(HOP0001).
```

By issuing these commands, you direct SMP/E to receive SYSMODs and HOLDDATA for the product whose FMID is HOP0001 from the service tape into the global zone.

Requesting a new PTF order with the RECEIVE ORDER command

Assume that you want to order two specific PTFs (UQ12345 and UQ98765). You can accomplish this task by specifying the following SMP/E job:

```
//jobname   JOB ...
//RECEIVE   EXEC PGM=GIMSMP
//SMPCSI    DD DSN=SMPE.GLOBAL.CSI,DISP=SHR
//SMPNTS    DD PATH='/u/smpe/smpnts/',PATHDISP=KEEP
//SMPOUT    DD SYSOUT=*
//SMPRPT    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SMPCNTL   DD *
  SET       BOUNDRY(GLOBAL).
  RECEIVE   SYSMODS HOLDDATA
            ORDER(                    /* Place an order for service */
              ORDERSERVER(ORDRSRVR)
              CONTENT(
                PTFS(UQ12345,UQ98765) /* Get these PTFs, and any.. */
                      )               /* ..requisites..            */
              FORTGTZONES(ZOS14)      /* ..for this target zone    */
                 ).
/*
//ORDRSRVR  DD *
  <ORDERSERVER
   url="https://eccgw01.boulder.ibm.com/services/projects/ecc/ws/"
   keyring="MRWKYRNG"
   certificate="SMPE Client Certificate">
  </ORDERSERVER>
/*
```

Note: An alternative URL for the IBM Automated Delivery Request server is `https://eccgw02.rochester.ibm.com/services/projects/ecc/ws`.

In addition to receiving the specified PTFs, you receive any requisites for these PTFs that are not already present in the ZOS14 target zone.

For a more complete description of all the RECEIVE command operands and other examples, see The RECEIVE command in *z/OS SMP/E Commands*.

Reporting output

When RECEIVE processing is complete, these reports will help you analyze the results:

- The RECEIVE summary report provides you with an at-a-glance look at all the SYSMODs that were processed during the RECEIVE command run. It shows you which SYSMODs were received, which were not received, and why.

  The SYSMODs listed in this report depend on the operands you specify on the RECEIVE command.
- The RECEIVE exception SYSMOD Data report provides you with a quick summary of the HOLDDATA information processed during the RECEIVE command run. It lists the SYSMODs requiring special handling or that are in error, and those SYSMODs no longer requiring special handling or that have had an error fixed.
- The File allocation report provides you with a list of the data sets used for RECEIVE processing and supplies information about these data sets.

For more information about these reports (and samples of actual reports), see SMP/E reports in *z/OS SMP/E Commands*.

Summary of the RECEIVE command

Let's summarize what you have learned about using the RECEIVE command to load a SYSMOD into SMP/E's storage area. The RECEIVE command:

- Copies the MCS for each SYSMOD to the SMPPTS data set
- Loads elements into SMPTLIB data sets for SYSMODs using the relative file packaging method
- Records what is received in the global zone
  - SYSMOD entries
  - HOLDDATA entries
- Reports the results of processing

The RECEIVE ORDER command:

- Places the service order directly with the IBM Server
- Waits for the order to be fulfilled
- Downloads and processes the service package contents.

Applying the SYSMOD to the target libraries

After the SYSMODs have been received, you can use the APPLY command to install them into the appropriate target system libraries. The APPLY command calls system utilities, which are responsible for the actual updating of those libraries.

You will learn about the following topics:

- What happens during APPLY processing
- How SMP/E keeps track of APPLY processing
- Examples of using the APPLY command
- A summary of the APPLY command

What happens during APPLY processing

Throughout the APPLY process, SMP/E helps you manage the complexities of your system when installing SYSMODs.

Selecting the SYSMODs

You can specify operands on the APPLY command that tell SMP/E which of the received SYSMODs are to be selected for installation in the target libraries. SMP/E checks to make sure all other required SYSMODs (prerequisites) have been installed or are being installed concurrently and in the proper sequence. For more information about prerequisites, see "Keeping track of the elements of the system" on page 8.

Selecting the elements

During APPLY processing, SMP/E uses the information provided in the selected SYSMODs to determine which elements should be installed in the target libraries. The selection of elements is monitored by SMP/E to make sure that the correct functional level of each element is selected.

Checking the APPLY process

SMP/E provides you with an option to stop APPLY processing just before any updating takes place so you can ensure all prerequisites are satisfied before the installation of the SYSMODs. This helps you see what will happen (and helps you detect problem SYSMODs) without actually updating the target libraries.

Updating the target libraries

After the proper SYSMODs have been selected and the proper functional and service level of each element has been determined, the APPLY command directs SMP/E to call the system utilities. It is the system utilities that actually place the elements into the target libraries described in the target zone. The source of the elements is the SMPTLIB data sets, the SMPPTS data set, or the indirect libraries, depending on how the SYSMOD was packaged.

Note: Because the APPLY command updates the system libraries, you should never use it on a live production system. When you process the APPLY command, you should always use a copy of the target libraries and target zone. By using a copy, you minimize the risk of new code causing an outage of your system. This process of copying is called *cloning* and is explained in detail in the *OS/390 Software Management Cookbook*, SG24-4775.

How SMP/E keeps track of APPLY processing

SMP/E updates the information about the SYSMODs that have been applied. Remember, the target zone reflects the contents of the target libraries. Therefore, after the utility work is complete, and the target libraries have been updated, the target zone is updated to accurately reflect the status of those libraries.

- A SYSMOD entry is created in the **target zone** for each SYSMOD that has been applied. Element entries (such as MOD and LMOD) are also created in the target zone for those elements that have been installed in the target libraries.
- SYSMOD entries in the **global zone** are updated to reflect that the SYSMOD has been applied to the target zone.
- BACKUP entries are created in the **SMPSCDS data set** so the SYSMOD can later be restored, if necessary.

Figure 14 on page 24 shows what you have learned about APPLY processing.

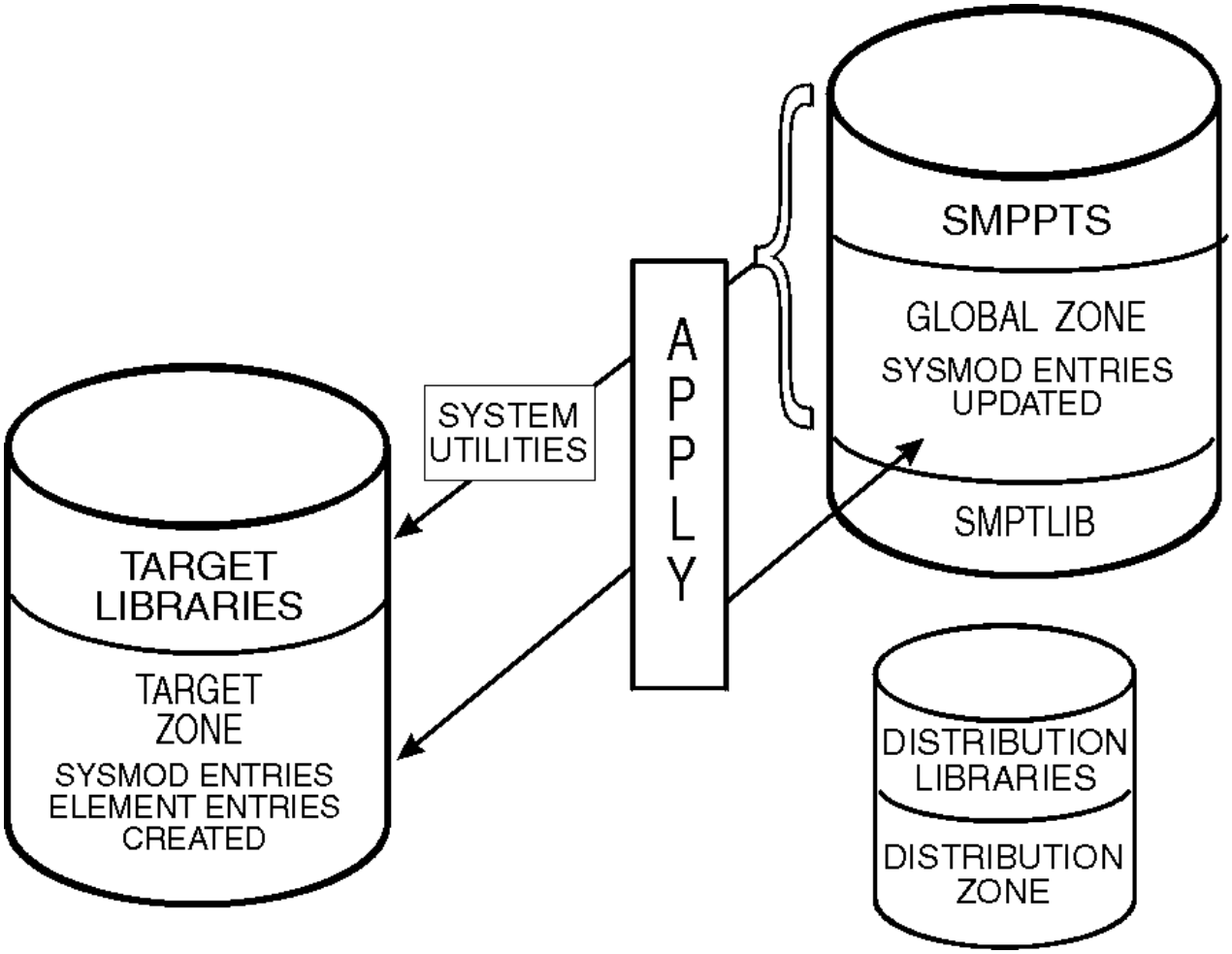


Figure 14. Results of APPLY processing

Using the APPLY command

The APPLY command has many operands that allow you great flexibility in choosing which SYSMODs you want installed in your target libraries. It also provides you with a variety of output based on the operands you specify.

Examples

Let's look at a few examples of how you might use the APPLY command.

Applying PTF SYSMODs

After you have received the SYSMODs into the global zone, you can tell SMP/E that you want to install only the PTF SYSMODs. You can do this by specifying the following commands:

```
SET      BDY(ZOSTGT1).
APPLY    PTFS.
```

By issuing these commands, you direct SMP/E to apply all eligible PTF SYSMODs to target zone ZOSTGT1.

Suppose you do not want to install all the PTF SYSMODs, but only a select few. You can do this by specifying the following commands:

```
SET      BDY(ZOSTGT1).
APPLY    SELECT(UZ00001,UZ00002).
```

Issuing these commands results in the selection of only PTFs UZ00001 and UZ00002 for installation in target zone ZOSTGT1.

Applying APAR and USERMOD SYSMODs

You may want to install just corrective fixes (APARs) or user modifications (USERMODs) into the target libraries. You can accomplish this with the following commands:

```
SET      BDY(ZOSTGT1).
APPLY    APARS
         USERMODS.
```

When you issue these commands, SMP/E installs all eligible APARs and USERMODs into target zone ZOSTGT1.

Applying SYSMODs for selected products

There may be times when you want to update only certain products on your system with the SYSMODs contained on a service tape. Assume you want to install all PTFs for a particular product to your system. This can be accomplished by specifying the following commands:

```
SET      BDY(ZOSTGT1).
APPLY    PTFS
         FORFMID(HOP0001).
```

or

```
SET      BDY(ZOSTGT1).
APPLY    FORFMID(HOP0001).
```

In both cases, SMP/E applies all applicable PTFs for the product with FMID HOP0001 to target zone ZOSTGT1. Unless you specify otherwise, PTFS is the default SYSMOD type.

Applying SYSMODs having prerequisites

When installing a SYSMOD, you might not always know if it has prerequisites, or if the prerequisites are available. (Sometimes a prerequisite SYSMOD might not be received, or it might be held because it is in error.) In cases such as this, you can direct SMP/E to check whether an equivalent (or superseding) SYSMOD is available, by specifying the GROUPEXTEND operand.

Assume you want to update a product with all the eligible PTFs and APARs. You can do this by specifying the following commands:

```
SET      BDY(ZOSTGT1).
APPLY    PTFS
         APARS
         FORFMID(HOP0001)
         GROUPEXTEND.
```

By issuing these commands, you direct SMP/E to apply all PTFs and APARs, along with any other required SYSMODs to the product whose FMID is HOP0001 and is located in the ZOSTGT1 target zone. If SMP/E cannot find a required SYSMOD, it looks for and uses a SYSMOD that supersedes the required one.

Applying SYSMODs using the CHECK operand

In the previous example, you directed SMP/E to automatically include all SYSMODs needed for the specified product. There may be times when you want to see which SYSMODs are included before you actually install them. You can do this with the CHECK operand by issuing the following commands:

```
SET      BDY(ZOSTGT1).
APPLY    PTFS
         APARS
         FORFMID(HOP0001)
         GROUPEXTEND
         CHECK.
```

After these commands are processed, you can check the SYSMOD Status report to see which SYSMODs would have been installed if you had not specified the CHECK operand. If you are satisfied with the results of this trial run, you can issue the commands again, without the CHECK operand, to actually install the SYSMODs.

For a more complete description of all the APPLY command operands, and for additional examples, see The APPLY command in *z/OS SMP/E Commands*.

Reporting output

When APPLY processing is complete, these reports will help you analyze the results:

- The SYSMOD status report provides you with a summary of the processing that took place for each eligible SYSMOD, based on the operands you specified on the APPLY command. It shows you which SYSMODs were applied, which were not applied, and why.
- The Element summary report provides you with a detailed look at each element affected by APPLY processing. It tells you in which libraries the elements were installed.
- The Causer SYSMOD summary report provides you with a list of SYSMODs that caused other SYSMODs to fail, and describes the errors that must be fixed to successfully process the SYSMODs. This report can reduce the amount of work involved in figuring out which errors caused SYSMODs to fail.
- The File allocation report provides you with a list of the data sets used for APPLY processing and supplies information about these data sets.

Additional reports may be produced depending on the work being done and the content of the SYSMODs. For more information about all the reports produced by the APPLY command (and samples of actual reports), see *z/OS SMP/E Commands*.

Summary

Let's summarize what you have learned about using the APPLY command to install a SYSMOD in the target libraries. The APPLY command:

- Selects SYSMODs to install
- Checks that all other required SYSMODs have been (or are being) installed
- Based on SYSMODs, selects elements to install
- Directs SMP/E to call the system utilities to update the target libraries
- Records what is applied:
  - Target zone: Creates SYSMOD entries and element entries
  - Global zone: Updates SYSMOD entries
  - SMPSCDS data set: Creates BACKUP entries
- Reports the results of processing

Remember, you should never perform APPLY processing on a live production system!

Restoring the target libraries to the previous level

If you discover that a particular SYSMOD is causing a problem in your target libraries, you can remove it and replace the elements affected by it with the previous level of those elements, which is obtained from the backup (or distribution) libraries. If you are wondering how a backup version came to exist in the distribution libraries, this topic is covered in “Accepting the SYSMOD into the distribution libraries” on page 30.

You can use the RESTORE command to remove SYSMODs from the target libraries and restore them to a previous level. The RESTORE command reverses APPLY processing, but has no effect on ACCEPT processing.

In this chapter, you will learn about the following topics:

- What happens during RESTORE processing
- How SMP/E keeps track of RESTORE processing
- Examples of using the RESTORE command
- A summary of the RESTORE command

What happens during RESTORE processing

SMP/E provides you with a method for removing an applied SYSMOD when its installation results in unexpected problems.

Removing the SYSMODs

SMP/E ensures the eligibility of the selected SYSMODs and checks whether other SYSMODs are affected before continuing with RESTORE processing. Because of the various relationships and dependencies among the many SYSMODs, this checking is very important to the integrity of your system. In fact, to ensure that the requisites for a SYSMOD being restored are processed appropriately, SMP/E may require the whole chain of prerequisites to be restored.

Selecting the elements

During RESTORE processing, SMP/E uses the information provided in the selected SYSMODs to determine which elements in the target zone should be replaced by elements in the related distribution libraries. The selection of elements is monitored by SMP/E to make sure that the correct functional level of each element is selected.

Checking the RESTORE process

SMP/E provides you with an option to stop RESTORE processing just before any updating takes place so you can ensure all prerequisites are satisfied before restoring any SYSMODs. This helps you see what will happen without actually making any changes to the elements in the target libraries.

Replacing the elements in the target libraries

When SMP/E is satisfied that the proper SYSMODs have been selected, it uses information from the target zone to determine which distribution zone describes the elements necessary to replace the SYSMOD's elements in the target libraries. The RESTORE command directs SMP/E to call system utilities that replace the elements in the target libraries with the previous level of the elements from the related distribution libraries.

How SMP/E keeps track of RESTORE processing

SMP/E updates the information about the SYSMODs that have been restored. Remember, the target zone reflects the contents of the target libraries. Therefore, after the utility work is complete, and the target libraries have been updated, the target zone is updated to accurately reflect the status of those libraries.

- All information in the target zone pertaining to the restored SYSMOD is removed. The element entries in the target zone are restored to reflect the distribution zone level of the elements.
- The global zone SYSMOD entries and MCS statements, which are stored in the SMPPTS data set, are deleted for those SYSMODs that have been restored. Any SMPTLIB data sets created during RECEIVE processing are also deleted for the restored SYSMOD. SMP/E automatically performs this global zone clean-up, unless you specify otherwise.

Figure 15 on page 28 shows what you have learned about RESTORE processing.

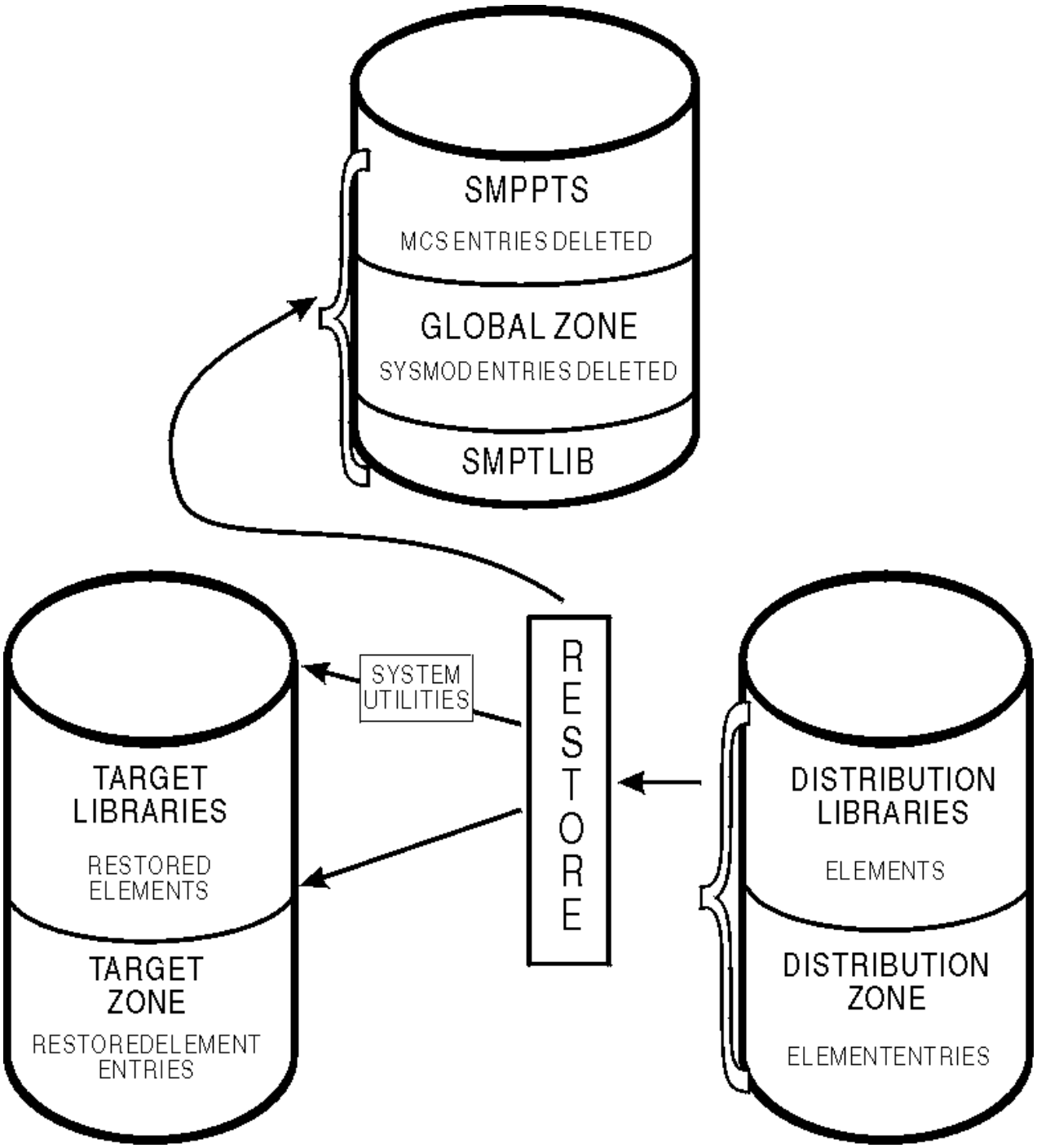


Figure 15. Results of RESTORE processing

Using the RESTORE command

The RESTORE command has operands that allow you to specify the criteria for removing SYSMODs from the target libraries. It also produces output that reports on its processing.

Examples

Let's look at a few examples of how you might use the RESTORE command.

Restoring a single SYSMOD

Assume you have applied a SYSMOD and, after some initial testing, you discover that a PTF SYSMOD is causing problems on your system. You can remove this SYSMOD by specifying the following commands:

```
SET      BDY(ZOZTGT1).
RESTORE  SELECT(UZ00001).
```

By issuing these commands, you instruct SMP/E to remove PTF UZ00001 from target zone ZOZTGT1 and replace its elements in the target libraries with the previous level of elements from the distribution libraries.

Restoring SYSMODs using the GROUP operand

When you want to remove a particular SYSMOD, it is not always easy to determine other SYSMODs that need to be restored in order to remove the bad one. Assume a particular PTF SYSMOD is causing a problem, and you want to know if it is dependent on any other SYSMODs so you can also restore those SYSMODs. This can be accomplished by specifying the following commands:

```
SET      BDY(ZOZTGT1).
RESTORE  SELECT(UZ00003)
         GROUP.
```

By issuing these commands, you instruct SMP/E to restore PTF UZ00003 and any other related PTFs from target zone ZOZTGT1, and replace their elements with the previous level from the distribution zone.

Restoring SYSMODs using the CHECK operand

In the previous example, you directed SMP/E to restore any dependent SYSMODs in order to remove the bad one. There may be times when you want to see which SYSMODs are restored without actually restoring them. You can do this with the CHECK operand by issuing the following commands:

```
SET      BDY(ZOZTGT1).
RESTORE  SELECT(UZ00003)
         GROUP
         CHECK.
```

After these commands are processed, you can check the SYSMOD Status report to see which SYSMODs would have been restored if you had not specified the CHECK operand. If you are satisfied with the results of this trial run, you can issue the commands again, without the CHECK operand, to actually restore the SYSMODs.

For a more complete description of all the RESTORE command operands, and for additional examples, see The RESTORE command in *z/OS SMP/E Commands*.

Reporting output

When RESTORE processing is complete, these reports will help you analyze the results:

SYSMOD status report
: Provides you with a summary of the processing that took place for each eligible SYSMOD, based on the operands you specified on the RESTORE command. It shows you which SYSMODs were restored, which were not restored, and why.

Element summary report
: Provides you with a detailed look at each element replaced or modified by RESTORE processing. It tells you in which libraries the elements were restored.

Causer SYSMOD summary report
: Provides you with a list of SYSMODs that caused other SYSMODs to fail, and describes the errors that must be fixed to successfully process the SYSMODs. This report can reduce the amount of work involved in figuring out which errors caused SYSMODs to fail.

File allocation report
Provides you with a list of the data sets used for RESTORE processing and supplies information about these data sets.

Additional reports may be produced depending on the work being done and the content of the SYSMODs. For more information about all the reports produced by the RESTORE command (and samples of actual reports), see The RESTORE command in *z/OS SMP/E Commands*.

Summary

Let's summarize what you have learned about using the RESTORE command to remove a SYSMOD from the target libraries. The RESTORE command:

- Removes the SYSMOD from the indicated target zone
- Calls system utilities to replace the SYSMOD's elements in the target libraries with elements from the related distribution libraries
- Records what is restored:
  - Target zone: Restores element entries to reflect their distribution zone level and deletes all information about restored SYSMOD.
  - Global zone: Deletes SYSMOD entries and MCS statements in SMPPTS for restored SYSMOD. Any SMPTLIB data sets created during RECEIVE processing are also deleted for the restored SYSMOD. (This global zone processing is optional.)
  - SMPSCDS data set: Deletes BACKUP entries for restored SYSMOD.
- Reports the results of processing

Note: Not all SYSMODs can be restored. For example, SMP/E cannot restore a SYSMOD that deletes another SYSMOD or that deletes a load module during APPLY processing.

Accepting the SYSMOD into the distribution libraries

You can use the ACCEPT command to install software in backup (or distribution) libraries. ACCEPT processing is very similar to APPLY processing with one important exception: ACCEPT processing is irreversible.

You will learn about the following topics:

- What happens during ACCEPT processing
- How SMP/E keeps track of ACCEPT processing
- Examples of using the ACCEPT command
- A summary of the ACCEPT command

What happens during ACCEPT processing

After you are satisfied that an applied SYSMOD has performed reliably in your target system, you can install it in your backup system (distribution) libraries.

Selecting the SYSMODs

You can specify operands on the ACCEPT command that tell SMP/E which of the received SYSMODs are to be selected for installation in the distribution libraries. SMP/E ensures that all other required SYSMODs have been installed or are being installed concurrently and in the proper sequence.

Selecting the elements

During ACCEPT processing, SMP/E uses the information provided in the selected SYSMODs to determine which elements should be installed in the distribution libraries. The selection of elements is monitored by SMP/E to make sure that the correct functional level of each element is selected.

Checking the ACCEPT process

SMP/E provides you with an option to stop ACCEPT processing before any updating takes place so you can ensure all prerequisites are satisfied before the installation of the SYSMODs. This helps you see what will happen (and helps you detect problem SYSMODs) without actually updating the distribution libraries.

Updating the distribution libraries

After the proper SYSMODs have been selected and the proper functional and service level of each element has been checked, SMP/E calls the system utilities (in the same manner as APPLY and RESTORE) to place the elements into the distribution libraries described in the distribution zone. The source of the elements is the SMPTLIB data sets, the SMPPTS data set, or the indirect libraries, depending on how the SYSMOD was packaged.

Note: When ACCEPT processing has been completed, there is no way it can be undone.

How SMP/E keeps track of ACCEPT processing

SMP/E updates the information about the SYSMODs that have been accepted. Remember, the distribution zone reflects the contents of the distribution libraries. Therefore, after the utility work is complete, and the distribution libraries have been updated, the distribution zone is updated to accurately reflect the status of those libraries.

- A SYSMOD entry is created in the **distribution zone** for each SYSMOD that has been accepted. Element entries (such as MOD and LMOD) are also created in the distribution zone for the elements that have been installed in the distribution libraries.
- **Global zone** SYSMOD entries and MCS statements in the SMPPTS data set are deleted for those SYSMODs that have been accepted into the distribution zone. Any SMPTLIB data sets created during RECEIVE processing are also deleted. If you do not want SMP/E to do this global zone clean-up, you have the option to indicate this to SMP/E, and the information is saved.

Figure 16 on page 32 shows what you have learned about ACCEPT processing.

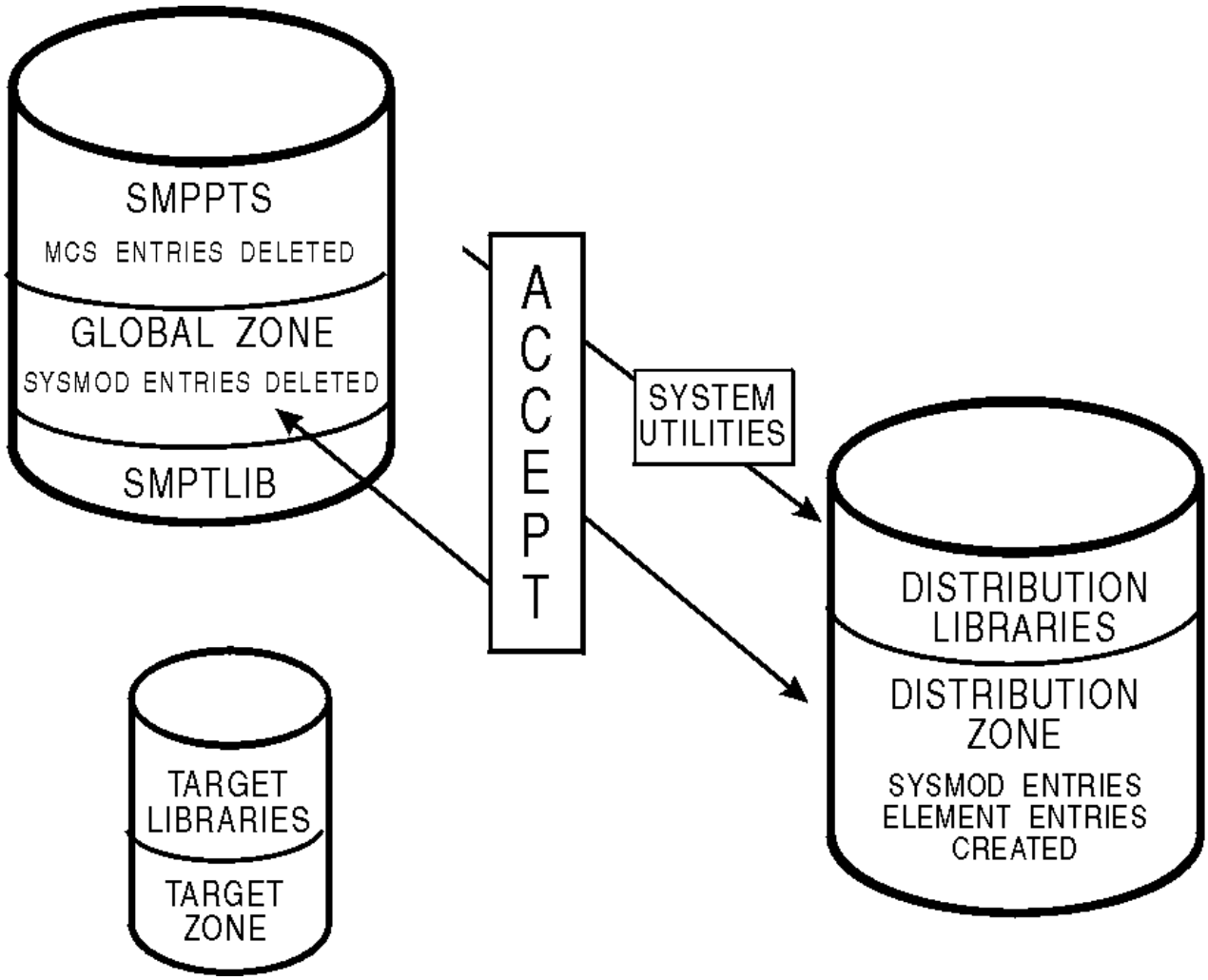


Figure 16. Results of ACCEPT processing

Using the ACCEPT command

The ACCEPT command has many operands that allow you great flexibility for further defining which SYSMODs you want installed in your distribution libraries. It also provides you with a variety of output based on the operands you specify.

Examples

Let's look at a few examples of how you might use the ACCEPT command.

Accepting PTF SYSMODs

After you have applied the SYSMODs into the target zone, you can define to SMP/E that you want to install only the PTF SYSMODs into the distribution zone. You can do this by specifying the following commands:

```
SET      BDY(ZOSDLB1).
ACCEPT   PTFS.
```

By issuing these commands, you direct SMP/E to accept all eligible PTF SYSMODs into distribution zone ZOSDLB1.

Suppose you do not want to install all the PTF SYSMODs, but only a select few. You can do this by specifying the following commands:

```
SET      BDY(ZOSDLB1).
ACCEPT   SELECT(UZ00001,UZ00002).
```

When you issue these commands, only PTFs UZ00001 and UZ00002 are installed in distribution zone ZOSDLB1.

Accepting SYSMODs for selected products

There may be times when you want to update only certain products on your system with the SYSMODs contained on a service tape. Assume you want to install all PTFs for a particular product. This can be accomplished by specifying the following commands:

```
SET      BDY(ZOSDLB1).
ACCEPT   PTFS
         FORFMID(HOP0001).
```

or

```
SET      BDY(ZOSDLB1).
ACCEPT   FORFMID(HOP0001).
```

In both cases, SMP/E accepts all applicable PTFs for the product whose FMID is HOP0001 and that is located in distribution zone ZOSDLB1. Unless you specify otherwise, PTFS is the default SYSMOD type.

Accepting SYSMODs having prerequisites

When installing a SYSMOD, you might not always know if it has prerequisites, or if the prerequisites are available. (Sometimes a prerequisite SYSMOD might not be received, or it might be held because it is in error.) In cases such as this, you can direct SMP/E to check whether an equivalent (or superseding) SYSMOD is available, by specifying the GROUPEXTEND operand.

Assume you want to process all PTFs for a product on your system, and you want to ensure that all other required SYSMODs are also processed. You can do this by specifying the following commands:

```
SET      BDY(ZOSDLB1).
ACCEPT   PTFS
         FORFMID(HOP0001)
         GROUPEXTEND.
```

By issuing these commands, you direct SMP/E to accept all PTFs, along with any other required SYSMODs, to the product whose FMID is HOP0001 and is located in the ZOSDLB1 distribution zone. If SMP/E cannot find a required SYSMOD, it looks for and uses a SYSMOD that supersedes the required one.

Accepting SYSMODs using the CHECK operand

In the previous example, SMP/E was directed to automatically include all SYSMODs needed for the specified product. There may be times when you want to see which SYSMODs are included before you actually install them. You can do this with the CHECK operand by issuing the following commands:

```
SET      BDY(ZOSTGT1).
ACCEPT   PTFS
         FORFMID(HOP0001)
         GROUPEXTEND
         CHECK.
```

After these commands are processed, you can check the SYSMOD Status report to see which SYSMODs would have been installed if you had not specified the CHECK operand. If you are satisfied with the results of this trial run, you can issue the commands again, without the CHECK operand, to actually install the SYSMODs.

For a more complete description of all the ACCEPT command operands and other examples, see The ACCEPT command in *z/OS SMP/E Commands*.

Reporting output

When ACCEPT processing is complete, these reports will help you analyze the results:

SYSMOD status report
Provides you with a summary of the processing that took place for each eligible SYSMOD, based on the operands you specified on the ACCEPT command. It shows you which SYSMODs were accepted, which were not accepted, and why.

Element summary report
Provides you with a detailed look at each element affected by ACCEPT processing. It tells you in which libraries the elements were installed.

Causer SYSMOD summary report
Provides you with a list of SYSMODs that caused other SYSMODs to fail, and describes the errors that must be fixed to successfully process the SYSMODs. This report can reduce the amount of work involved in figuring out which errors caused SYSMODs to fail.

File allocation report
Provides you with a list of the data sets used for ACCEPT processing and supplies information about these data sets.

Additional reports may be produced depending on the work being done and the content of the SYSMODs. For more information about all the reports produced by the ACCEPT command (and samples of actual reports), see *z/OS SMP/E Commands*.

Summary

Let's summarize what we have learned about using the ACCEPT command to install a SYSMOD in the distribution (or backup) libraries. The ACCEPT command:

- Selects SYSMODs to install
- Checks that all other required SYSMODs have been (or are being) installed
- Based on SYSMODs, selects elements to install
- Directs SMP/E to call the system utilities to update the distribution libraries
- Records what is accepted:
  - Distribution zone: Creates SYSMOD entries and element entries.
  - Global zone: Deletes SYSMOD entries and MCS statements in SMPPTS. Any SMPTLIB data sets created during RECEIVE processing are also deleted. (This global zone processing is optional.)
- Reports the results of processing

Remember, once you have accepted a SYSMOD, it cannot be restored.

Displaying SMP/E data

You can use SMP/E to provide helpful information for planning new installations, debugging problems, and other instances when you want to know the function and service level of your product software. There are several ways you can display data in the SMP/E database.

You will learn about the kinds of information that help you manage your system and the best method by which the information can be obtained.

Query dialogs
The easiest and fastest way to obtain just the information you want.

LIST command
When you need an all-inclusive hardcopy listing of information about your system.

REPORT commands
To check and compare the zone contents and generate command output that can be used to update your system.

SMP/E CSI application programming interface
To write an application program to query the contents of your system's CSI data sets.

Using the query dialogs

The SMP/E dialogs provide you with an online method of system management, software inventory, data base inquiries, and guidance. For example, with the Query dialogs, you can look up information in the CSI data set. The Query dialogs are one of the easiest and most direct methods you can use to obtain the content and status of any SYSMOD that has been processed by SMP/E. You can use the Query dialogs to display an entry in either a specific zone (CSI query) or in all zones (cross-zone query).

You can use the SMP/E dialogs to view a SYSMOD, even if it has been compacted. Use the Query dialog (zone is GLOBAL, entry type MCS, entry name is the SYSMOD name) and you will be shown the complete SYSMOD in an edit session. You may save the SYSMOD in a different location from this session. If you are using SMPPTS spill data sets, there is another benefit of viewing the SYSMOD from the Query dialog, in that you do not have to know in which SMPPTS data set the SYSMOD is stored; SMP/E will find it for you.

To get to the Query dialogs, you select Query (option 3) on the main menu for SMP/E (GIM@PRIM). This takes you to the initial Query panel, shown in Figure 17 on page 35. If you need assistance with using the Query dialogs, (or any of the SMP/E dialogs), help panels are available.

Let's assume you want to find out which SYSMODs have been applied to a particular target zone on your system. You can accomplish this task using the QUERY SELECTION MENU and selecting the CSI QUERY option (1), as shown in Figure 17 on page 35.

```
 GIMQUPO                        QUERY SELECTION MENU
===> 1

   1  CSI QUERY           - Display SMPCSI entries
   2  CROSS-ZONE QUERY    - Display status of an entry in
                            all zones
   3  SOURCEID QUERY      - Display SOURCEIDs for specified zone


   D  DESCRIBE            - Overview of using QUERY

   T  TUTORIAL            - Information on using QUERY

To return to the SMP/E primary option menu, enter  END .


5650-ZOS 5655-G44(C) COPYRIGHT IBM CORP 1982, 2013
```

Figure 17. Query selection menu

When the CSI QUERY panel is displayed (see Figure 18 on page 36), you can indicate that you want SMP/E to check target zone ZOSTGT1 for all SYSMOD entries.

```
GIMQU1PO                          CSI QUERY
===>

Specify the zone, entry type, and name to be queried:

      ZONE NAME      ===> ZOSTGT1 Name of the zone to be queried.
                                       To display a list of all zones,
                                       leave blank

      ENTRY TYPE     ===> SYSMOD  Entry type to be queried.
                                       To display a list of all valid
                                       entry types, leave ENTRY TYPE
                                       and ENTRY NAME blank

      ENTRY NAME     ===>         Entry name to be queried.
                                       Leave blank or use a wildcard
                                       (entry name pattern) to display
                                        a selection list.


To return to the Query selection menu, enter  END .
```

Figure 18. CSI query panel

Because the ENTRY NAME was left blank on the CSI QUERY panel, SMP/E displays another panel (see Figure 19 on page 36) that lists all the SYSMOD entries in target zone ZOSTGT1.

```
GIMQUSEA                CSI QUERY - SELECT ENTRY
===>                                                     SCROLL ===> PAGE

 Select one entry to query from target zone ZOSTGT1 :

 S   NAME         ACTION
     AZ00005
 s   UZ00001
     UZ00002
```

Figure 19. CSI query - Select entry panel

The CSI QUERY - SELECT ENTRY panel shows that SYSMODs AZ00005, UZ00001, and UZ00002 have been applied to target zone ZOSTGT1. If you want more information about the contents of SYSMOD UZ00001, you can select that entry by entering an S next to it, and another panel is displayed (see Figure 20 on page 36).

```
GIMQIT26                CSI QUERY - SYSMOD ENTRY
===>                                                     SCROLL ===> PAGE

 To return to the previous panel, enter  END .


 Entry Type:  SYSMOD                               Zone Name:  ZOSTGT1
 Entry Name:  UZ00001                              Zone Type:  TARGET

   Type:  PTF                     Status:  APPLIED
   FMID:  HOP0001
   Date/Time:  07.341    13.57:31    APPLIED



   -------- -------- -------- -------- -------- -------- --------
 MODS    MOD01   MOD02
```

Figure 20. CSI query - SYSMOD entry panel

The CSI QUERY - SYSMOD ENTRY panel displays all the relevant information pertaining to SYSMOD UZ00001.

As you can see, the QUERY dialog panels provide a quick and easy way for you to obtain information about your system.

Using the LIST command

In the course of managing your system, there may be times when you need a hardcopy listing of some type of information. You can use the LIST command to accomplish this task. For example, it might be necessary for you to have a record of the following:

- All entries of a specific type
- Selected entries of a specific type
- All entries that meet certain criteria

The LIST command can provide you with a listing of this information.

Examples

Let's look at a few basic examples of how you might use the LIST command.

Listing entries in a particular zone

In the course of managing your system, you might need to know which SYSMOD entries exist in the global zone. You can find this out by specifying the following commands:

```
SET      BDY(GLOBAL).
LIST     SYSMODS.
```

By issuing these commands, you direct SMP/E to list all the SYSMOD entries in the global zone.

Listing Specific Entries

Suppose you discover a problem on your system and need to determine whether a particular SYSMOD has been installed in the target zone. You can accomplish this by specifying the following commands:

```
SET      BDY(ZOSTGT1).
LIST     SYSMOD(UZ00001).
```

By issuing these commands, you direct SMP/E to provide you with information about SYSMOD UZ00001 in target zone ZOSTGT1.

Listing SYSMODs that are received but not installed

You might have received service into the global zone and are in the process of installing the service on your system. You want to see which of the SYSMODs you have received have not yet been installed in a target zone. This can be accomplished by specifying the following commands:

```
SET      BDY(GLOBAL).
LIST     SYSMODS
         NOAPPLY(ZOSTGT1).
```

By issuing these commands, you direct SMP/E to list the SYSMODs that have been received, but have not yet been applied to target zone ZOSTGT1.

Reporting output

When LIST processing is complete, these reports will provide you with the information that was requested:

- The LIST summary report provides you with information about the type of entry, name of entry, and status of entry for zones and data sets you have specified.
- The File allocation report provides you with a list of the data sets used for LIST processing, and supplies information about these data sets.

For a more complete description of the LIST command, additional examples, and samples of actual reports, refer to The LIST command in *z/OS SMP/E Commands*.

Using the REPORT commands

You can use the REPORT commands to check and compare the SYSMODs installed in the different zones that exist on your system. In addition to this checking, you can tell SMP/E to generate the necessary commands to synchronize the specified zones. You can later modify these commands, if necessary, and use them to install the indicated SYSMODs.

One of the REPORT commands (REPORT SYSMODS) is useful if you want to compare the SYSMODs installed in two zones. Use it to make the following comparisons:

- One distribution zone to another distribution zone
- One target zone to another target zone
- A distribution zone to a target zone
- A target zone to a distribution zone

Example

Let's look at a basic example of how you might use the REPORT SYSMODS command. Assume you have two systems using the same global zone, and you want to check which SYSMODs are installed in a target zone on one system, but are not installed in a target zone on the other system. You can accomplish this by specifying the following commands:

```
SET    BDY(GLOBAL).
REPORT SYSMODS
       INZONE(ZOSTGT1)
       COMPARED(ZOSTGT2).
```

By issuing these commands, you direct SMP/E to compare the SYSMOD content of zone ZOSTGT1 to that of zone ZOSTGT2. Any SYSMODs that are in zone ZOSTGT1 and are not in zone ZOSTGT2 appear in the resulting report.

SMP/E also provides output you can use to install those SYSMODs you deem appropriate.

Reporting output

When REPORT SYSMODs processing is complete, these reports will provide you with the information that was requested:

- The SYSMOD comparison report provides you with a summary of the SYSMODs found in the input zone, but not found in the comparison zone. It can help you determine which SYSMODs might need to be installed in the comparison zone so its content reflects that of the input zone.
- The File allocation report provides you with a list of the data sets used for REPORT processing, and supplies information about these data sets.

For a more complete description of the REPORT commands, additional examples, and samples of actual reports, see the section on SMP/E reports in *z/OS SMP/E Commands*.

SMP/E CSI application programming interface

The SMP/E CSI application program interface (GIMAPI) allows you to write application programs that have read-only access to data stored in SMP/E's CSI (Consolidated Software Inventory) data sets. GIMAPI is described in detail in *z/OS SMP/E Reference*.

Summary

Let's summarize what you have learned about using the Query dialogs, the LIST command, the REPORT command, and the CSI API to check SMP/E's records for your system:

- Query dialogs: Easy and fast way to obtain information
- LIST command: Best for hardcopy listing
- REPORT commands: Best for checking and comparing zone contents

- SMP/E CSI application programming interface: Best for writing an application program to query the contents of your system's CSI data sets.

Chapter 2. SMP/E concepts

This topic summarizes some basic concepts that you will need to understand before you can use SMP/E. It briefly describes:

- What SMP/E is
- What system modifications are
- The data sets used by SMP/E
- How SMP/E can help you install and maintain products, and monitor changes to products

What is SMP/E?

SMP/E is the basic tool for installing and maintaining software in z/OS systems and subsystems. It controls these changes at the element level by:

- Selecting the proper levels of elements to be installed from a large number of potential changes
- Calling system utility programs to install the changes
- Keeping records of the installed changes

SMP/E is an integral part of the installation, service, and maintenance processes for CBPDOs, ProductPacs, RefreshPacs, and selective follow-on service for CustomPacs. In addition, SMP/E can be used to install and service any software that is packaged in SMP/E system modification (SYSMOD) format.

SMP/E can be run either using batch jobs or using dialogs under Interactive System Productivity Facility/Program Development Facility (ISPF/PDF). SMP/E dialogs help you interactively query the SMP/E database, as well as create and submit jobs to process SMP/E commands.

These are some of the types of software that can be installed by SMP/E:

- Products and service provided in CBPDOs and CustomPac offerings
- Products and service from IBM Software Distribution Centers not provided in CBPDOs or CustomPac offerings
- Service provided in Expanded Service Options (ESOs)
- Other products and service

SMP/E can install software from any of these sources, provided it is packaged as a system modification, or SYSMOD.

What are SYSMODs?

Software, whether it is a product or service, consists of elements such as macros, modules, source, and other types of data such as CLISTs or sample procedures. For software to be installed by SMP/E, it must include control information for the elements. This information describes the elements and any relationships the software has with other products or service that may also be installed on the same system. The combination of elements and control information is called a *system modification*, or SYSMOD.

There are four types of SYSMODs:

Function SYSMODs (or functions)

Function SYSMODs introduce a new product, a new version or release of a product, or updated functions for an existing product into the system.

There are two types of function SYSMODs:

- A base function SYSMOD either adds or replaces an entire functional area in the system. Examples of base functions are SMP/E and MVS.

© Copyright IBM Corp. 1986, 2026

- A dependent function SYSMOD provides an addition to an existing functional area in the system. It is called dependent because its installation depends on a base function already being installed. Examples of dependent functions are the language features for SMP/E.

PTFs
PTFs are IBM-supplied, tested fixes for reported problems. They are meant to be installed in all environments. PTFs may be used as preventive service to avoid certain known problems that may have not yet appeared on your system, or they may be used as corrective service to fix problems you have already encountered. The installation of a PTF must always be preceded by that of a function SYSMOD, and often other PTFs as well.

APAR fixes
Authorized program analysis reports (APARs) are temporary fixes designed to fix or bypass a problem for the first reporter of the problem. These fixes may not be applicable to your environment. The installation of an APAR must always be preceded by that of a function SYSMOD, and sometimes of a particular PTF. That is, an APAR is designed to be installed on a particular preventive-service level of an element.

User modifications (USERMODs)
USERMODs are SYSMODs built by you, either to change IBM code or to add independent functions to the system. The installation of a USERMOD must always be preceded by that of a function SYSMOD, sometimes certain PTFs, APAR fixes, or other USERMODs.

Note: If you want to package a user application program or new system function in SMP/E format, the correct way is to build a base or dependent function SYSMOD, not a USERMOD.

Figure 21 on page 43 shows the hierarchy of the various SYSMOD types. This example shows two service chains: one for the base function HZY1101 and one for the dependent function JZY1121.

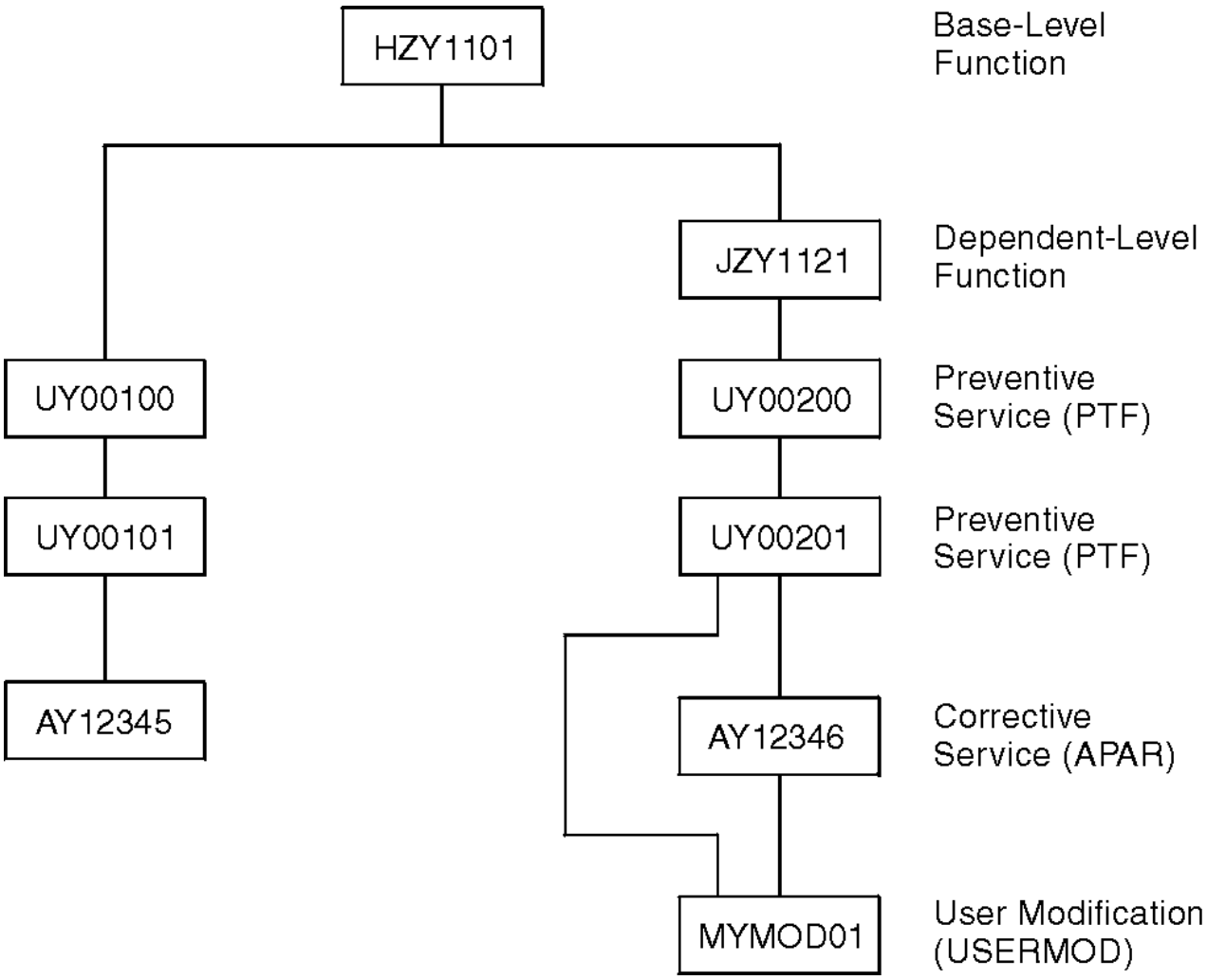


Figure 21. Example of a SYSMOD hierarchy

SMP/E keeps track of the functional and service levels of each element and uses the SYSMOD hierarchy to determine such things as which functional and service levels of an element should be installed and the correct order for installing updates for elements. For more information about how SMP/E determines the processing order of changes, as well as the functional and service levels of elements, refer to the sections on the APPLY command and the ACCEPT command in *z/OS SMP/E Commands*.

Data sets used by SMP/E

When SMP/E processes SYSMODs, it installs the elements in the appropriate libraries and updates its own records of the processing it has done. SMP/E installs program elements into two types of libraries:

- **Target libraries** contain the executable code needed to run your system (for example, the libraries from which you run your production system or your test system).
- **Distribution libraries** (DLIBs) contain the master copy of each element for a system. They are used as input to the SMP/E GENERATE command or the system generation process to build target libraries for a new system. They are also used by SMP/E for backup when elements in the target libraries have to be replaced or updated.

To install elements in these libraries, SMP/E uses a database made up of several types of data sets:

- **SMPCSI (CSI) data sets** are Virtual Storage Access Method (VSAM) data sets used to control the installation process and record the results of processing.

The CSI data set is a VSAM data set in which SMP/E maintains information about the system. A CSI can be divided into multiple partitions through the VSAM key structure. Each partition is referred to as a *zone*.

There are three types of zones:

- A single **global zone** is used to record information about SYSMODs that have been received into the SMPPTS data set. The global zone also contains information enabling SMP/E to access the other two types of zones, information about system utilities that SMP/E calls to install elements from SYSMODs, and information allowing you to tailor SMP/E processing.
- One or more **target zones** are used to record information about the status and structure of the operating system (or target) libraries. Each target zone also points to the related distribution zone, which can be used during APPLY, RESTORE, and LINK when SMP/E is processing a SYSMOD and needs to check the level of the elements in the distribution libraries.
- One or more **distribution zones** are used to record information about the status and structure of the distribution libraries (DLIBs). Each DLIB zone also points to the related target zone, which is used when SMP/E is accepting a SYSMOD and needs to check if the SYSMOD has already been applied.

Figure 22 on page 45 shows the relationships between SMP/E zones and libraries.

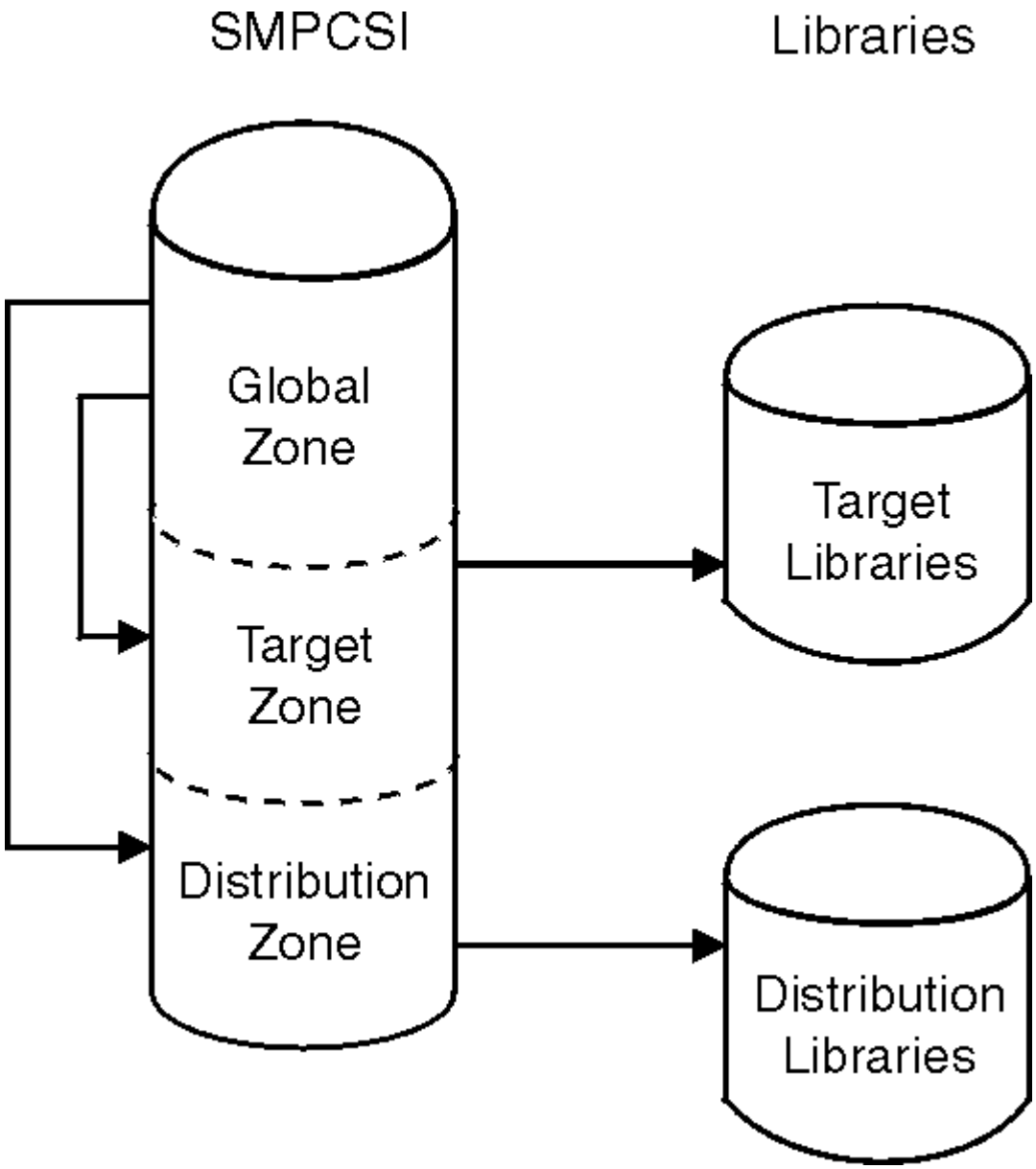


Figure 22. Summary of zone relationships

There can be more than one zone in an SMPCSI data set (in fact, there can be up to 32766 zones per data set). For example, an SMPCSI data set can contain a global zone, several target zones, and several distribution zones. The zones can also be in separate SMPCSI data sets. One SMPCSI data set can contain just the global zone, a second SMPCSI data set the target zones, and a third SMPCSI data set the distribution zones. For more information about ways to structure SMPCSI data sets, see *z/OS SMP/E Reference*.

- An **SMPPTS (PTS) data set** is a data set for temporary storage of SYSMODs waiting to be installed.

  The PTS is used strictly as a storage data set for SYSMODs. The RECEIVE command stores SYSMODs directly on the PTS without any modifications of SMP/E information. The PTS is related to the global zone in that both data sets contain information about the received SYSMODs. Only one PTS can be used for a given global zone. Therefore, you can look at the global zone and the PTS as a pair of data sets that must be processed (for example, deleted, saved, or modified) concurrently.
- The **SMPSCDS (SCDS) data set** contains backup copies of target zone entries modified during APPLY processing. Therefore, each SCDS is directly related to a specific target zone, and each target zone must have its own SCDS.

SCDS data sets are used by SMP/E to store backup copies of target zone entries modified during APPLY processing. Therefore, each SCDS is directly related to a specific target zone, and each target zone must have its own SCDS.

SMP/E also uses the following data sets:

- The **SMPMTS (MTS) data set** is a library in which SMP/E stores copies of macros during installation when no other target macro library is identified. Therefore, the MTS is related to a specific target zone, and each target zone must have its own MTS data set.
- The **SMPSTS (STS) data set** is a library in which SMP/E stores copies of source during installation when no other target source library is identified. Therefore, the STS is related to a specific target zone, and each target zone must have its own STS data set.
- The **SMPLTS (LTS) data set** is a library that maintains the base version of a load module. The load module in this library specifies a SYSLIB allocation in order to implicitly include modules. Therefore, the LTS is related to a specific target zone, and each target zone must have its own LTS data set.
- **Other utility and work data sets**.

SMP/E uses information in the CSI data sets to select proper element levels for installation, to determine which libraries should contain which elements, and to identify which system utilities should be called for the installation.

System programmers can also use the CSI data sets to obtain the latest information about the structure, content, and status of the system. SMP/E provides this information in reports, listings, and dialogs to help you:

- Investigate function and service levels
- Understand intersections and relationships of SYSMODs (either installed or waiting to be installed)
- Build job streams for SMP/E processing

How SMP/E can help you install and maintain products

This section briefly describes the tasks involved in general SMP/E processing, installing SYSMODs, and monitoring and maintaining your system, as well as the commands used to accomplish these tasks. Following is a guide to the commands that help you with these tasks. (You can use SMP/E dialogs for all the tasks in this list.) For more information about these commands, see *z/OS SMP/E Commands*.

Where to begin

You must specify a SET command before SMP/E can process any other commands.

Specifying the zone to be processed: SET

The main SMP/E database is a set of CSIs. When you establish the SMP/E database, you use the UCLIN command or the administration dialogs to divide the CSI into one or more partitions called *zones*. A zone contains control information for the following:

- A set of target libraries. (These zones are called *target zones*.)
- A set of distribution libraries. (These zones are called *distribution zones*.)
- The data present in the PTS and for general SMP/E processing. (This zone is called the *global zone*.)

You must tell SMP/E which zone you are working with before it can execute any other commands. You direct SMP/E processing to a specific zone by coding the zone name on the SET command.

Installing SYSMODs

The primary purpose of SMP/E is to install SYSMODs. This section describes the tasks and commands you can use.

Loading SYSMODs into the SMPPTS: RECEIVE

The RECEIVE command performs the following functions:

- Reads in SYSMODs from a sequential input file, defined by the SMPPTFIN DD statement
- Reads in HOLDDATA for exception SYSMODs from another sequential input file, defined by the SMPHOLD DD statement
- Determines which SYSMODs and HOLDDATA are applicable to your system
- Stores the SYSMODs and HOLDDATA in a storage data set (the PTS data set) and in the global zone

Note: The RECEIVE command can also be used to transfer software packages from a TCP/IP network connected server directly into an SMP/E directory or data sets.

Using internet service retrieval to request PTF or HOLDDATA: RECEIVE ORDER

You can use the RECEIVE ORDER command to request a new PTF service order from the IBM Automated Delivery Request server, or to download pending PTF service orders that resulted from a previous request.

Requesting a new PTF service order

The RECEIVE ORDER command performs the following functions:

1. Builds a software inventory based on the target zones specified. If no zones were specified, then SMP/E uses all target zones.
2. Submits a HOLDDATA or PTF order request to the IBM Automated Delivery Request server. The order request includes the content criteria specified by the user and contains the software inventory.
3. The server responds to SMP/E to indicate that the order was accepted.
4. SMP/E records information about the order in the global zone by creating a new ORDER entry. At this point the order is in a pending state, and remains so until its package is downloaded.
5. When the server receives the order request, it initiates the manufacturing processes to fulfill the HOLDDATA or PTF order. The IBM manufacturing processes use the supplied content criteria and the software inventory information to build a package containing either PTFs or HOLDDATA, or both, as specified in the request.
6. SMP/E waits a period of time for the order to be fulfilled. During this time SMP/E periodically queries the status of the order on the server. If the order is not fulfilled within the maximum allowed time, then the RECEIVE command processing ends and the order remains in a pending state.
7. If the order is fulfilled within the allowed time, the server sends SMP/E the information required to download the resultant package (FTP server name, directory where the package resides on the FTP server, user ID, password, and the package SHA-1 hash value).
8. SMP/E downloads the package to the local SMPNTS directory using FTP and the capabilities of the RECEIVE FROMNETWORK command. Once the package has been completely downloaded, SMP/E changes the status value for the order's ORDER entry in the global zone from PENDING to DOWNLOADED.
9. SMP/E expands the contents of the downloaded package and traditional RECEIVE processing stores PTFs and HOLDDATA into the global zone and SMPPTS data set (SMP/E skips this step if the user specified TRANSFERONLY on the RECEIVE command).

Downloading pending orders

When a requested PTF service order is not fulfilled within the allowed time, RECEIVE command processing stops and the order remains in the pending state. The IBM Automated Delivery Request server, however, continues to fulfill the order. You can use the RECEIVE ORDER command to process a pending order.

When the RECEIVE command is executed to process a pending order, SMP/E performs the following steps:

1. Reads the global zone to find the entry for the specified ORDER and determines if the order has a status of PENDING.
2. Queries the status for the pending order on the server. The server responds to indicate either that the order has been fulfilled and is ready to be downloaded, or that the order is not yet ready.
3. If the order is not yet ready for downloading, SMP/E again waits a period of time for the order to be fulfilled and be ready for downloading. During this time, SMP/E periodically queries the status of the order on the server. If the order is not fulfilled within the maximum allowed time, then RECEIVE processing ends and the order remains in a pending state.
4. If the order is fulfilled within the maximum allowed time, the server responds to SMP/E with the information required to download the resultant package (FTP server, directory where the package resides on the server, user ID, password, and the package SHA-1 hash value).
5. SMP/E downloads the package to the local SMPNTS directory using FTP and the capabilities of the RECEIVE FROMNETWORK command. Once the package has been completely downloaded, SMP/E changes the status value for the order's ORDER entry in the global zone from PENDING to DOWNLOADED.
6. SMP/E expands the contents of the downloaded package and traditional RECEIVE processing stores PTFs and HOLDDATA into the global zone and SMPPTS data set (SMP/E skips this step if the user specified TRANSFERONLY on the RECEIVE command).

Installing SYSMODs in the target libraries: APPLY

After the SYSMODs have been received, you can use the APPLY command to direct SMP/E to install them in the appropriate target libraries.

Installing SYSMODs in the distribution libraries: ACCEPT

After installing a SYSMOD in the target libraries and testing it to your satisfaction, you can use the ACCEPT command to have SMP/E install that SYSMOD in the distribution libraries and delete it from the PTS and the global zone.

Building a system from a set of distribution libraries: GENERATE

Use the GENERATE command to create a job stream that builds a system (a set of target libraries) from a set of distribution libraries.

Monitoring your system

This section describes the tasks and commands you can use to monitor your system.

Displaying information from the SMP/E database: LIST

Use the Query dialogs or the LIST command to display the data SMP/E retains. There are several operands you can use to list subsets of the data.

Checking and comparing zone contents: REPORT

The REPORT command helps you obtain information about SYSMODs installed on your system. There are several types of REPORT commands that you can use to do the following:

- **REPORT CROSSZONE** to list conditional requisites that must be installed in certain zones because of SYSMODs installed in other zones. This information can help you synchronize service for related products that are in different zones.
- **REPORT ERRSYSMODS** to list SYSMODs that have already been installed but are affected by error holds subsequently received.

- **REPORT SOURCEID** to list source IDs assigned to SYSMODs in a given zone or ZONESET.
- **REPORT SYSMODS** to list SYSMODs installed in one zone and applicable to a second zone, but not yet installed in it.

Managing the SMP/E database

This section describes the tasks and commands you can use to remove SYSMODs and process the SMP/E database as a part of your system maintenance.

Note:

1. There is no SMP/E command to remove a SYSMOD from the distribution libraries once you have accepted it.
2. Because all SMP/E processing is controlled by information in the CSI data set, you must provide the required information in the CSI before you process any SYSMODs.

Removing SYSMODs from the target libraries: RESTORE

After you have installed a SYSMOD in the target libraries, you should test it to make sure the change works correctly. If during testing you find an error in one of the SYSMODs you applied, you can use the RESTORE command to remove that SYSMOD from the system.

The RESTORE command causes the elements in the distribution libraries to be reinstalled on the target libraries. Therefore, when restoring a SYSMOD, you may have to restore more than the one SYSMOD in error. All SYSMODs that have not been accepted and that affect some of the same elements or need the same service level requisites as the one you are restoring must also be restored.

Removing SYSMODs from the PTS: REJECT

After receiving a SYSMOD, you can use the REJECT command to remove that SYSMOD from the PTS and the global zone. This can be done to rework a SYSMOD.

You can also use REJECT after you use NOPURGE to accept a SYSMOD into the distribution libraries. Using NOPURGE in the OPTIONS entry prevents SMP/E from deleting the global zone SYSMOD entry and the PTS MCS entry during ACCEPT processing. Later, you can delete the SYSMOD and MCS entries by using REJECT.

Processing the SMP/E database to create, update, or delete a single entry: UCLIN

You can use the Administration dialogs or the UCLIN command to create or change entries in CSI data sets. UCLIN can be used to create entries required during SMP/E processing, such as entries for:

- Identifying the utilities SMP/E is to use
- Providing information for dynamic allocation support

You can use UCLIN to delete orders from the SMPCSI data set. You can also use UCLIN to correct errors in entries or to alter processing information (including the ORDER RETENTION subentry in the OPTIONS entry).

Note: This command should be used with **extreme** caution; an incorrect change can cause a SYSMOD to be installed incorrectly later.

UCLIN updates only entries in SMP/E data sets. It does nothing to any elements or load modules in any product libraries. You must ensure that the appropriate changes are made to the libraries.

Processing the SMP/E database to update several entries of the same type: ZONEEDIT

Use the ZONEEDIT command to quickly change the values for a field in different DDDEF or UTILITY entries in the same zone. You can also use ZONEEDIT to change the cross-zone subentries of MOD, LMOD, and TARGETZONE entries.

Processing the SMP/E database to define the structure of the target libraries: JCLIN

To install a SYSMOD successfully, SMP/E must have information about the structure of your target libraries, such as:

- The library in which an element resides
- How modules are link-edited together to form load modules
- Where the load modules exist and their characteristics

The JCLIN command enables you to define the target library structure. In some instances, such as defining the target library structure for data elements, it is not necessary to use JCLIN, because the definition in the MCS statement is sufficient.

When processing the JCLIN command, you provide SMP/E with a job stream containing all the job steps (such as copies, link-edits, and assemblies) needed to create a set of target libraries from a set of distribution libraries. SMP/E then scans that input and builds all required entries to define the target system structure.

Processing the SMP/E database to write information to the SMPLOG data set: LOG

Use the LOG command to write to the SMPLOG and SMPLOGA data sets. This is useful for maintaining documentation about your system, such as who installed a certain SYSMOD, and why.

Processing the SMP/E database to clean up the SMPLTS, SMPMTS, SMPSTS, and SMPSCDS data sets: CLEANUP

Use the CLEANUP command to delete entries from the SMPMTS, SMPSTS, and SMPSCDS data sets for a particular target zone, if any entries still exist after the associated SYSMOD has been accepted into the related distribution zone.The CLEANUP command can also be used to remove the base version of load modules that are no longer needed from the SMPLTS data set.

Managing zones

This section describes the tasks and commands you can use to manage zones.

Copying a zone from one CSI to another: ZONECOPY

Use the ZONECOPY command to copy an entire target or distribution zone from one CSI data set to another. When you use the ZONECOPY command, SMP/E checks that the zone you copy into is empty except for a ZPOOL record.

Copying a zone to a sequential data set: ZONEEXPORT

The ZONEEXPORT command creates a copy of a specified distribution, target, or global zone and places it into another data set (a variable-block sequential data set). Having this copy of a specified zone gives you two advantages:

- You can use it as a backup copy to recreate a zone that has been accidentally erased or destroyed.
- You can use it as a “transportable” copy to create the same zone on another system or in another CSI data set on the same system.

Copying a zone from a sequential data set to a CSI data set: ZONEIMPORT

Use the ZONEIMPORT command to reload an exported zone into a distribution, target, or global zone. ZONEIMPORT can be used only with output from ZONEEXPORT.

Deleting a zone: ZONEDELETE

Sometimes you need to delete the SMP/E data related to one of the systems you are supporting. These are some examples of when you need to delete a zone:

- After a full system generation, you have to delete the information describing the previous target system libraries. Then you rebuild that information to describe the new set of target system libraries built from the distribution libraries.
- After installing a new level of a product that existed in its own target zone and distribution zone, you want to delete the information about the old level of the product and continue processing only the new level.

Merging zones: ZONEMERGE

Use the ZONEMERGE command to merge one zone into another zone after you use the GENERATE command or do a full system generation. When you use the ZONEMERGE command, SMP/E looks through the zones that ZONEMERGE works on and merges the data from them.

Renaming a zone: ZONERENAME

Use the ZONERENAME command to rename a zone. This command can help you keep meaningful names for your zones. You can also use ZONERENAME after a GENERATE command or a full system generation to help you change the name of the distribution zone that GENERATE or the system generation uses to build a new target zone.

Linking and relinking modules

This section describes the tasks and commands you can use to link or relink modules into load modules.

Handling cross-zone link-edits: LINK MODULE

Use the LINK MODULE command to link modules in one zone with load modules in another zone and to track this inclusion so that subsequent APPLY and RESTORE processing can automatically maintain the affected load modules.

Relinking load modules that use CALLLIBS: LINK LMODS

Use the LINK LMODS command to relink load modules that have a CALLLIBS subentry list in their LMOD entry (and therefore use the automatic library call option to implicitly include modules from a specified library). The LINK LMODS command may also be used to relink specific load modules that do not contain CALLLIBS, rebuilding them from scratch if necessary.

General SMP/E processing

This section describes general SMP/E processing tasks and the commands you can use to do them.

Requesting diagnostic processing: DEBUG

Use the DEBUG command to display the name of the issuing routine in each SMP/E message, to dump SMP/E control blocks, to dump VSAM request parameter list (RPL) control blocks, and to get a SNAP dump of SMP/E and its work areas whenever a specified message is issued.

Resetting return codes: RESETRC

Normally, the execution of an SMP/E command depends on the successful execution of all preceding commands. However, you can use the RESETRC command to reset the return code to zero. This allows SMP/E to run the next command, even if a preceding command failed.

Chapter 3. Preparing to use SMP/E

This topic discusses how to prepare to use SMP/E after it has been installed. It describes how to do the following:

1. Authorize use of SMP/E commands and services.
2. Allocate and initialize data sets in the SMP/E database.
3. Define entries in the CSI data set to do the following:
   - Create the zones associated with the PTS, distribution libraries, and target libraries.
   - Define the product and SMP/E libraries used during SMP/E processing.
   - Define the utility programs and associated parameters used during SMP/E processing.
   - Define the libraries that are eligible for retry processing after *x*37 abends.
4. Connect the SMP/E dialogs to ISPF.
5. Set up SMP/E for easier operation:
   - Specify SMP/E OPTIONS entry.
   - Specify link edit utility output DDDEF entries.
   - Specify automatic cross-zone requisite checking.
6. Define the information needed to invoke SMP/E.
7. Define exit routines, if desired, to customize SMP/E processing.

Authorizing use of SMP/E commands and services

The System Authorization Facility (SAF) restricts the use of certain SMP/E functions to users who have appropriate access to the SAF resources that protect those functions. The functions being controlled are all the SMP/E commands processed by program GIMSMP (for example, SET, RECEIVE, APPLY, ACCEPT, UCLIN, LIST, REPORT, and so on), the GIMZIP and GIMUNZIP service routines, and the GIMIAP copy utility invocation program.

The SAF FACILITY class resource names corresponding to these functions are of the following form:

- GIM.CMD.*command* for the SMP/E commands, where *command* is the name of the current SMP/E command being attempted. For example, GIM.CMD.APPLY for the APPLY command.
- GIM.PGM.*program* for the GIMZIP, GIMUNZIP, or GIMIAP service routines, where *program* is the name of the service routine being processed. For example, GIM.PGM.GIMZIP for GIMZIP.

To allow SMP/E users to execute SMP/E functions, you must protect the appropriate SAF FACILITY class resources in the active security manager and grant read access to those users that should be allowed to invoke the controlled SMP/E functions.

However, of all the functions described previously, several need to be controlled carefully. Users who are granted access to these resources have the potential to undermine system security regardless of any data set protections you may have in place. Therefore, they should be as trusted, for example, as users who have authority to update APF-authorized libraries. The functions that need to be controlled carefully and the corresponding SAF FACILITY class resources that SMP/E checks, are as follows:

Table 2. Functions and resource names that must be carefully controlled

Function	Resource name
RECEIVE command	GIM.CMD.RECEIVE
APPLY command	GIM.CMD.APPLY
ACCEPT command	GIM.CMD.ACCEPT

© Copyright IBM Corp. 1986, 2026

Table 2. Functions and resource names that must be carefully controlled (continued)	
Function	**Resource name**
RESTORE command	GIM.CMD.RESTORE
REJECT command	GIM.CMD.REJECT
LINK command	GIM.CMD.LINK
CLEANUP command	GIM.CMD.CLEANUP
Program GIMZIP	GIM.PGM.GIMZIP
Program GIMUNZIP	GIM.PGM.GIMUNZIP
Program GIMIAP	GIM.PGM.GIMIAP

You may define discrete profiles to control individual SMP/E functions, or you may choose to define generic profiles. However, if the resources are not protected by the security manager, or a user does not have READ authority to those resources, then SMP/E processing will stop. A sample RACF® command to define a single generic FACILITY class profile and to define a user ID in the access list of that profile is as follows:

- RDEFINE FACILITY GIM.* UACC(NONE)
- PERMIT GIM.* CLASS(FACILITY) ID(user ID) ACCESS(READ)

If you have activated SETROPTS RACLIST processing for the FACILITY class, you must also refresh SETROPTS RACLIST processing for the updates to take affect:

- SETROPTS RACLIST(FACILITY) REFRESH

It might be difficult to identify and add all necessary user IDs to the access list for the subject profiles, whether using a single generic profile as in the previous example, or multiple discrete profiles. With this in mind, although not recommended by IBM, it is possible to define the profiles with WARNING and AUDIT(FAILURES(READ)) to help identify and log all user IDs that currently invoke SMP/E functions and will require eventual definition in the profiles' access list. After sufficient analysis and after the access list has been updated, then profiles should be changed to NOWARNING.

Note: The preceding sample commands to define a FACILITY class profile and to define a user ID in the access list of that profile assume the use of RACF as the security manager. If you use a security manager other than RACF, see the appropriate documentation for equivalent commands.

Authorizing use of unsigned GIMZIP packages

By default, SMP/E does not use the deprecated SHA-1 hash algorithm to verify GIMZIP packages. Consequently, it does not download or process unsigned GIMZIP packages, since those packages rely solely on SHA-1 hashes. Unsigned GIMZIP packages are packages that are not digitally signed and do not contain SHA-256 hashes, which are more secure.

With APAR I029474, this behavior can be overridden. Although SMP/E still does not use SHA-1 by default, support is provided for SHA-1 hash verification for unsigned GIMZIP packages under specific conditions. If a user executing SMP/E commands such as RECEIVE FROMNETWORK, RECEIVE FROMNTS, RECEIVE ORDER, or running the GIMGTPKG program has UPDATE access to the SAF resource GIM.ACTION.ALLOW_UNSIGNED_GIMZIP in the FACILITY class, then SMP/E will allow and verify unsigned GIMZIP packages by using the deprecated SHA-1 hash algorithm. In this case, you must determine if your software vendors provide digitally signed GIMZIP packages. If they only supply unsigned GIMZIP packages, then you must ensure that the user identities performing SMP/E RECEIVE operations have the necessary UPDATE access to the SAF resource.

The following sample RACF commands authorize a user to download and use unsigned GIMZIP packages. If you use a security manager other than RACF, see the appropriate documentation for equivalent commands.

```
RDEFINE FACILITY +
  GIM.ACTION.ALLOW_UNSIGNED_GIMZIP +
  UACC(NONE)

PERMIT +
  GIM.ACTION.ALLOW_UNSIGNED_GIMZIP +
  CLASS(FACILITY) ID(ibmuser) ACCESS(UPDATE)

SETROPTS RACLIST(FACILITY) REFRESH
```

Allocating and initializing data sets in the SMP/E database

To install SYSMODs, SMP/E needs information about them and about the target and distribution libraries where they are to be installed. This information is kept in a database composed of the following data sets:

- SMPCSI (CSI)
- SMPPTS (PTS)
- SMPSCDS (SCDS)

CSI data sets

SMP/E uses CSIs to keep records of the system. To define the CSI data sets for your system, you need to do the following:

1. Decide how to organize the CSI data sets.
2. Understand catalog considerations for CSI data sets.
3. Allocate the CSI data sets.
4. Define alias entries for user catalogs.
5. Initialize the CSI data sets.
6. Define the zones for your system.
7. Understand how to reorganize a CSI data set to reclaim space.

Deciding how to organize CSI data sets

Before you allocate any CSI data sets, you must decide how to organize those data sets. Consider the following when you make your decision:

- The DASD configuration of your system libraries
- The organization of your system support structure
- How you want to use SMP/E

There are two basic structures for CSI data sets:

- Single-CSI
- Multiple-CSI

Descriptions of these structures are followed by examples.

Single-CSI structure

You can define the CSI structure to have one CSI that keeps track of all your system activity. The single-CSI data set has one global zone and one or more target and distribution zones. These are some reasons for having a single-CSI data set:

- The single-CSI data set optimizes the use of direct access storage.

- The single-CSI data set puts your whole establishment in one VSAM data set. This provides you with a single control point and one source of information for your whole system.

Single-CSI systems do have a drawback. When SMP/E needs to process a zone, it cannot request access to that specific zone; it must request access to the CSI data set containing that zone. As a result, if you have a single-CSI system, you can run only one background SMP/E job at a time.

Figure 23 on page 56 shows a single-CSI data set structure.

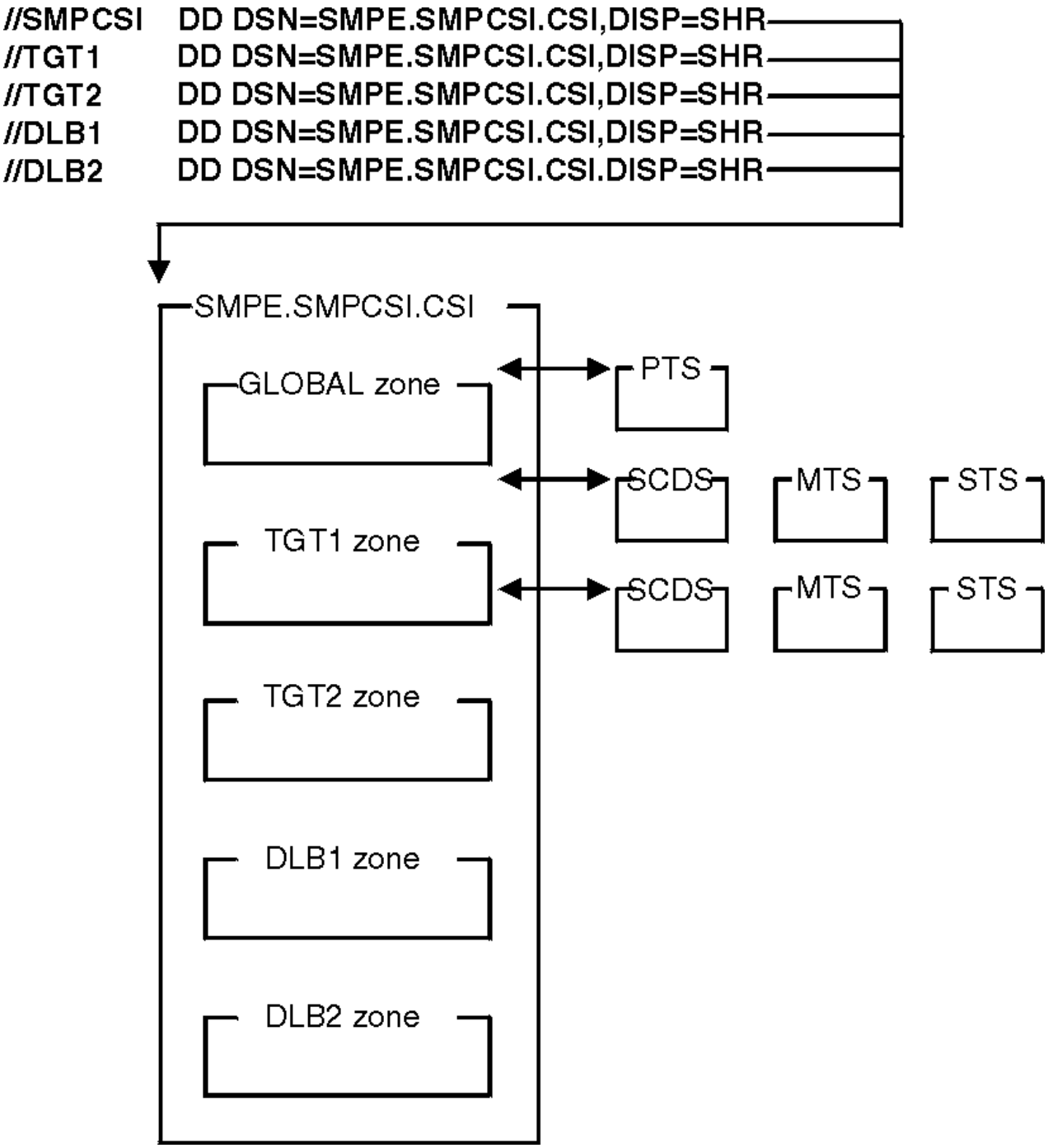


Figure 23. A single-CSI structure

Multiple-CSI structure

Multiple CSI data sets can be:

- Separate from each other, each with its own global zone.

- Connected by ZONEINDEX entries to a single global zone. (The global zone must be in one of the CSI data sets.)

Multiple CSIs enable you to use more than one VSAM data set for the global, target, and distribution zones. These are some reasons for having multiple CSI data sets:

- Your system may need multiple CSIs because of the characteristics of a particular installation—its programming support, its backup and update needs, and its need for added security and data integrity. For example, keeping libraries and their associated zones synchronized when you dump them for backup is easier if you keep them on the same physical DASD.
- Your system may need multiple CSIs if the support teams for different subsystems—such as MVS, CICS®, IMS, and NCP—are at different places.
- You may want to be able to run more than one background SMP/E job at a time. When SMP/E needs to process a zone, it cannot request access to that specific zone; it must request access to the CSI data set containing that zone. If your zones are in separate CSI data sets, processing for one zone does not prevent access to another zone.

Figure 24 on page 58 shows a multiple-CSI data set structure.

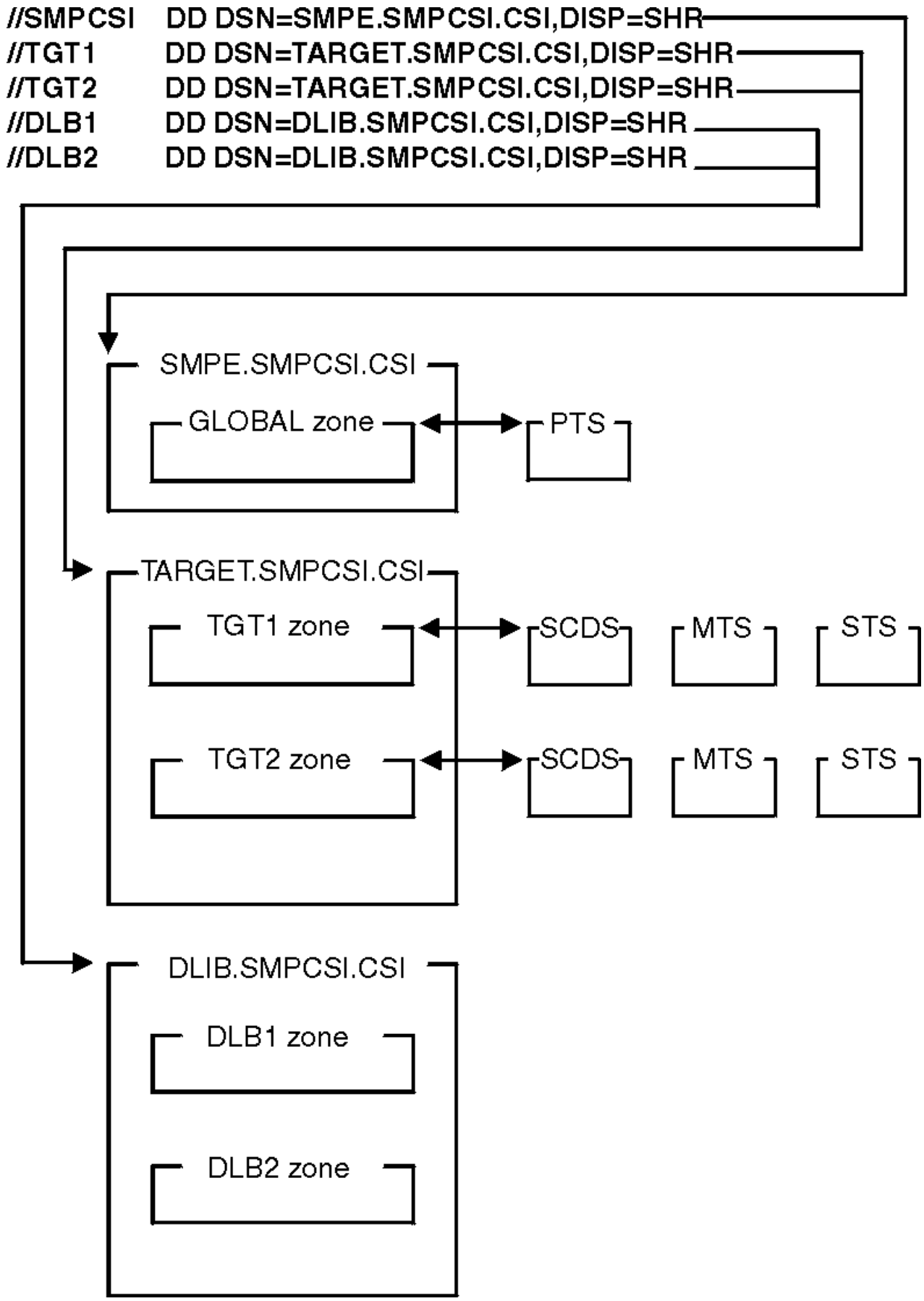


Figure 24. A multiple-CSI structure

Examples of CSI structures

The following examples show several ways to define a CSI structure describing a sample z/OS system with Job Entry Subsystem 3 (JES3) and an Information Management System (IMS) subsystem. In some of the examples, a single CSI contains multiple zones. Others show zones that are separated into multiple CSI data sets. Zones in separate CSIs can be connected by a single global zone. The CSI that contains the global zone is called the master CSI.

Example 1: Using a separate global zone for each subsystem

In Figure 25 on page 59, the existing DASD structure for the libraries is maintained, and a separate global zone is defined for each of the subsystems (MVS, JES3, and IMS). This CSI structure keeps control of the three subsystems separate. You might use this structure if the system programming support for the three subsystems is organizationally separate. A disadvantage is that there is no single control point (global zone) from which to manage or query the entire system.

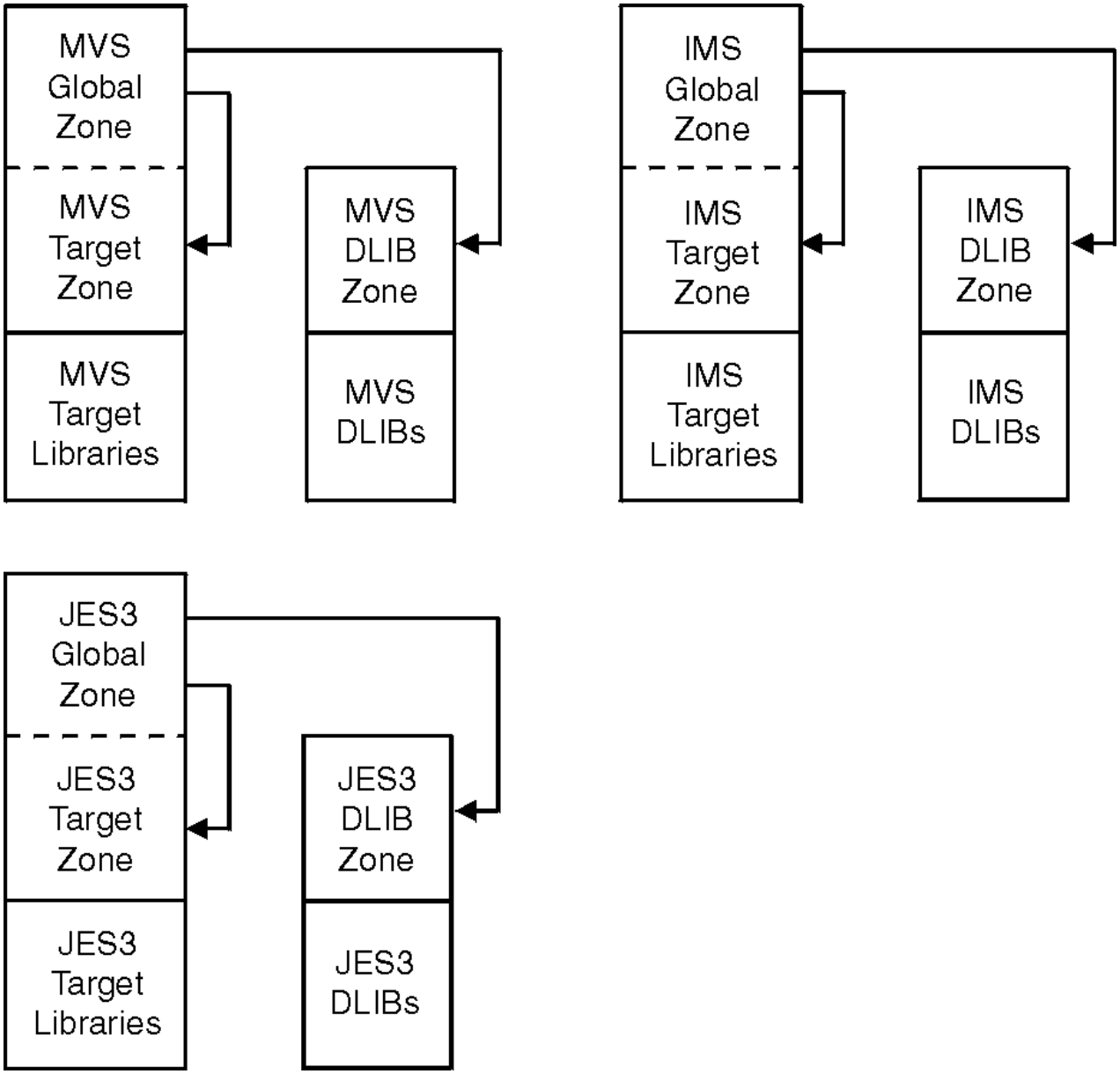


Figure 25. Using a separate global zone for each subsystem

Example 2: Using a single CSI for the whole system

In Figure 26 on page 60, all the SMP/E control information for the system is contained in a single CSI. The system structure is reflected by separate zones for MVS, JES3, and IMS. This CSI structure provides a single control point from which to manage or query the entire system.

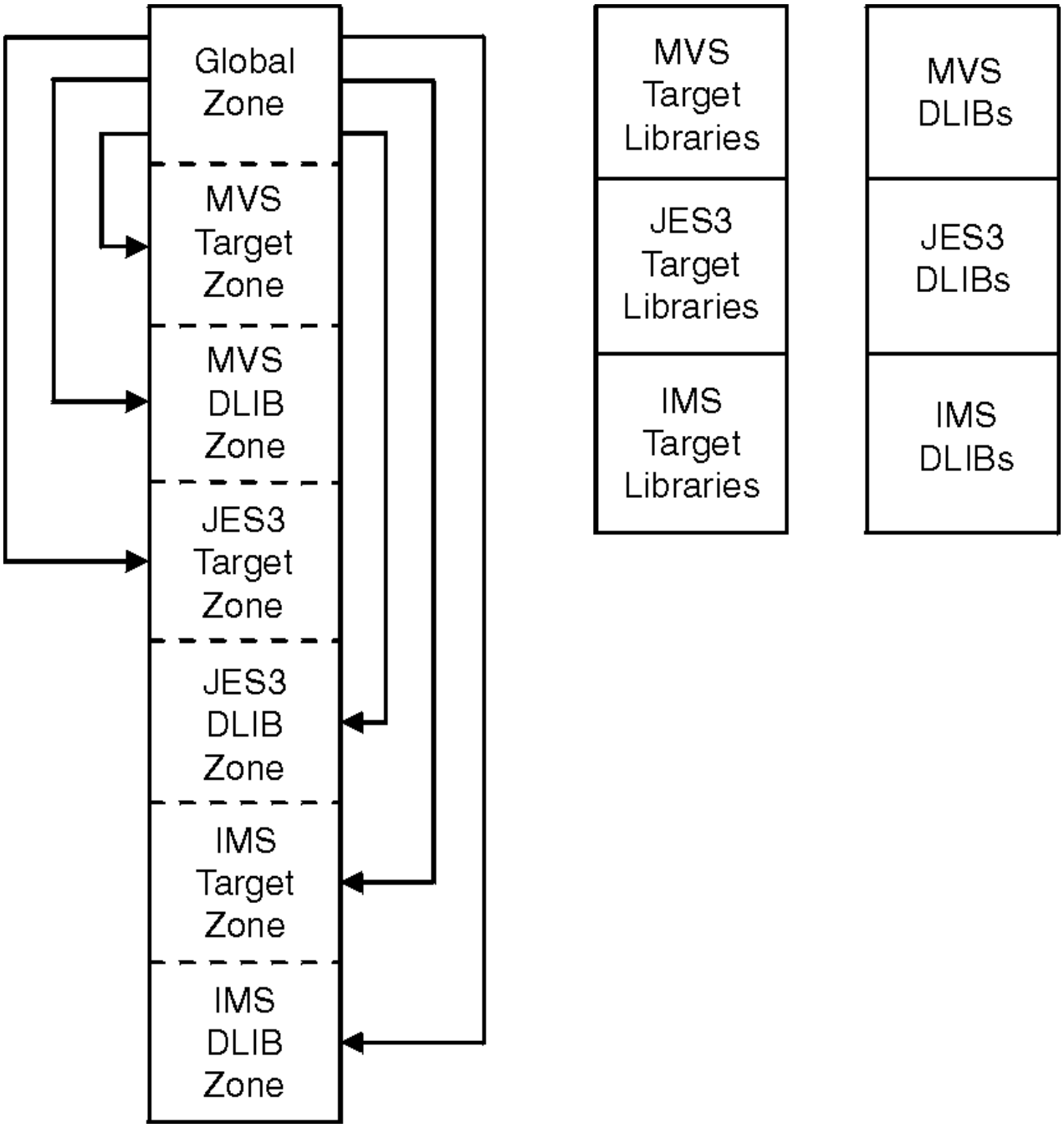


Figure 26. Using one CSI for the whole system

Example 3: Using a master CSI and multiple CSIs

In Figure 27 on page 61, one global zone is defined for the entire system in a master CSI, and separate CSIs are allocated for the JES3 and IMS subsystems on the packs where the subsystem libraries reside. This CSI structure provides the advantages of a common control point (as in Example 2) and keeps the SMP/E control information physically associated with the libraries it describes. This is useful when you dump packs for backup.

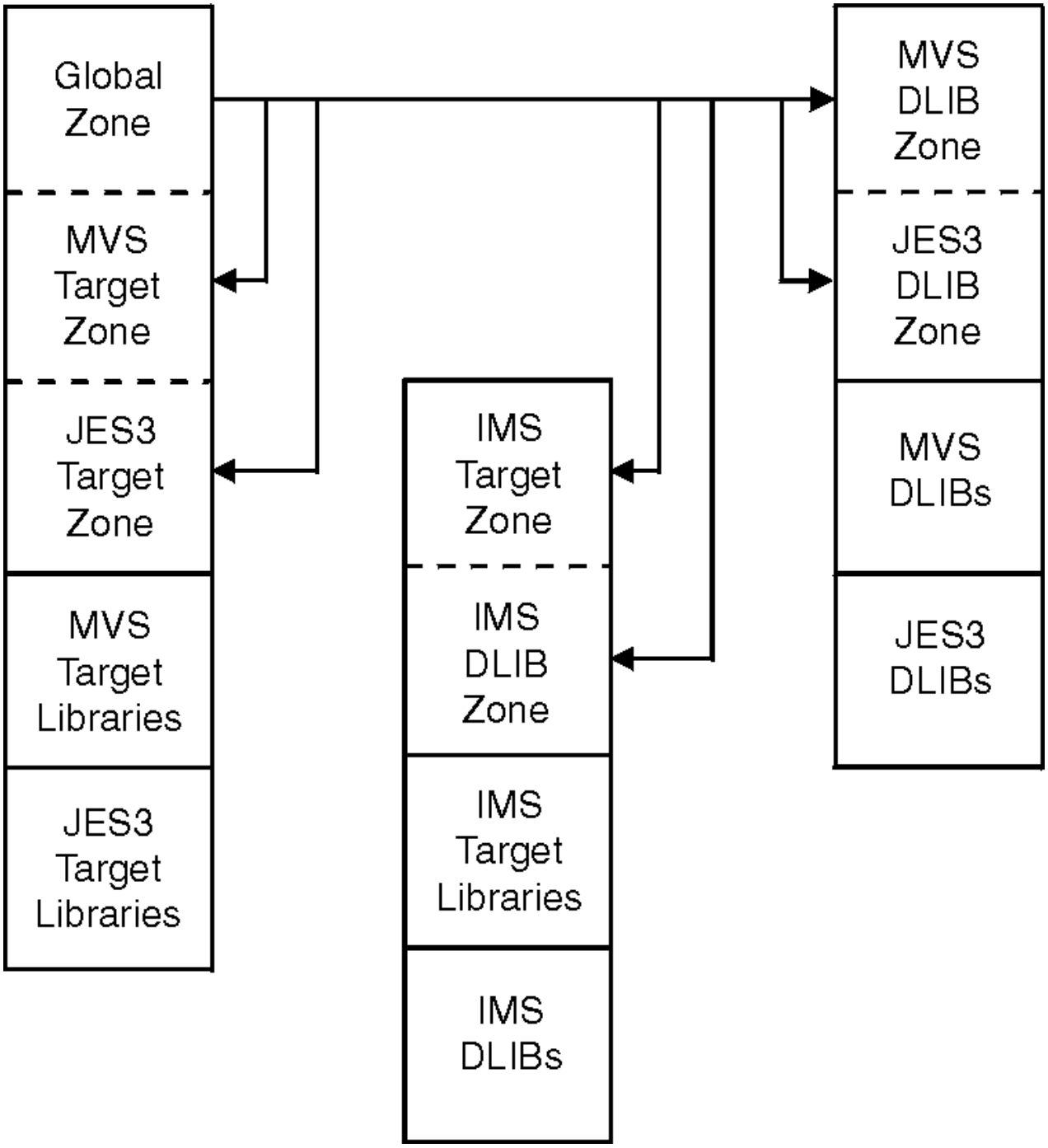


Figure 27. Using a master CSI

Example 4: Using a master CSI and a separate CSI for each zone

In Figure 28 on page 62, one global zone is defined for the entire system in a master CSI, and separate CSIs are allocated for the JES3 and IMS subsystems on the packs where the subsystem libraries reside. Unlike Example 3, each zone is in its own separate CSI. This CSI structure provides the advantages of a common control point (as in Example 2) and keeps the SMP/E control information physically associated with the libraries it describes. This is useful when you dump packs for backup.

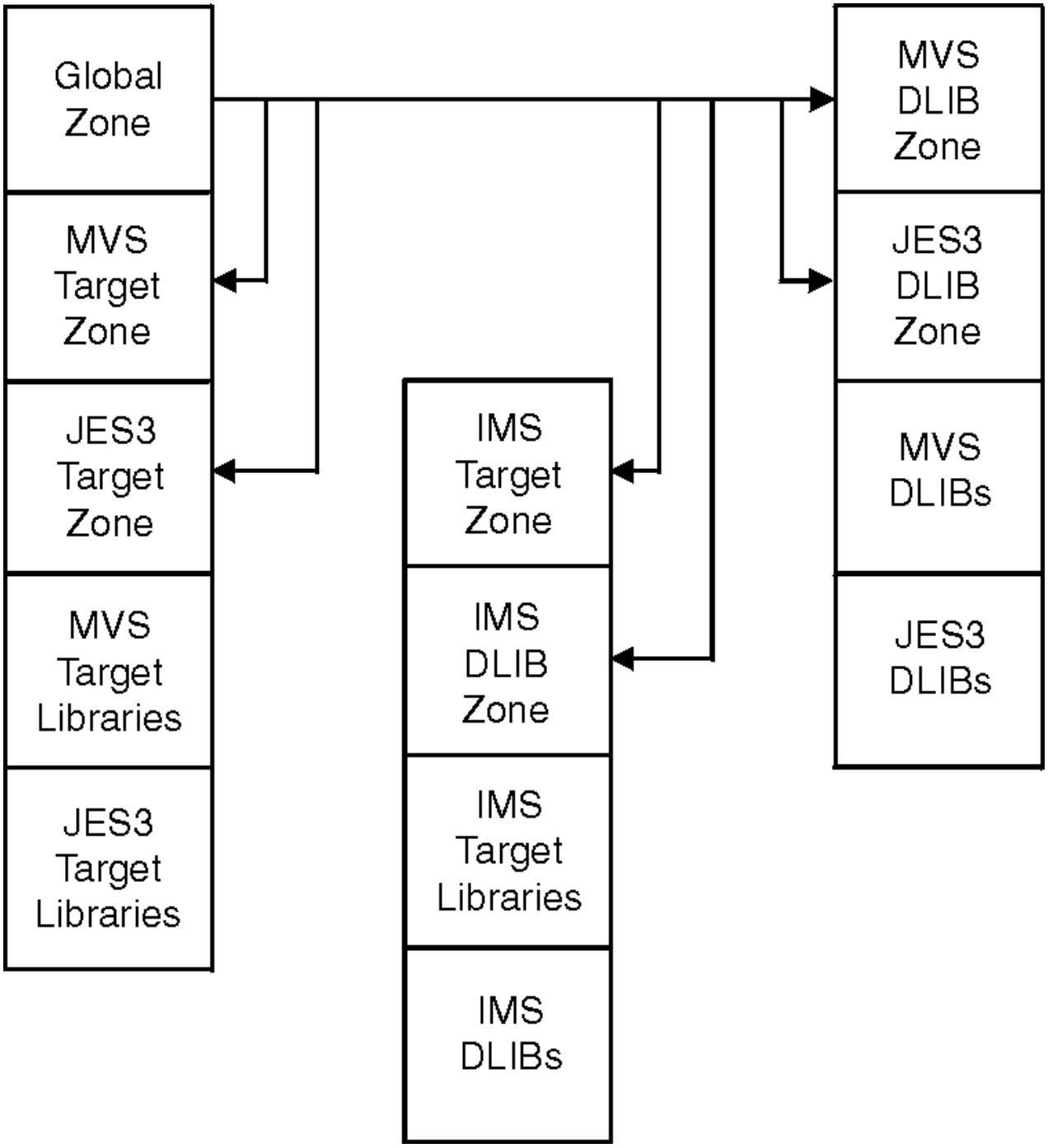


Figure 28. Using a master CSI and a separate CSI for each zone

Example 5: Using a master CSI and one CSI per SREL

In Figure 29 on page 63, one global zone is defined for the entire system in a master CSI, and separate CSIs are allocated for each SREL (MVS, CICS, NCP, and IMS/DB2). The target zones and DLIB zones associated with a given SREL are in the same CSI. This CSI structure provides the advantages of a common control point through one global zone (as in Example 2) and keeps the SMP/E control information physically associated with the libraries it describes. This is useful when you dump packs for backup.

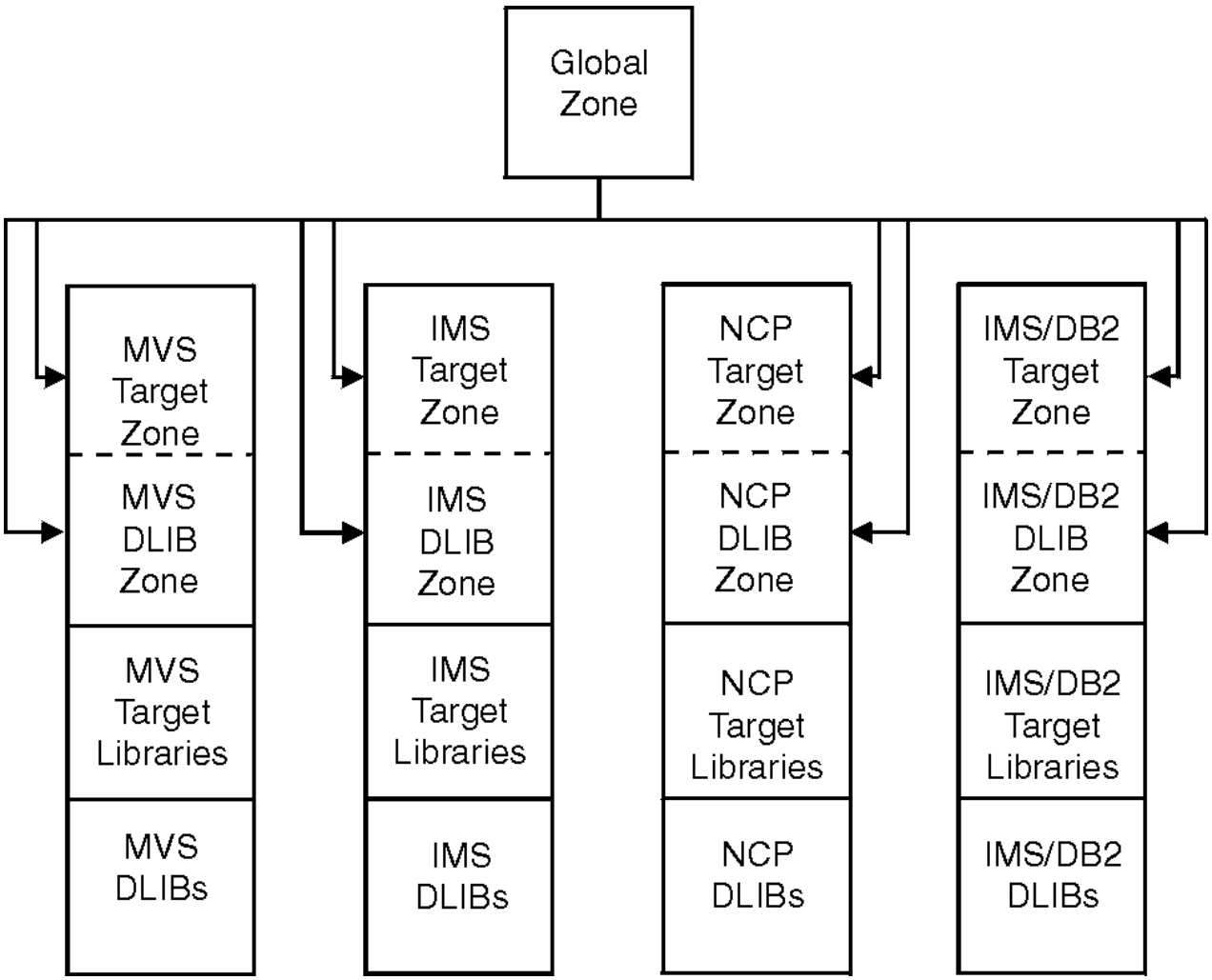


Figure 29. Using a master CSI and one CSI per SREL

Catalog considerations

When you catalog the CSI data sets used by SMP/E, remember these considerations:

User catalogs
You should catalog each CSI in a user catalog, not in the master catalog. However, the user catalog does not need to be on the same volume as the CSI.

Alias entries for user catalogs
Catalog information should be accessible through your master catalog. You can do it by defining each user catalog as an alias in the master catalog. For an example of defining an alias for a user catalog, see "Defining an alias entry for a user catalog" on page 65. Defining alias entries for user catalogs enables you to access all the CSI data sets and it eliminates the need to restore both the DASD containing the master catalog and the DASD containing the CSI after an I/O error.

- :
- :

Allocating a CSI data set

CSI data sets are keyed VSAM clusters and are allocated by use of access method services. For additional information and a description of the parameters, see *z/OS DFSMS Access Method Services Commands*.

The GIMSAMPU member in SYS1.SAMPLIB is a sample job to allocate and prime CSI and SMP/E operational data sets. The following sample job step, which is taken from the sample job in GIMSAMPU,

allocates a CSI data set with enough space to have multiple target or distribution zones and then initializes the CSI with the zpool record:

```
//DEFZONES EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//GIMZPOOL DD DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
//SYSIN    DD *
   DEFINE CLUSTER(                            +
            NAME(SMPE.GLOBAL.CSI)             +
            VOLUMES(volid)             +
            CYLINDERS(100 10)                 +
            FREESPACE(10 5)                   +
            KEYS(24 0)                        +
            RECORDSIZE(24 143)                +
            SHAREOPTIONS(2 3)                 +
                )                             +
       DATA (                                 +
            NAME(SMPE.GLOBAL.CSI.DATA)        +
            CONTROLINTERVALSIZE(8192)         +
            )                                 +
      INDEX (NAME(SMPE.GLOBAL.CSI.INDEX)      +
            CONTROLINTERVALSIZE(4096)         +
            )
   REPRO INFILE(GIMZPOOL)                     +
            OUTDATASET(SMPE.GLOBAL.CSI)
 /*
```

In your own job, be sure to include:

- NAME

  These are the naming conventions for CSI data sets:

  – The high-level qualifier must not be SYS1 if the CSI data set is cataloged in a user catalog rather than in the master catalog. However, the user catalog should be accessible through an alias entry in the master catalog.

  – The low-level qualifier must be `CSI`.

  These are examples of SMP/E data set names:

  ```
  'SMPE.SMPCSI.CSI'
  'PP.SMPCSI.CSI'
  'IMS.SMPCSI.CSI'
  'TEST.CSI'
  ```

- KEYS(24 0)
- RECORDSIZE(24 143)
- SHAREOPTIONS(2 3)

  SMP/E requires 2 as the cross-region SHAREOPTIONS value. It uses the default value of 3 as the cross-system SHAREOPTIONS value. Because SMP/E does not support cross-system sharing of the CSI, you cannot specify 4 as the cross-system value for SHAREOPTIONS. If you want to support cross-zone sharing, you must either use Global Resource Serialization (GRS) or a similar function, or ensure that the data set is not updated by multiple systems simultaneously.

- CONTROLINTERVALSIZE (CISIZE)

  If you allocate more than one CSI data set, ensure that they all have the same data CI size and index CI size. Doing so will allow SMP/E to take advantage of local shared resources (LSR) and VSAM resource pools. If the CSI data sets have different CISIZE values, SMP/E may open the data sets without using LSR.

- CYLINDERS

  The CYLINDERS value is only an estimated starting value. Your cylinder value may vary according to the device type, the software arrangement, the amount of service you install, and the number of CSIs.

Defining an alias entry for a user catalog

After allocating the CSI data sets, you should define alias entries for the high-level qualifiers of your CSI data sets in your master catalog and relate them to your SMP/E user catalog.

The following job creates an alias entry in the master catalog for a CSI data set named `SMPE.SMPCSI.CSI` that is cataloged in a user catalog named `SMPECAT`:

```
//CREATE   JOB    'accounting info',MSGLEVEL=(1,1)
//ALIAS    EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=A
//SYSIN    DD     *
  DEFINE ALIAS                      -
         (NAME(SMPE)                -
         RELATE(SMPECAT))           -
         CATALOG(AMASTCAT/password)
/*
```

If the CSI data sets are cataloged in different user catalogs, they must have different high-level qualifiers.

Defining zones for your system

Once you have allocated and initialized the CSI data sets, you need to create within them the entries SMP/E uses to maintain your system. The first entries you need to define are the zone definition entries (GLOBALZONE, TARGETZONE, and DLIBZONE entries) which set up zones in CSI data sets.

GLOBALZONE entry
: A global zone is created by defining a GLOBALZONE entry. The GLOBALZONE entry contains processing-related information for SMP/E. It is also used by SMP/E as an index to target and distribution zones, either in the same CSI or in different CSI data sets. The GLOBALZONE entry must be defined before you can do any other processing for that global zone.

TARGETZONE entry
: A target zone is created by defining a TARGETZONE entry. The TARGETZONE entry contains information SMP/E uses to process a specific target zone and the associated target libraries. It must be defined before you can do any other processing for that target zone.

DLIBZONE entry.
: A distribution zone is created by defining a DLIBZONE entry. The DLIBZONE entry contains the information SMP/E uses to process a specific distribution zone and the associated distribution libraries. It must be defined before you can do any other processing for that distribution zone.

Figure 30 on page 66 illustrates how zone definition entries define the relationships between zones.

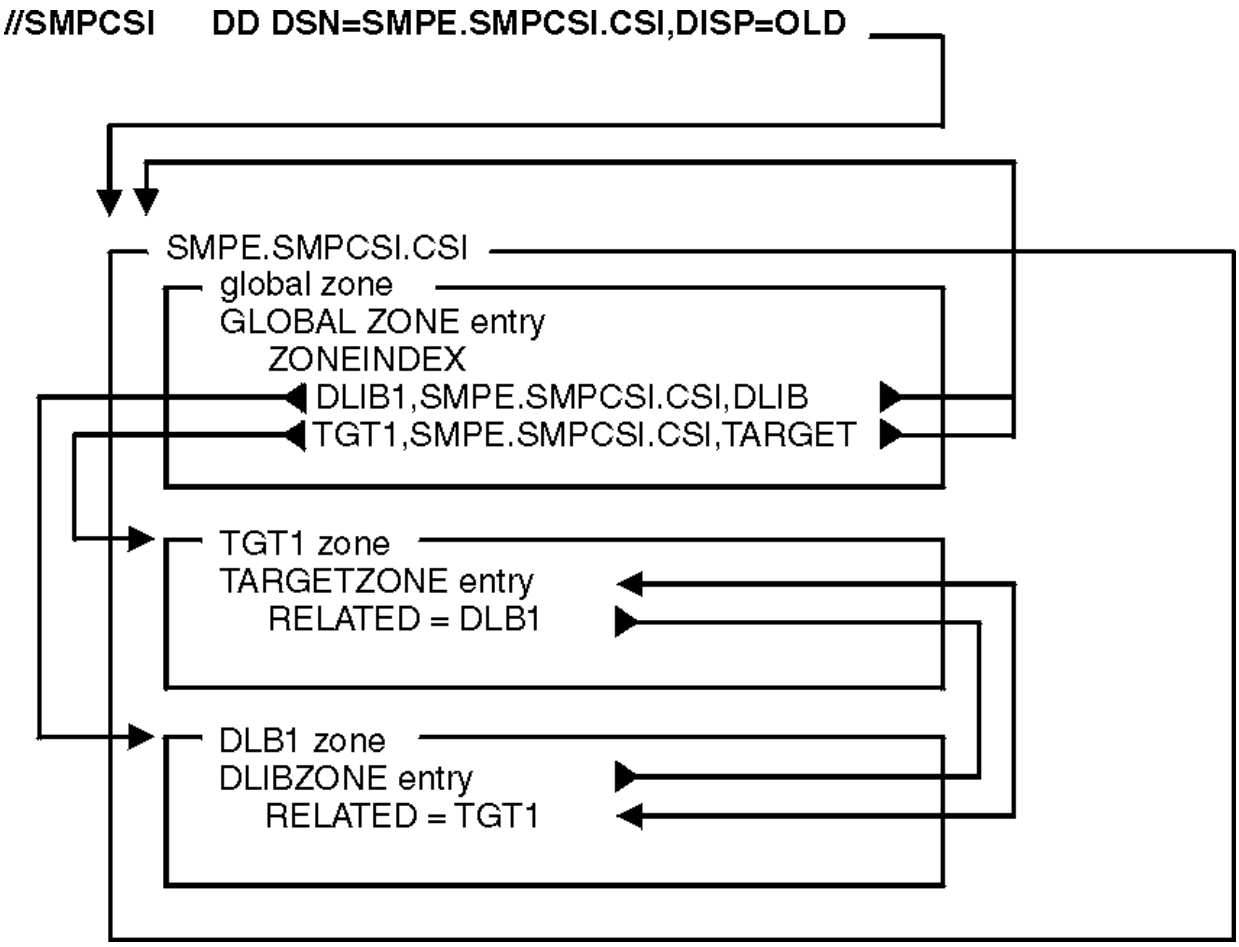


Figure 30. Relationships between zone definition entries

After you have defined the zones for your system, you can create other entries. SMP/E zones contain two basic types of entries:

- Entries controlling SMP/E processing.

  You generally define processing control entries through the SMP/E Administration dialogs or with the UCLIN command. Table 3 on page 67 summarizes the information specified in these entries.
- Entries describing the structure and status of the target and distribution libraries.

  Status and structure entries are generally created by SMP/E when you install SYSMODs, run the JCLIN command, or copy entries from one zone to another. Table 4 on page 67 summarizes the information specified in these entries.

SMP/E provides a member in SYS1.SAMPLIB (GIMSAMPU) containing sample UCLIN statements to define entries for a basic z/OS system. You can access this member by use of standard system utilities. The sample definitions are syntactically correct and can be used as the basis for your CSI entries. This sample is not complete for all systems, but it is an example of the types of information various entries need. For examples of UCLIN to define entries, see *z/OS SMP/E Commands*, which has the UCLIN syntax for each entry type. Also see the section on SMP/E data set entries in *z/OS SMP/E Reference*, which contains a description of the syntax plus examples and notes on its use.

Table 3. Entries controlling SMP/E processing

Type of information	Entry type	Zone where defined		
		Global	Target	DLIB
Data set definitions for dynamic allocation[1]	DDDEF	X	X	X
DLIB zone processing information	DLIBZONE			X
Exception (held) SYSMODs[2]	HOLDDATA[2]	X		
FMIDs to limit the SYSMODs processed by an SMP/E command	FMIDSET	X		
Global zone processing information	GLOBALZONE	X		
Processing options[3]	OPTIONS	X		
Target zone processing information	TARGETZONE		X	
Utility program parameters[4]	UTILITY	X		
Zone names to limit the SYSMODs processed by an SMP/E command	ZONESET	X		

1. For more information about dynamically allocating data sets, see "How to dynamically allocate data sets to be used during SMP/E processing" on page 69.
2. For more information about processing exception SYSMODs, see Chapter 11, "Managing exception SYSMODs," on page 143. HOLDDATA entries cannot be updated by UCLIN or the Administration dialogs.
3. For more information about defining information to be used during SMP/E's retry processing after *x*37 abends, see "Recovering after errors from utility processing" on page 75.
4. For more information about defining utility programs and associated parameters, see "Defining utility programs and associated parameters to SMP/E" on page 71.

Table 4. Entries describing the status and structure of the target and distribution libraries

Type of information	Entry type	Zone where defined		
		Global	Target	DLIB
Assembler statements that can be assembled to create an object module	ASSEM		X	X
Data elements installed in the target or distribution libraries (data elements are elements other than macros, modules, or source)	Data element entries		X	X
Distribution libraries that were totally copied to target libraries	DLIB		X	X
Elements installed in a UNIX file system	Hierarchical file system element entries		X	X
Java Archive files	JAR		X	X
Java Archive update files	JARUPD		X	X
Load module information	LMOD		X	X

Table 4. Entries describing the status and structure of the target and distribution libraries (continued)

Type of information	Entry type	Zone where defined		
		Global	Target	DLIB
Macros that have been installed in the target or distribution libraries	MAC		X	X
Module source that has been installed in the target or distribution libraries	SRC		X	X
Modules used to create load modules in the target libraries	MOD		X	X
Program objects	PROGRAM		X	X
Software product information	PRODUCT	X		
Software product feature information	FEATURE	X		
SYSMODs that have been processed	SYSMOD	X	X	X

Reorganizing a CSI data set to reclaim space

During normal SMP/E processing, VSAM control interval splits and control area splits can occur. These splits use up some of the free space in each control interval or control area, and this can degrade SMP/E performance and DASD space utilization. You should monitor your VSAM data sets regularly to determine how many splits have occurred and how much free space remains. The following job lists the catalog entry for data set `SMPE.SMPCSI.CSI`:

```
//LISTCAT  JOB    'accounting info',MSGLEVEL=(1,1)
//STEP01   EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=A
//SYSIN    DD     *
  LISTCAT                          -
         ENTRIES(SMPE.SMPCSI.CSI) -
         ALL
/*
```

After examining the LISTCAT output, you may determine that the CSI should be reorganized to eliminate splits in the control intervals or control areas and to reset the amount of free space available. This can be done through the access method services EXPORT and IMPORT commands. Once a CSI has been exported, a new CSI can be allocated, and the exported CSI can be imported so that normal SMP/E processing can continue.

Note: These examples are not the only way of compressing the CSI. You may prefer to use another method, drawing on your experience and knowledge of VSAM.

The following is a sample job for exporting the CSI:

```
//EXPORT   JOB    'accounting info',MSGLEVEL=(1,1)
//STEP01   EXEC   PGM=IDCAMS
//SMPCSI   DD     DSN=SMPE.SMPCSI.CSI,DISP=OLD
//TEMPCSI  DD     DSN=SMPCSI.DATA,DISP=OLD
//SYSPRINT DD     SYSOUT=A
//SYSIN    DD     *
  EXPORT SMPE.SMPCSI.CSI  -
         INFILE(SMPCSI)   -
         OUTFILE(TEMPCSI)
/*
```

The following is a sample job for importing the CSI:

```
//IMPORT   JOB    'accounting info',MSGLEVEL=(1,1)
//STEP01   EXEC   PGM=IDCAMS
//SMPCSI   DD     DSN=SMPE.SMPCSI.CSI,DISP=OLD
//TEMPCSI  DD     DSN=SMPCSI.DATA,DISP=OLD
//SYSPRINT DD     SYSOUT=A
```

```
//SYSIN    DD     *
  IMPORT  INFILE(TEMPCSI)  -
          OUTFILE(SMPCSI)  -
          INTOEMPTY
/*
```

Note:

1. If you want to delete the original CSI (SMPE.SMPCSI.CSI) when the exported copy (SMPCSI.DATA) is created, do not use the IDCAMS TEMPORARY keyword on the EXPORT command.

   If you want to make a backup copy of the CSI, you can use the TEMPORARY keyword on the EXPORT command to keep the original CSI intact.
2. Use a sequential data set to receive the exported CSI.
3. After allocating a new CSI to be imported into, do not prime it with the GIMZPOOL record provided in SYS1.MACLIB; if you do, the import operation will fail.

PTS data sets

The PTS data set is used as temporary storage for SYSMODs. It contains one member for each SYSMOD received. Each member is called a modification control statement (MCS) entry and is an exact copy of the SYSMOD as it was received from the SMPPTFIN data set. The name of an MCS entry matches the ID of the SYSMOD it contains. Generally, the MCS entries are kept on the PTS until the SYSMOD is accepted; then, under normal processing, they are deleted.

Each PTS data set must be associated with only one global zone. The allocation of space and directory blocks for the SMPPTS depends on your plans for installing and maintaining the functions managed by the global zone. For more information about allocating the SMPPTS data set, see *z/OS SMP/E Reference*.

SCDS data sets

The SCDS data set contains backup copies of target-zone entries that are modified during APPLY processing. These backup copies are made before the entries are (1) changed by inline JCLIN, a ++MOVE MCS, or a ++RENAME MCS or (2) deleted by an element MCS with the DELETE operand. The backup copies are used during RESTORE processing to return the entries to the way they were before APPLY processing.

Each backup copy of an entry is associated with the SYSMOD that caused the entry to be backed up. Together, the collection of entries associated with a SYSMOD is called the BACKUP entry for that SYSMOD. When you process the SCDS (for example, to list entries), you can specify only BACKUP entries; you cannot process individual entries within a BACKUP entry.

Each target zone within a CSI must have its own SCDS. The correct SCDS to be used during processing is determined either by the SMPSCDS DDDEF entry in each specific target zone or by a DD statement in the JCL. An SCDS can be allocated the same way as any normal partitioned data set. The allocation of space and directory blocks for this data set depends on your plans for installing and maintaining functions. For more information about allocating the SMPSCDS, see *z/OS SMP/E Reference*.

How to dynamically allocate data sets to be used during SMP/E processing

The processing of SMP/E commands requires a variety of data sets. You can either provide the DD statements for these data sets (such as in a cataloged procedure) or have SMP/E allocate the data sets dynamically. Dynamic allocation has the advantage that data sets are allocated only as they are needed; DD statements must successfully allocate all data sets, regardless of whether they are needed for the command being processed.

There are some drawbacks to using DD statements. For example, all the data sets defined by DD statements must be successfully allocated, regardless of whether they are needed for the command being processed. In addition, if you are running several SMP/E commands, you must be careful to use the correct DD statements for each command. If you are processing zones that are in different CSI data sets,

you must make sure to provide a DD statement that points to each of those zones and their associated CSIs.

With dynamic allocation, you do not have these problems. Subsequent topics describe the sources from which SMP/E can get the information it needs to allocate data sets dynamically and how it chooses which of these sources to use.

Sources of information for dynamic allocation

SMP/E can use several sources of information to allocate data sets dynamically:

- DDDEF entries
- SMPPARM member GIMDDALC
- Standard defaults

DDDEF entries

You can use DDDEF entries to provide SMP/E with information it needs to allocate any of the following:

- Permanent data sets, such as target libraries, distribution libraries, and SMP/E data sets
- Temporary data sets
- SYSOUT data sets
- Work data sets
- Path names for elements and load modules residing in a UNIX file system

Note: A member of a partitioned data set cannot be specified using a DDDEF entry.

The name of the DDDEF entry must match the ddname of the data set it describes and the entry must exist in the zone that uses the data set. DDDEF entries provide more flexibility than DD statements; they enable different zones to use different data sets for the same ddname and they use resources more efficiently because they allow SMP/E to allocate only the data sets it needs.

DDDEF entries can include the following information:

- Data set name
- Unit type
- Volume serial number
- Initial data set status: NEW, OLD, MOD, or SHR
- Final data set status: KEEP, DELETE, or CATALOG
- How the data set is to be allocated: blocks, cylinders, or tracks
- Primary and secondary values for space allocation
- Whether the data set should be RACF-protected
- Whether the data set is SMS-managed
- Directory information used to allocate the pathname for an element or load module residing in a UNIX file system.

For more information about DDDEF entries, see *z/OS SMP/E Reference*.

SMPPARM member GIMDDALC

Another way to provide SMP/E with information about data sets is through GIMDDALC control statements in SMPPARM member GIMDDALC. For more information about GIMDDALC, see GIMDDALC control statements in *z/OS SMP/E Reference*.

Standard defaults

SMPOUT and SYSPRINT are critical for SMP/E to operate properly. Therefore, in case they are not defined, SMP/E has built-in defaults for them.

- SMPOUT is allocated either as SYSOUT (for background processing) or to the terminal (for foreground processing).
- SYSPRINT is allocated as SYSOUT.

How dynamic allocation works

Once SMP/E has determined which data sets are needed for the command it is processing, SMP/E checks whether DD statements have been provided for any of those data sets. SMP/E uses information from those DD statements in allocating the data sets to which they apply. If any data sets lack DD statements, SMP/E must allocate them dynamically. To get the information it needs to do this, SMP/E checks the following sources in the order shown.

DDDEF entries
If the zone specified on the SET command contains a DDDEF entry for the required data set, SMP/E uses that entry to allocate the data set. Otherwise, it checks the next source.

SMPPARM member GIMDDALC
If a GIMDDALC control statement has been defined in SMPPARM member GIMDDALC, SMP/E uses that information to allocate the data set. Otherwise, it checks the next source.

Standard defaults
If the data set is for SMPOUT or SYSPRINT, SMP/E uses the standard default to allocate the data set. Otherwise, data set allocation fails.

Note:

1. When a data set is part of a concatenation (such as the SYSLIB concatenation), SMP/E does not do the previously described checking. All data sets in a concatenation must be defined the same way. For example, if a DDDEF entry specifies a concatenation, all the specified entries must also be defined by DDDEF entries.
2. For the cross-zone phase of APPLY and RESTORE processing, a DD statement cannot be used to allocate the SYSLIB for a cross-zone load module. This library can be allocated only through a DDDEF entry in the cross-zone containing the LMOD entry for the cross-zone load module.
3. The DD name SMPDUMMY is always allocated as a DUMMY data set. SMP/E ignores any allocation information specified for SMPDUMMY by a DDDEF entry or GIMDDALC member. If SMPDUMMY was previously allocated outside of SMP/E, SMP/E frees the SMPDUMMY DD and then reallocates it as a DUMMY data set.

Defining utility programs and associated parameters to SMP/E

SMP/E calls utility programs to install SYSMODs in the target and distribution libraries. These utilities must reside in either the LNKLST concatenation or the link pack area (LPA), as defined in SYS1.PARMLIB. SMP/E uses default utility programs unless you define OPTIONS entries and UTILITY entries specifying other utilities and parameters.

- OPTIONS entries point to the specific UTILITY entry to be used for each type of utility. These are the types of utilities you can point to:
  - Access methods services
  - Assembler
  - Compress
  - Copy
  - Link-edit
  - Retry after *x*37 abends

  - Update
  - Superzap
- Each UTILITY entry defines the information to be used when invoking a specific type of utility:
  - The name of the utility program
  - The maximum utility return code that SMP/E should consider to be successful
  - The ddname to be used for utility output
  - Parameters to be passed to the utility

Using default values for utility programs

If you do not define UTILITY entries and OPTIONS entries to specify which utility programs to use, SMP/E uses default utility programs, as well as its own default values, for return codes, print values, and the parameters to be passed. Table 5 on page 72 lists the default values for the various types of utility programs.

Table 5. Default values for UTILITY entries

Utility	NAME (see note 1)	RC	PRINT	PARM
Access method services	IDCAMS	0	SYSPRINT	
Assembler	ASMA90	4	SYSPRINT	XREF, NOOBJECT, DECK
Compress	IEBCOPY	0	SYSPRINT	
Copy	IEBCOPY	0	SYSPRINT	SPCLCMOD and CMWA=256K (for program elements and copied load modules)
Hierarchical file system copy	BPXCOPY	0	SYSPRINT (see note 3)	
Link-edit utility	IEWBLINK	8	SYSPRINT	LET, LIST, NCAL, XREF (see note 2)
Retry after *x*37 abends	IEBCOPY	0	SYSPRINT	
Update	IEBUPDTE	0	SYSPRINT	Determined by SMP/E during processing
Superzap	IMASPZAP	4	SYSPRINT	

Notes:

1. If you replace a default utility program, the replacement utility program must be compatible with the default utility it replaces, both in the way it processes any control statements and execution parameters generated by SMP/E and in the return codes that it returns to SMP/E.
2. When the load module being link-edited contains a CALLLIBS subentry list, SMP/E does not always use NCAL by default. In this case, SMP/E uses CALL for the link to the actual target library or NCAL for the link to the SMPLTS library. SMP/E always uses NCAL for ACCEPT processing.
3. If SYSTSPRT is specified as the PRINT value, it is ignored and the default of SYSPRINT is used instead.

Defining values for utility programs

If you want to use utility programs other than the defaults, or if you want to specify different parameters for the default utility programs, you need to identify the programs to SMP/E by defining UTILITY entries and OPTIONS entries. For example, the installation information for a particular product you are about to install may direct you to use a specific link-edit utility and may indicate that the maximum successful return code from the utility is 4, not 8.

Figure 31 on page 73 shows how OPTIONS entries, UTILITY entries, zone definition entries, and the SET command are related.

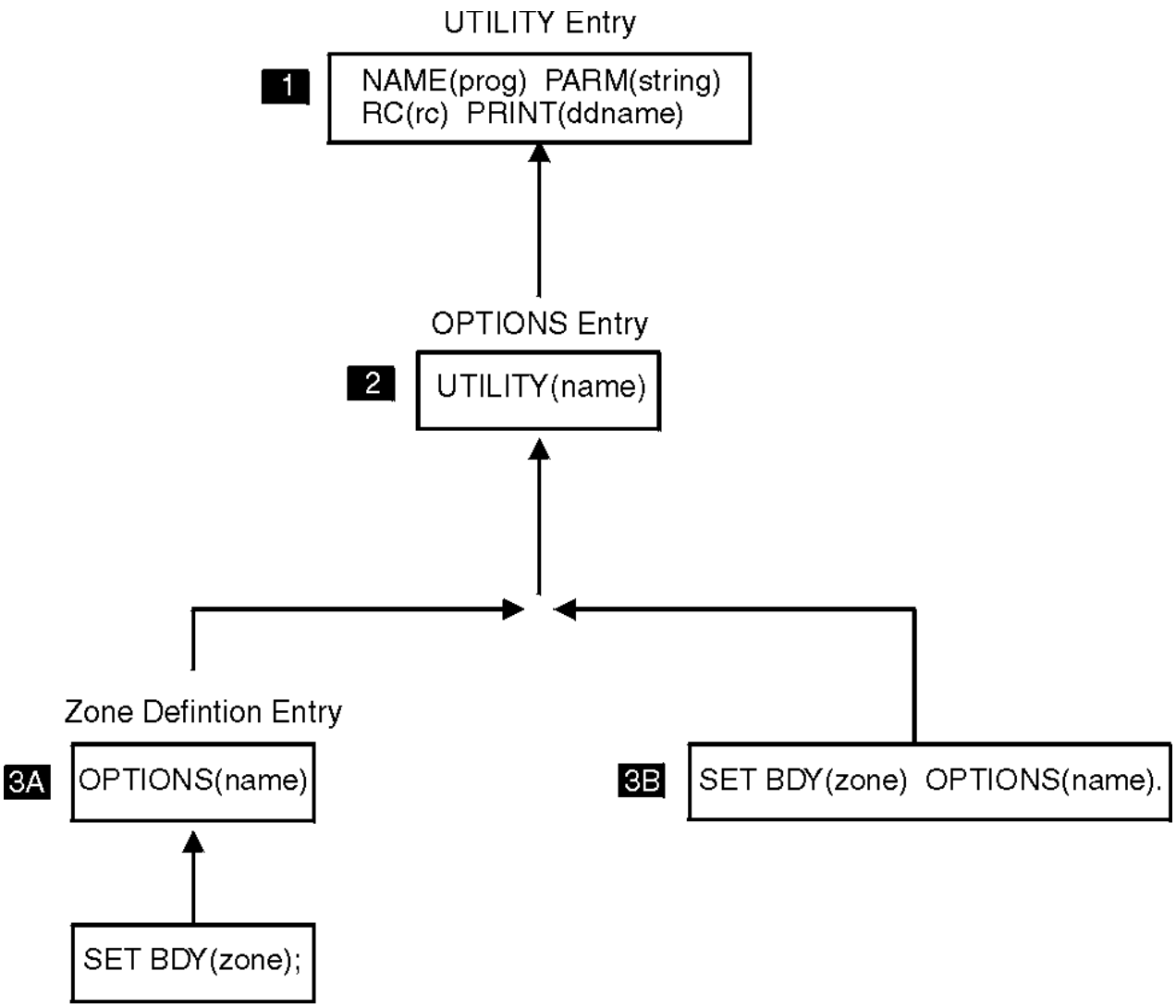


Figure 31. Relationships of OPTIONS, UTILITY, zone definition entries and the SET command

These are the basic steps to follow:

1
Define the desired utility name and parameters in a UTILITY entry.

2
Define an OPTIONS entry that points to that UTILITY entry.

3
Put the OPTIONS entry into effect by doing one of the following:

A

If the information should be the default for processing a particular zone, update the associated zone definition entry to point to the desired OPTIONS entry. The default OPTIONS ENTRY is always used for processing that zone, unless you override the OPTIONS entry with the SET command.

B

Otherwise, use the SET command to indicate which OPTIONS entry to use when processing the zone specified on the SET command. The information in the specified OPTIONS entry overrides the default OPTIONS entry defined for the zone.

For examples of these steps, see "Example: How to request the desired utility processing" on page 74. For more detailed information, see OPTIONS entry (global zone), UTILITY entry (global zone), GLOBALZONE entry (global zone), DLIBZONE entry (distribution zone), and TARGETZONE entry (target zone) in *z/OS SMP/E Reference*. For more information about the SET command, see The SET command in *z/OS SMP/E Commands*.

Note: You can specify which utility programs SMP/E can call by using the PROGRAM class of the z/OS Security Server (RACF). For more information, see Protecting programs in *z/OS Security Server RACF Security Administrator's Guide*.

Example: How to request the desired utility processing

Table 6 on page 74 describes the steps you need to follow in order to get the desired utility processing. For details on syntax and coding considerations, see UTILITY entry (global zone) in *z/OS SMP/E Reference*.

Table 6. How to request the utility processing

Steps	Sample scenario

1 Define the desired utility name and parameters in a UTILITY entry.	You want SMP/E to call a user routine, USERRCVR, rather than IEBCOPY, to recover from *x*37 abends. This is the processing you need:
	• The program must receive parameter TYPE=FAST.
	• The output should go to X37PRINT rather than SYSPRINT.
	• A return code of 4 or less is acceptable.
	• You want to suppress the listing of member names during retry processing done by your program.
	The following UCL (Update Control Language) defines the desired UTILITY entry for your program:
	`SET BDY(GLOBAL) /* Set to global zone. */.`
	`UCLIN /* */.`
	`ADD UTILITY(MYX37) /* Retry/recovery program. */`
	`NAME(USERRCVR) /* Program name. */`
	`PARM(TYPE=FAST) /* PARM value. */`
	`PRINT(X37PRINT) /* SYSPRINT ddname. */`
	`RC(4) /* Highest acceptable */`
	`/* return code. */`
	`LIST(NO) /* No list of member names. */`
	`/* */.`
	`ENDUCL /* */.`

2 Connect the UTILITY entry to an OPTIONS entry.	Once you have created the desired UTILITY entry, you need to point to it from an OPTIONS entry. The following UCL defines an OPTIONS entry (MYOPT1) that points to UTILITY entry MYX37:
	`SET BDY(GLOBAL) /* Set to global zone. */.`
	`UCLIN /* */.`
	`ADD OPTIONS(MYOPT1) /* New OPTIONS entry. */`
	`RETRY(MYX37) /* Connect to retry. */`
	`/* */.`
	`ENDUCL /* */.`

Table 6. How to request the utility processing (continued)

Steps	Sample scenario

3A Use the zone definition entry to specify your OPTIONS entry as the default OPTIONS entry.	You might want your OPTIONS entry to be the default for processing target zone TGT1. In this case, the TARGETZONE entry for TGT1 must point to your OPTIONS entry. The following UCL updates the existing TARGETZONE entry for TGT1 so it points to OPTIONS entry MYOPT1:
	`SET BDY(TGT1) /* Set to target zone, TGT1.*/.`
	`UCLIN /* */.`
	`REP TARGETZONE /* Update zone definition. */`
	`OPTIONS(MYOPT1) /* OPTIONS entry to be used.*/`
	`/* */.`
	`ENDUCL /* */.`

3B Use the SET command to have your OPTIONS entry override the default OPTIONS entry for the zone.	Instead of changing the default OPTIONS entry, you might want to override whatever the default might be and use OPTIONS entry (MYOPT1) for processing particular commands or SYSMODs. In this case, the SET command preceding the commands you want to process must point to your OPTIONS entry. The following UCL points the SET command to OPTIONS entry MYOPT1:
	`SET BDY(TGT1) /* Set to target zone, TGT1.*/`
	`OPTIONS(MYOPT1) /* OPTIONS entry used. */.`
	⋮

Recovering after errors from utility processing

To complete as many updates as possible to your product libraries, SMP/E tries to recover from errors that occur during processing by the utilities it invokes. The type of recovery SMP/E attempts for such errors depends on the type of error that occurred and the type of utility that was called.

- **Batched updates, no out-of-space problems.** Items to be processed by the link-edit utility or the copy utility are often batched. If an error other than an *x*37 abend occurs during such a utility call, SMP/E debatches the items and reinvokes the utility to attempt updates for individual members. This recovery is attempted automatically by SMP/E; no user intervention is required.
- **Out-of-space errors (*x*37 abends).** If a list of data sets is specified by the user, SMP/E attempts "retry" processing for data sets that have run out of space. During retry processing, the data set that ran out of space is compressed, and the utility is called again to retry the updates.

  If retry processing does not reclaim sufficient space and input to the utility was batched (copy or link-edit utility only), SMP/E debatches the input and retries the utility for each member separately. If this final attempt fails, the resulting *x*37 abend is treated as an unacceptable utility return code, and processing continues for other eligible updates.

This section explains what you need to define in order to have SMP/E attempt retry processing for *x*37 abends.

Overview of your input to retry processing

Your input to retry processing is through subentries in the OPTIONS entry, an optional retry exit routine, and the RETRY command operand.

OPTIONS entry

The RETRYDDN and EXRTYDD subentries in the OPTIONS entry indicate which libraries are eligible for retry processing.

The RETRYDDN subentry specifies which libraries should be included in retry processing. Without this list of libraries, SMP/E does not attempt retry processing.

The EXRTYDD subentry specifies which libraries should be excluded from retry processing. This makes it easier for you to include all but a few specific libraries in retry processing.

Exit routine
The retry exit routine enables you to control retry processing when an *x*37 abend occurs, instead of having SMP/E compress the out-of-space data set and reinvoke the failing utility.

If SMP/E determines that a retry can be attempted, it cancels the abend dump and calls the retry exit routine. The routine can then either cancel retry processing or perform some other method of recovery.

RETRY operand
The RETRY operand tells SMP/E whether to attempt retry processing for the specific SMP/E command that is being processed. RETRY can be specified on the ACCEPT, APPLY, LINK LMODS, LINK MODULE, and RESTORE commands.

You do not need to specify this operand in order to request retry processing, because the default is RETRY(YES). However, you can explicitly specify RETRY(YES) if you want to.

To prevent retry processing for a specific command, specify RETRY(NO) instead of using RETRY(YES).

Example: How to request the desired retry processing

Table 7 on page 76 describes the steps you need to follow in order to get the desired retry processing. For details on syntax and coding considerations, see *z/OS SMP/E Reference*.

Table 7. How to request the retry processing

Steps	Sample scenario
1 Decide which data sets should be included in or excluded from retry processing.	You want retry processing for all libraries except LINKLIB, MIGLIB, and NUCLEUS.

2 Define the necessary subentries in the OPTIONS entries that you plan to use.	You already have been using OPTIONS entry OPT1 for all your ACCEPT, APPLY, LINK LMODS, LINK MODULE, and RESTORE commands. You need to add RETRYDDN and EXRTYDD values to specify the libraries that are eligible for retry processing.
	`SET BDY(GLOBAL).`
	`UCLIN .`
	`ADD OPTIONS(OPT1)`
	`RETRYDDN(ALL)`
	`EXRTYDD(LINKLIB,MIGLIB,`
	`NUCLEUS)`
	`.`
	`ENDUCL .`

3 Decide whether to use the default RETRY UTILITY entry, or to point to and define a RETRY UTILITY entry that specifies the desired information.	You will use the defaults for the retry utility as shown in Table 5 on page 72.
4 If desired, define an exit routine for retry processing. The section for Retry Exit Routine in *z/OS SMP/E Reference* provides all the information you need about the interface for this routine and what SMP/E expects the routine to do.	You will use SMP/E's retry processing instead of writing a retry exit routine.

Table 7. How to request the retry processing (continued)

Steps	Sample scenario

5 Make sure the desired OPTIONS entry is in effect for the zone you are processing.
 	
The methods you can use are shown in Figure 31 on page 73.
	As indicated in step 2, you have been using OPTIONS entry OPT1. Instead of specifying it on the SET command, you have defined TZONE entry (TGT1) and DZONE entry (DLIB1), which specify OPT1 as the OPTIONS entry to be used. (These zones are already defined in the global zone by ZONEINDEX subentries.)
	
	`SET BDY(TGT1).`
	`UCLIN .`
	`ADD TARGETZONE(TGT1)`
	`OPTIONS(OPT1)`
	`SREL(Z038)`
	`RELATED(DLIB1)`
	`.`
	`ENDUCL .`
	`SET BDY(DLIB1).`
	`UCLIN .`
	`ADD DLIBZONE(DLIB1)`
	`OPTIONS(OPT1)`
	`SREL(Z038)`
	`RELATED(TGT1)`
	`.`
	`ENDUCL .`

6 Use RETRY(YES) on the commands you want retry processing to be done for (RETRY can be specified on the ACCEPT, APPLY, LINK LMODS, LINK MODULE, and RESTORE commands.).
 	
Note: To prevent retry processing for a specific command, specify RETRY(NO) instead.
	You are installing product HYY2102 and the related service, and you want SMP/E to attempt retry processing. You choose to use RETRY(YES) as the default instead of specifying it explicitly.
	
	`SET BDY(TGT1).`
	`APPLY FORFMID(HYY2102)`
	`FUNCTIONS PTFS`
	`GROUPEXTEND`
	`.`

Connecting SMP/E dialogs to ISPF

The SMP/E dialogs run under ISPF. Therefore, if you plan to use the dialogs, you must connect them to ISPF. Follow these steps:

1. Make sure you have the required programs on your system.
2. If you have the TSO/VS2 programming control facility (PCF), add the necessary dialog modules to the PCF command table.
3. Concatenate the dialog libraries in a logon procedure or a command list (CLIST).
4. Connect the dialogs to ISPF.
5. Customize the dialogs, if desired.

These steps are described in the topics that follow.

Check for required programs

Make sure the following programs are on your system:

- ISPF (Version 4 Release 2, or later)
- ISPF/PDF (Version 2 Release 3, or later)
- TSO/E (Version 2 Release 5, or later)

The SMP/E dialogs use the TSO OUTPUT, SUBMIT, and STATUS commands. Therefore, to support these commands, you must provide a TSO IKJEFF53 user installation exit routine that does not restrict the TSO

OUTPUT and STATUS commands to job names beginning with the user's user ID. For details, see *z/OS TSO/E Customization*.

Add dialog modules to the PCF command table

If you have the TSO/VS2 programming control facility (PCF), add the following modules to the PCF command table:

GIMPSBTP
GIMISID1
GIMISID2
GIMISID3
GIMISID4
GIMISID5

Concatenate the dialog libraries

Table 8 on page 78 lists the SMP/E dialog libraries needed to concatenate with ISPF libraries To use the dialogs, concatenate these libraries in a TSO logon procedure or in a CLIST. Here are some recommendations for concatenating the libraries:

- There are two sets of dialog libraries for SMP/E: one for English and one for Japanese. You can include the libraries for only one of these features in a given logon procedure or CLIST. If you want to be able to use both features on the same system, you need a logon procedure or CLIST for each feature. To switch from one feature to the other, you have to exit ISPF, run the procedure or CLIST for the other feature, and get back into ISPF.
- Do not use the ISPF LIBDEF service to concatenate the following libraries. Instead, use either the TSO ALLOCATE command or DD statements and JCL.
  - Dialog CLISTs (GIM.SGIMCLS0): If you use LIBDEF, the CLISTs and EXECs in the data set cannot be executed.
  - Dialog load module library (GIM.SGIMLMD0): If LIBDEF is used to concatenate the load module libraries, the SMP/E dialogs cannot find required load modules.

Table 8. ISPF libraries and related SMP/E target libraries

DDNAME	SMP/E library	Contents
SYSPROC	GIM.SGIMCLS0	Dialog CLISTs
ISPLLIB	GIM.SGIMLMD0	Dialog load modules
ISPMLIB	GIM.SGIMMENU	Dialog messages
ISPPLIB	GIM.SGIMPENU	Dialog panels
ISPSLIB	GIM.SGIMSENU	Skeleton JCL procedures
ISPTLIB	GIM.SGIMTENU (see Note 1)	Table input library
ISPCTL1	(see Note 2)	Generated JCL
ISPCTL2	(see Note 2)	Generated JCL
SMPTABL	A user-defined name (see Note 3)	SMP/E table library

Note:

1. Include GIM.SGIMTENU and the SMPTABL data set in the ISPTLIB concatenation.

2. Use the ISPCTL1 and ISPCTL2 files to generate JCL for submitted SMP/E jobs. The SMP/E job submit facility lets you browse and edit this JCL. You can omit these files from your logon procedure and let ISPF automatically allocate them as needed.

   To save the input JCL generated by the dialogs, either:

   - Use EDIT CREATE while in the generated JCL to save it in another (permanent) data set, or
   - Allocate a permanent sequential data set to ISPCTL1 (LRECL=80, RECFM=FB) before you enter the SMP/E dialogs.

3. Allocate a single, installation-wide table data set to the ISPTLIB and SMPTABL DD statements.
   - SMP/E uses this table data set to save process status information for the SYSMOD management dialogs.
   - The data set must be a partitioned data set (LRECL=80, RECFM=FB). Because the data set is also in the concatenation of ISPTLIB, make the block size compatible with the block size of the corresponding ISPF data sets.
   - Ensure all users of the SMP/E dialogs have the appropriate access defined in your security product to update this table data set.
   - For more information about the SMPTABL data set, see "SMPTABL space allocation" on page 79.

SMPTABL space allocation

The amount of space required for SMPTABL depends on the number of concurrent processes you want to support for the SMP/E SYSMOD management dialog and on the amount of data that is saved for each process. Use Table 9 on page 79 as a guide.

Table 9. SMPTABL data set allocations

Number of processes	BLKSIZE	Number of blocks	Directory blocks
16	8800	60	4
32	8800	120	8

Sample logon procedure

Figure 32 on page 80 is an example of a logon procedure that concatenates the libraries for the SMP/E dialogs.

Note: Depending on the system you are connecting the dialogs to, you may need to change your logon procedure, a CLIST, or both. It is common practice to have a logon procedure call a CLIST every time a user logs on. This is normally done so that minor changes to concatenations do not require changes to the logon procedure, or so users with the same logon procedure can have the CLIST perform different allocations from the ones performed for other users.

If you make changes only to your logon procedure and the procedure calls a CLIST that changes a concatenation, you may end up missing changes you made in your logon procedure. In this case, instead of (or in addition to) updating the concatenations in your logon procedure, you should update the concatenations in the CLIST that is called.

```
//SMPE     EXEC PGM=IKJEFT01
//SYSPROC  DD DSN=GIM.SGIMCLS0,DISP=SHR     /* SMP/E CLISTs. */
//         DD DSN=ISP.SISPCLIB,DISP=SHR
//SYSHELP  DD DSN=SYS1.HELP,DISP=SHR
//SYSPRINT DD TERM=TS
//SYSIN    DD TERM=TS
//*************************************************************
//*
//* ISPF temporary data sets (no ISPCTL1 or ISPCTL2)
//*
//*************************************************************
//ISPLST1 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//        DCB=(LRECL=121,BLKSIZE=0,RECFM=FBA)
//ISPLST2 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//        DCB=(LRECL=121,BLKSIZE=0,RECFM=FBA)
//*************************************************************
//*
//* ISPF/SMPE load modules, panels, skeletons, messages,
//*           tables
//*    NOTE:  SMPTABL is a single installation-wide
//*           ISPF table data set shared by all SMP/E
//*           users. It is initially allocated as an empty PDS
//*           LRECL=80, RECFM=FB, BLKSIZE= (compatible with
//*           ISP.V2R3M0.ISRTLIB).
//*
//*************************************************************
//ISPLLIB DD DSN=GIM.SGIMLMD0,DISP=SHR     /* SMP/E LMODs.  */
//ISPPLIB DD DSN=GIM.SGIMPENU,DISP=SHR     /* SMP/E panels. */
//        DD DSN=ISP.SISPPENU,DISP=SHR
//ISPSLIB DD DSN=GIM.SGIMSENU,DISP=SHR   /* SMP/E skeletons.*/
//        DD DSN=ISP.SISPSENU,DISP=SHR
//ISPMLIB DD DSN=GIM.SGIMMENU,DISP=SHR   /* SMP/E messages. */
//        DD DSN=ISP.SISPMENU,DISP=SHR
//ISPTLIB DD DSN=GIM.SGIMTENU,DISP=SHR     /* SMP/E tables. */
//        DD DSN=ISP.SISPTENU,DISP=SHR
//        DD DSN=user-defined name,DISP=SHR /* SMP/E tables.*/
//SMPTABL DD DSN=user-defined name,DISP=SHR /* SMP/E tables.*/
```

Figure 32. Sample logon procedure that concatenates SMP/E and ISPF libraries

Connect the dialogs to ISPF

You can get to the SMP/E dialogs from the SMP/E primary option menu. To provide access to that menu, you must connect the dialogs to ISPF. Sample ISPF panels are provided with z/OS to enable panels for most z/OS elements, including SMP/E. These panels are in the SISPPENU data set after APPLY processing. The panels supplied are:

ISR@390
This is an ISPF Primary Options Menu. It is identical to the ISR@PRIM Primary Options Menu, except that it includes the additional options 12 and 13, which point to the next two panels.

ISR@390S
This is a secondary panel, with options used by system programmers and administrators, including SMP/E.

ISR@390U
This is a secondary menu panel, including the options used by most ISPF users.

Use one of the methods documented in the *Program Directory* for this level of SMP/E to use ISR@390 instead of ISR@PRIM. The SMP/E dialogs will then be accessible through panel ISR@390S.

Customize the SMP/E dialogs

After you install the SMP/E dialogs, you can change the default values they use.

When you select option 0 (SETTINGS) on the SMP/E Primary Option Menu, SMP/E displays panel GIM@PARM, which allows you to

- Modify the values for allocating temporary SMP/E utility (SYSUT*n*) and SMP/E work (SMPWRK*n*) data sets . The options you specify are saved permanently in the ISPF profile pool for use later by other SMP/E dialog processes.

- Specify a data set name to be used by SMP/E for the SMPPARM data set during background execution. The SMPPARM data set is used to define exit routines and specify allocation information used by SMP/E processing. If a data set name is specified on panel GIM@PARM, SMP/E generates an SMPPARM DD statement in all JCL jobs created by the SMP/E dialogs that invoke SMP/E.

```
GIM@PARM                      SMP/E DIALOG SETTINGS
===>

Reset to use SMP/E's default settings? ===> NO  (Yes or No)

Enter or verify the information below used to allocate temporary data sets.
These temporary data sets contain generated JCL jobs, output for jobs which
have completed execution, and MCS entries for viewing:

   UNIT   ===> SYSALLDA
   VOLUME ===>

Enter or verify the information below used to create DD statements in
generated JCL jobs:

   Space for SMP/E Utility Work data sets (SYSUTn):
             Block Size  Primary  Secondary
     SYSUT1     3120       380       760
     SYSUT2     3120       380       760
     SYSUT3     3120       380       760
     SYSUT4     3120        38       100

   Space for SMP/E Work data sets (SMPWRKn):
             Block Size  Primary  Secondary  Directory Blocks
     SMPWRK1    3120       364       380          500
     SMPWRK2    3120       364       380          500
     SMPWRK3    3120       364       380          500
     SMPWRK4    3120       364       380          500
     SMPWRK6    3120       364       380          500

   Unit for SYSUTn and SMPWRKn data sets:
     UNIT ===> SYSDA

   Data set name to use for SMPPARM.  If a name is specified, an SMPPARM DD
   statement will be generated.
     DSN  ===>

Enter or verify the information below used to allocate permanent SMPWRK3
data sets created and used by the SYSMOD Management dialogs:

   UNIT   ===>
   VOLUME ===>
   PREFIX ===>          (TSO Prefix is used if no prefix is specified)

To save the changes, press ENTER .
To ignore the changes, enter END .
```

JOB statement customization

If you have never entered a JOB statement on panels GIMCGSUB, GIMRCSUB, or GIMSB01, then SMP/E primes those panels with the following default JOB statement:

```
//useridA  JOB (ACCOUNT),'NAME'
//*
//*
//*
```

This initial default JOB statement is not customizable, but when you enter a JOB statement on panels GIMCGSUB, GIMRCSUB, or GIMSB01, the statement you enter is saved in your ISPF profile pool and will be used as the new default when you use those panels again.

Setting up SMP/E for easier operation

SMP/E provides several optional facilities that you can use to make SMP/E operations easier and more efficient. To take advantage of these facilities, you must setup a few SMP/E options. Normally, these set up procedures need only be done once.

The major tasks are:

- Specifying SMP/E OPTIONS entry
- Specifying link edit utility output DDDEF entries
- Specifying automatic cross-zone requisite checking

Recommended values for OPTIONS entry

IBM recommends the following OPTIONS entry values:

Recommended value
Purpose

MSGFILTER(YES)
MSGFILTER(YES) causes SMP/E to filter the messages it writes to SMPOUT during APPLY, ACCEPT, and RESTORE command processing. When SMP/E filters messages, most non-critical informational messages are not written to SMPOUT. The result is less output to read through when it is necessary to investigate an SMP/E operation. MSGFILTER(NO) is the default.

MSGWIDTH(80)
MSGWIDTH(80) causes SMP/E to format its messages to an 80 character width. This makes online viewing simpler by eliminating the need to scroll right to view the entire message text. MSGWIDTH(120) is the default.

RECZGRP
Often the RECEIVE command will receive a PTF that has already been accepted and purged from the global zone and SMPPTS data set. There is no need to receive such PTFs and they only add to the space used by the SMPPTS. To prevent RECEIVE from receiving such PTFs, you need to tell SMP/E what dlib zones to check when determining if a PTF has already been accepted. You can specify the list of dlib zones using the RECEIVE Zone Group (RECZGRP) subentry in an OPTIONS entry.

The RECZGRP subentry allows you to set a policy and specify the list of zones once. This list is then used for all future RECEIVE operations whenever the OPTIONS entry is active. With the list of dlib zones set, during RECEIVE processing, SMP/E will check each of the zones specified first before receiving a PTF. If that PTF is accepted in any of the specified zones, the PTF will not be received again.

RETRYDDN(ALL)
RETRYDDN(ALL) causes SMP/E to compress out-of-space libraries and retry processing after an *x*37 abend. When you use this option, make sure you are **not** updating production data sets.

Note: Do not specify a PEMAX value. Allow SMP/E to use its default value of 25,000.

Sample UCLIN job

Here is a sample UCLIN job to build an OPTIONS entry with the recommended values:

```
//job      JOB accounting info…
//step     EXEC PGM=GIMSMP
//SMPCSI   DD DSN=smp.global.csi,DISP=SHR
//SMPCNTL  DD *
  SET BDY(GLOBAL).
  UCLIN.
  ADD OPTIONS(OPTENT)
      MSGFILTER(YES)
      MSGWIDTH(80)
      RETRYDDN(ALL)
      RECZGRP( zosdlib
               os390dlib
               jes2dlib
               jes3dlib
               cicsdlib
               db2dlib
               imsdlib ).
  ENDUCL.
/*
```

Activating the OPTIONS entry

After the OPTIONS entry has been defined, IBM recommends that you make it active by defining it as the default OPTIONS entry for the global, target, and DLIB zones. Otherwise, you must specify it on the SET command before using any other SMP/E command.

Recommended DDDEF entries for link-edit utility output

To exploit utility multitasking in SMP/E, ensure that the ddname that is to contain the link-edit utility output is defined with either a DDDEF entry that identifies a SYSOUT class or an entry in the SMPPARM GIMDDALC member that identifies a SYSOUT class. SMP/E's default ddname for utility output is SYSPRINT, but can be changed using the PRINT subentry of the LKED UTILITY entry. When multitasking, SMP/E will invoke multiple instances of the link-edit utility at the same time, thus decreasing the total time required to complete an ACCEPT, APPLY, LINK LMODS, or RESTORE command. Multitasking of link edit can occur when there are different target libraries, and there are no dependencies on previous and subsequent link edits. If you do not define the print ddname using either a DDDEF entry or an SMPPARM GIMDDALC member, or if the print ddname definition identifies something other than a SYSOUT class, or if you override the SYSPRINT DDDEF with a ddname in your JCL, SMP/E will not multitask link-edit utility operations.

Specifying automatic cross-zone requisite checking

The installation of software service often requires the synchronization of service levels across multiple SMP/E zones. For example, service for software in the MVS zone may require related service for the JES2, CICS, DB2®, and other zones to permit all software within the system image to operate properly. To help ensure proper synchronization across zones, you can tell SMP/E to automatically check for cross-zone requisites during APPLY, ACCEPT, and RESTORE command processing.

To enable automatic cross-zone requisite checking, you must tell SMP/E which zones contain software to be checked for requisites. The set of zones identified for cross-zone requisite checking is called the *zone group*. SMP/E provides two methods to identify the zones within the zone group:

1. Define a default zone group.
2. Specify the zones directly on the APPLY, ACCEPT, or RESTORE command.

Defining a default zone group

You can define a default zone group by creating a ZONESET entry that contains the XZREQCHK(YES) subentry and the list of zones to be included in the default zone group. SMP/E will use this default zone group to determine which zones to check for requisites whenever the APPLY, ACCEPT, or RESTORE commands process a zone named in this ZONESET. To create such a ZONESET, use the SMP/E Administration Dialogs or use the UCLIN command, as in this example:

```
//job      JOBaccounting info…
//step     EXEC PGM=GIMSMP
//SMPCSI   DD DSN=smp.global.csi,DISP=SHR
//SMPCNTL  DD *
  SET BDY(GLOBAL).
  UCLIN.
    ADD ZONESET(ZONEGRP)
        XZREQCHK(YES)
/* use this ZONESET for cross-zone req checking */
        ZONE(zostgt   zosdlib
             os390tgt os390dlib
             jes2tgt  jes2dlib
             jes3tgt  jes3dlib
             cicstgt  cicsdlib
             db2tgt   db2dlib
             imstgt   imsdlib).
  ENDUCL.
/*
```

The ZONESET should contain the names of all the zones to be checked for cross-zone requisites. Once the ZONESET is created and the XZREQCHK(YES) subentry is set, the zones defined in the ZONESET are used

as the default zone group any time the APPLY, ACCEPT, or RESTORE commands process any zone found in the ZONESET. For example, if an APPLY command is initiated for the *cicstgt* zone, all zones found in the ZONESET entry named ZONEGRP are used for the zone group.

Specifying the zone group on a command

If you do not have a default zone group defined, or you want to use a different set of zones for the zone group, you can specify the zones on the APPLY, ACCEPT, or RESTORE command using the XZGROUP operand. This is simply a matter of specifying the zones to be checked for cross-zone requisites, as shown in this example:

```
//job      JOB accounting info…
//step     EXEC PGM=GIMSMP
//SMPCSI   DD DSN=smp.global.csi,DISP=SHR
//SMPCNTL  DD *
  SET BDY(zostgt).
  APPLY SOURCEID(HIPER)
        CHECK
        XZGROUP(os390tgt,jes2tgt,jes3tgt,cicstgt,db2tgt,imstgt)
        BYPASS(HOLDSYS).
/*
```

Define a ZONEINDEX for each zone

Each of the zones specified in a ZONESET or on the XZGROUP operand must be defined by a ZONEINDEX in the current global zone, even if the zones are already defined in another global zone (more than one global zone may contain a ZONEINDEX for the same target or dlib zone). This allows the APPLY, ACCEPT, and RESTORE commands initiated from the current global zone to access the specified zones. To add ZONEINDEX subentries for each of the zones, use the SMP/E Administration Dialogs or use the UCLIN command, as in this example:

```
//job      JOB accounting info…
//step     EXEC PGM=GIMSMP
//SMPCSI   DD DSN=smp.global.csi,DISP=SHR
//SMPCNTL  DD *
  SET BDY(GLOBAL).
  UCLIN.
    ADD GLOBALZONE ZONEINDEX(
        (zostgt,zos.target.csi,TARGET)
        (zosdlib,zos.dlib.csi,DLIB)
        (os390tgt,os390.target.csi,TARGET)
        (os390dlib,os390.dlib.csi,DLIB)
        (jes2tgt,jes2.target.csi,TARGET)
        (jes2dlib,jes2.dlib.csi,DLIB)
        (jes3tgt,jes3.target.csi,TARGET)
        (jes3dlib,jes3.dlib.csi,DLIB)
        (cicstgt,cics.target.csi,TARGET)
        (cicsdlib,cics.dlib.csi,DLIB)
        (db2tgt,db2.target.csi,TARGET)
        (db2dlib,db2.dlib.csi,DLIB)
        (imstgt,ims.target.csi,TARGET)
        (imsdlib,ims.dlib.csi,DLIB)
        ).
  ENDUCL.
/*
```

Cross-zone requisite checking

Whether you define a default zone group or specify a zone group on the APPLY, ACCEPT, and RESTORE command, SMP/E will determine during command processing whether any cross-zone requisites are unsatisfied. Cross-zone requisites are caused by ++IF statements, where a SYSMOD containing a ++IF statement resides in one zone and the function identified on the ++IF resides in another zone. If the requisite identified on the ++IF statement does not reside in the same zone as the identified function, then the condition is not satisfied.

Unsatisfied cross-zone requisite conditions will cause APPLY, ACCEPT, and RESTORE command processing to fail for the SYSMOD containing the ++IF statement. Processing continues to fail until

the requisite is satisfied in the other zone, unless the BYPASS(XZIFREQ) operand is specified on the command.

Bypassing unsatisfied cross-zone requisites

The BYPASS(XZIFREQ) operand on the APPLY, ACCEPT, and RESTORE commands tells SMP/E to continue processing the command even if missing cross-zone requisites are detected. SMP/E warning messages are issued to identify the missing cross-zone requisites.

```
//job      JOB accounting info…
//step     EXEC PGM=GIMSMP
//SMPCSI   DD DSN=smp.global.csi,DISP=SHR
//SMPCNTL  DD *
  SET BDY(zostgt).
  APPLY SOURCEID(HIPER)
        CHECK
        BYPASS(HOLDSYS
               XZIFREQ).
/*
```

Note: This example assumes that a default zone group has been defined and will be used during APPLY command processing.

You can be broad or granular in the specification of what cross-zone requisites to bypass. You can indicate all cross-zone requisites are to be bypassed (as in the previous example), you can indicate that specific cross-zone requisite SYSMODs are to be bypassed, or you can indicate that only specific cross-zone requisite SYSMODs from specific zones are to be bypassed. Details of the BYPASS(XZIFREQ) operand and processing can be found in *z/OS SMP/E Commands*.

Resolving cross-zone requisites

If cross-zone requisites are bypassed and therefore cause unsatisfied cross-zone requisites, you must resolve those unsatisfied requisites. To do this, you need to APPLY or ACCEPT those requisites to the appropriate zones. To aid in this task, SMP/E provides a method to identify missing cross-zone requisite SYSMODs and make them candidates for APPLY and ACCEPT processing to resolve missing cross-zone requisites.

In order to select cross-zone requisite SYSMODs to be installed in a particular zone, the XZREQ operand can be used on the APPLY and ACCEPT commands. The XZREQ operand causes SMP/E to search the zones in the zone group for unsatisfied cross-zone requisites. If any are found which can be satisfied by installing a requisite SYSMOD to the current zone, those SYSMODs are made candidates for the APPLY and ACCEPT commands. Here is an example:

```
//job      JOBaccounting info…
//step     EXEC PGM=GIMSMP
//SMPCSI   DD DSN=smp.global.csi,DISP=SHR
//SMPCNTL  DD *
  SET BDY(cicstgt).
  APPLY CHECK
        BYPASS(HOLDSYS)
        XZREQ.
/*
```

Note: This example assumes that a default zone group was defined and will be used during APPLY command processing.

Using the XZREQ operand identifies and installs the needed cross-zone requisites. You can also use the REPORT CROSSZONE command to identify the needed cross-zone requisites. See the section on the REPORT CROSSZONE command in *z/OS SMP/E Commands* for details.

Defining the information needed to invoke SMP/E

There are several ways to call SMP/E after it has been installed:

- Use the SMP/E dialogs.

- Submit a background job that calls GIMSMP, the program name for SMP/E. This job can call SMP/E either directly or in a cataloged procedure.

This section describes the types of information you need to provide if you use a cataloged procedure to invoke SMP/E. It discusses required JCL statements and a sample cataloged procedure for SMP/E.:

Required JCL statements

Unless you are using the SMP/E dialogs, you must provide the following JCL statements to invoke SMP/E:

- A JOB statement
- An EXEC statement
- DD (data definition) statements

JOB statement

The JOB statement describes your installation-dependent parameters. The JOB statement (or the EXEC statement, or both) can also include the REGION parameter to set the size of the region in which SMP/E runs. For details, see *z/OS MVS JCL User's Guide* or *z/OS MVS JCL Reference*.

Note: To ensure that the maximum space above 16 megabytes is available for SMP/E processing, it is recommended that you specify REGION=0M. To allow long running SMP/E operations to complete, consider specifying TIME=1440 or TIME=NOLIMIT.

EXEC statement

The EXEC statement must specify PGM=GIMSMP or the name of your cataloged procedure for calling SMP/E. (For an example of a cataloged procedure, see "Sample cataloged procedure for SMP/E" on page 87.) The following can be specified in the EXEC statement PARM parameter:

COMPAT=WARNBYPASS or
COMPAT=NOWARNBYPASS
The COMPAT parameter is used to control incompatible behaviors of SMP/E processing.

COMPAT(WARNBYPASS)
Indicates that the APPLY and ACCEPT commands will issue warning messages to identify bypassed SYSTEM HOLD exceptions. This is the behavior for releases of SMP/E prior to V3R5.

COMPAT(NOWARNBYPASS)
Indicates that the APPLY and ACCEPT commands will issue informational messages to identify bypassed SYSTEM HOLD exceptions. If neither COMPAT(WARNBYPASS) or COMPAT(NOWARNBYPASS) is specified, the default is COMPAT(NOWARNBYPASS).

CSI= *dsname*
where *dsname* is the name of the CSI data set containing the global zone. (This data set is also known as the master CSI.) This parameter is used to enable SMP/E to allocate the master CSI data set dynamically.

Note: If there is an SMPCSI DD statement, the CSI=*dsname* operand is not allowed. If both are specified, SMP/E does not run.

DATE= *date*
where *date* can be one of the following options::

U or IPL
To use the IPL date of the system.

REPLY
To request the date from the operator. As a result, SMP/E issues message GIM399I.

yyddd
To specify a specific date, where *yy* is the year and *ddd* is the day of the year (the Julian date).

If DATE is not specified, the IPL date of the system is used.

PROCESS=WAIT or
PROCESS=END

The PROCESS parameter is used to control how long a job should wait if a CSI or PTS data set is not immediately available because it is currently being used either by another job or by a dialog.

- WAIT causes the job to wait until the data set is available. A message is issued to the system operator every 30 minutes while the job is waiting.
- END causes the job to wait for 10 minutes. If the data set is still not available after the 10-minute wait, the command requiring the data set is stopped.

If PROCESS is not specified, the default is PROCESS=WAIT.

For more information about obtaining and sharing CSI data sets, see Sharing SMP/E data sets in *z/OS SMP/E Commands*

Processing of the PTS data set is also affected by the WAITFORDSN value specified in its DDDEF entry. WAITFORDSN determines whether SMP/E should wait to allocate a data set that is not immediately available. If the DDDEF entry specifies WAITFORDSN=NO (or lets this value default to NO) and the data set is not available, allocation of the data set fails, regardless of the PROCESS value specified on the EXEC statement. If WAITFORDSN=NO, SMP/E does not wait to retry allocation of the data set.

For example, suppose a PTS with a disposition of OLD is already being used by a job, and a second job tries to access the same PTS data set by allocating it through a DDDEF entry. The DDDEF entry used by the second job for the PTS specifies WAITFORDSN=NO. As a result, allocation of the PTS fails for the second job.

DD statements

DD statements define the data sets that can be used in SMP/E processing. For information about the data sets required for each command, see the individual SMP/E commands in *z/OS SMP/E Commands*.

Note:

1. You can use DDDEF entries, rather than DD statements, to allocate many of the necessary data sets. For more information, see "How to dynamically allocate data sets to be used during SMP/E processing" on page 69.
2. To use a target system's instance of SMP/E, you can specify the STEPLIB DD statement. If you specify the SYS1.MIGLIB and ASM.SASMMOD1 data sets of the target system on the STEPLIB DD statement, you can safely use the target system's instance of SMP/E and system utilities, such as the assembler and binder.

Sample cataloged procedure for SMP/E

Figure 33 on page 89 is a sample cataloged procedure, SMPPROC, that can be used to run SMP/E. The numbers to the left of the statements correspond to the notes that follow the example. When you write a cataloged procedure for SMP/E, remember the following:

- Tailor your own cataloged procedure to fit your system and processing requirements.
- You may want to use a single procedure for all SMP/E processing, or you may want to define multiple procedures for specific SMP/E commands and include in each one just those DD statements required for that command. For example, a special procedure for RECEIVE might include the SMPPTFIN DD statement but no DD statements for the target and distribution libraries.

  Note: The SYSLIB concatenations for APPLY and ACCEPT should point to different libraries.
- Most of the data sets in the cataloged procedure can be allocated without DD statements. If you use the methods described for the data sets listed later in this section, you might not need a cataloged procedure.

  Master CSI data set
  The master CSI data set can be specified on the CSI= parameter of the EXEC statement for GIMSMP, rather than on the SMPCSI DD statement. For more information about parameters you can specify on the EXEC statement, see "Required JCL statements" on page 86.

Target and distribution zones
CSI data sets for target and distribution zones are normally dynamically allocated with zone indexes in the global zone. If you want to use the batch local shared resources (BLSR) subsystem, you must supply your own JCL statements. For examples of defining zone indexes and of specifying JCL for BLSR, see the SMPCSI section in *z/OS SMP/E Reference*.

Other data sets
Other data sets in the cataloged procedure can be defined with DDDEF entries. When you use DDDEF entries, only the data sets SMP/E needs for a particular run are allocated.

When you use DD statements, all the data sets defined must be online and allocated. Therefore, you might want to use a combination of DDDEF entries and a cataloged procedure shorter than the one in Figure 33 on page 89. For more information about DDDEF entries, see *z/OS SMP/E Reference*.

- Although they are not shown in the sample cataloged procedure, the following DD statements may also be required:
  - An SMPCNTL DD statement, pointing to the commands that SMP/E processes, is required by all commands.
  - An SMPPTFIN DD statement, pointing to the source of the SYSMODs that are processed, is required by the RECEIVE command.
  - An SMPHOLD DD statement, pointing to the source of the HOLDDATA that is processed, is required by the RECEIVE command.
  - An SMPPTS DD statement should be coded with `DISP=SHR`. This allows concurrent jobs to share the PTS as much as possible. For more information about how SMP/E shares data sets, see the topic on sharing SMP/E data sets in *z/OS SMP/E Commands*.
  - The SMPLTS data set is required when processing load modules with CALLLIBS.
  - An SMPMTS DD statement is required for changes to macros that do not reside in a target library.
  - An SMPSTS DD statement is required for changes to source code that does not reside in a target library.
  - If any of the SYSMODs being installed contain elements packaged with the LKLIB operand, a DD statement for the ddname specified on that operand is required by the APPLY and ACCEPT commands.
  - If any of the SYSMODs being installed contain elements or JCLIN packaged with the TXLIB operand, a DD statement for the ddname specified on that operand is required by the APPLY and ACCEPT commands.
- If any of the required data sets (such as the SMPPTFIN) are not defined in the cataloged procedure or by DDDEF entries, they must be specified in the JCL used to call SMP/E.
- For more information about the data sets required by each SMP/E command, see *z/OS SMP/E Commands*.

```
     //SMPPROC  PROC
     //SMP      EXEC PGM=GIMSMP
     //* -----  SYSOUT data sets --------------------------------
     //SMPOUT   DD SYSOUT=A
     //SMPRPT   DD SYSOUT=A
     //SMPLIST  DD SYSOUT=A
     //SMPSNAP  DD SYSOUT=A
     //SMPDEBUG DD SYSOUT=A
     //SYSPRINT DD SYSOUT=A
 1   //SMPPUNCH DD SYSOUT=B
     //*------  SMP/E data sets -------------------------------
     //SMPLOG   DD DSN=SYS1.SMPLOG,DISP=MOD
     //SMPLOGA  DD DSN=SYS1.SMPLOGA,DISP=MOD
 2   //* -----  Master CSI  ---------------------------------
     //SMPCSI   DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
     //SMPDATA1 DD DSN=MVSTGT1.SMPDATA1,DISP=MOD
     //SMPDATA2 DD DSN=MVSTGT1.SMPDATA2,DISP=MOD
 3   //* -----  SMP/E temporary data sets-----------------------
     //SMPWRK1  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
     //            DCB=BLKSIZE=6160
     //SMPWRK2  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
     //            DCB=BLKSIZE=6160
 4   //SMPWRK3  DD DSN=data set name,
                 UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,CATLG),
     //            DCB=BLKSIZE=3120
     //SMPWRK4  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
     //            DCB=BLKSIZE=3120
     //SMPWRK6  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
     //            DCB=BLKSIZE=6160
     //* -----  Utility data sets ----------------------------
     //SYSUT1   DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(,DELETE)
     //SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(,DELETE)
     //SYSUT3   DD UNIT=SYSDA,SPACE=(CYL,(2,1)),DISP=(,DELETE)
     //SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,(2,1)),DISP=(,DELETE)
     //* -----  Assembler SYSLIB data set ---------------------
 5   //SYSLIB   DD DSN=data set name,DISP=SHR
                    •  •  •
     //* -----  Target libraries ------------------------------
     //LINKLIB  DD DSN=SYS1.LINKLIB,DISP=OLD
                    •  •  •
     //* -----  Distribution libraries -------------------------
     //AOSC5    DD DSN=SYS1.AOSC5,DISP=OLD
                    •  •  •
     //         PEND
```

Figure 33. Sample SMP/E cataloged procedure

1

SMPPUNCH is required for the GENERATE, REPORT, and UNLOAD commands. Because it might have a high level of output, SMPPUNCH should be directed to disk or tape.

2

SMPCSI DD statements should be coded with DISP=SHR. This allows SMP/E to share the CSI data sets as much as possible. If DISP=OLD is specified, no data set sharing is attempted. For more information about how SMP/E shares data sets, see the section on sharing SMP/E data sets in *z/OS SMP/E Commands*.

3

SMPWRK1–SMPWRK6 show only sample sizes for the data sets. The actual size required depends on the number of SYSMODs being processed and the number of elements within those SYSMODs.

4

SMPWRK3 can be permanently allocated in order to reuse assemblies. For more information, see the description of the REUSE operand in the APPLY command section in *z/OS SMP/E Commands*.

5

SYSLIB concatenation depends on how you intend to use the distribution libraries. For details on which data sets to include and in what order, see "How to determine the appropriate SYSLIB concatenation" on page 90.

If you use a different SYSLIB concatenation for APPLY and ACCEPT and prefer to use a SYSLIB DD statement, you should have at least two procedures. If you use DDDEFs to point to the different library

concatenations, you can use one procedure. You can modify the examples to use the appropriate procedure.

The following job uses the cataloged procedure in Figure 33 on page 89 to call SMP/E.

```
//SMPJOB   JOB 'accounting info',MSGLEVEL=(1,1)
//SMPSTEP  EXEC SMPPROC
//SMPPTFIN DD  … points to the file or data set that contains
//*               the SYSMODs to be received
//SMPHOLD  DD  … points to the file or data set that contains
//*               the HOLDDATA to be received
//SMPTLIB  DD  UNIT=3380,VOL=SER=TLIB01
//SMPCNTL  DD  *
  SET      BDY(GLOBAL)        /* Set to global zone       */.
  RECEIVE  SYSMOD             /* receive SYSMODS and      */
           HOLDDATA           /* HOLDDATA                 */
           SOURCEID(MYPTFS)   /* Assign a source ID       */
                              /*                          */.
  LIST     MCS                /* List the cover letters   */
           SOURCEID(MYPTFS)   /* for the SYSMODs          */
                              /*                          */.
  SET      BDY(TARGET1)       /* Set to target zone       */.
  APPLY    SOURCEID(MYPTFS)   /* Apply the SYSMODs        */
                              /*                          */.
  LIST     LOG                /* List the target zone log */.
/*
```

How to determine the appropriate SYSLIB concatenation

The recommended method for determining the appropriate SYSLIB concatenation treats the distribution libraries as totally separate from the target libraries.

Note: The example shown is for processing a zone for SREL Z038 containing the z/OS base control program (BCP). For other zones, follow the recommendations for the products residing in those zones.

Treating the distribution libraries as separate from the target libraries ensures that only the latest tested version of a macro is used during an assembly. Thus, the SYSLIB concatenation at ACCEPT is different from that at APPLY.

The SMPMTS data set contains macros from SYSMODs that are applied. Therefore, the proper SYSLIB concatenation for APPLY processing includes the SMPMTS data set, as is shown in Figure 34 on page 90.

```
//* ------ Include SMPMTS first
//* ------ followed by all macro target libraries
//* ------ followed by all distribution
//*        libraries
//SYSLIB   DD  DSN=SYS1.SMPMTS,DISP=OLD
//         DD  DSN=SYS1.MACLIB,DISP=OLD
//         DD  DSN=SYS1.MODGEN,DISP=OLD
//* ------ IF YOU NEED TO ASSEMBLE JES SOURCE MODULES:
//* ------ either HASPSRC (JES2) or JES3MAC (JES3)
//         DD  DSN=SYS1.HASPSRC,DISP=OLD
//         DD  DSN=SYS1.SISTMAC1,DISP=OLD
//         DD  DSN=SYS1.ATSOMAC,DISP=OLD
//         DD     •
//         DD     • other macro target libraries
//         DD     •
//         DD     •
//         DD     • any macro distribution libraries needed
//         DD     •
```

Figure 34. APPLY SYSLIB concatenation: APPLY different from ACCEPT

During ACCEPT processing, the macros in the SMPMTS and in the target macro libraries are not considered to have been tested. The SMPMTS is, therefore, not concatenated. Figure 35 on page 91 shows the proper SYSLIB concatenation for ACCEPT.

```
//* ------ Include only macro distribution
//*        libraries
//         DD  DSN=SYS1.AMACLIB,DISP=OLD
//SYSLIB   DD  DSN=SYS1.AMODGEN,DISP=OLD
//* ------ IF YOU NEED TO ASSEMBLE JES SOURCE MODULES:
//* ------ either HASPSRC (JES2) or JES3MAC (JES3)
//         DD  DSN=SYS1.HASPSRC,DISP=OLD
//         DD  DSN=SYS1.AISTMAC1,DISP=OLD
//         DD  DSN=SYS1.ATSOMAC,DISP=OLD
//         DD     •
//         DD     • other macro distribution libraries
//         DD     •
```

Figure 35. ACCEPT SYSLIB concatenation: APPLY different from ACCEPT

Checking that you have the appropriate access

If you are applying a SYSMOD that has an HFS element with one of the attributes listed in the following table, you must run the SMP/E job with a user ID that has at least the corresponding access privileges, and be UID 0. The attributes for an HFS element are specified on the PARM operand of the HFS element MCS.

BPXCOPY attribute	Access
APF	READ access to the BPX.FILEATTR.APF resource in the FACILITY class
PROGCTL	READ access to the BPX.FILEATTR.PROGCTL resource in the FACILITY class
SHARELIB	READ access to the BPX.FILEATTR.SHARELIB resource in the FACILITY class
UID(owner)	READ access to the BPX.SUPERUSER resource in the FACILITY class
GID(group)	READ access to the BPX.SUPERUSER resource in the FACILITY class.

If you are applying a SYSMOD that contains a MOD element that gets linked into a load module that resides in a UNIX file system directory, and that load module has any of the attributes listed in the following table, then you must run the SMP/E job with a user ID that has at least the corresponding access privileges, or be UID 0. The attributes for a load module are defined using binder options specified on the SETOPT binder control statement. The SETOPT control statement is specified in the JCLIN that defines the load module.

Binder attribute	Access
EXTATTR(APF)	Read access to the BPX.FILEATTR.APF resource in the FACILITY class
EXTATTR(SHRLIB)	Read access to the BPX.FILEATTR.SHARELIB resource in the FACILITY class
EXTATTR(PGM)	Read access to the BPX.FILEATTR.PROGCTL resource in the FACILITY class
UID(owner)	READ access to the BPX.SUPERUSER resource in the FACILITY class.
GID(group)	READ access to the BPX.SUPERUSER resource in the FACILITY class

If the user ID does not have the appropriate access, then the SMP/E operation will fail with either the BPXCOPY or binder link-edit failure.

For more information about the EXTATTR binder option, see EXTATTR: Specify extended attributes in *z/OS MVS Program Management: User's Guide and Reference*. For a description of the BPXCOPY options, see BPXCOPY in *z/OS UNIX System Services Command Reference*.

Defining exit routines

There are two types of exit routines you can define to tailor SMP/E processing:

- The **RECEIVE exit routine** enables you to scan statements in the SMPPTFIN data set during RECEIVE processing.
- The **retry exit routine** enables you to control retry processing when a data set runs out of space during RETRY can be specified on the ACCEPT, APPLY, LINK LMODS, LINK MODULE, and RESTORE commands. processing. (In retry processing, the data set is compressed and the utility that failed is called again.)

See *z/OS SMP/E Reference* for more informaton about SMP/E exit routines.

Chapter 4. Preparing to use Internet Service Retrieval

You can use Internet Service Retrieval to submit requests for PTFs and HOLDDATA to a remote IBM server and automatically download the packages that result when those requests are fulfilled. Use the RECEIVE ORDER command to submit an Internet Service Retrieval request to the IBM Automated Delivery Request server. SMP/E uses the hypertext transfer protocol and Secure Sockets Layer (HTTPS) to communicate with the server and HTTPS or FTP to download the packages. To support the HTTPS communication infrastructure, SMP/E uses the capabilities of Java and x.509 certificates to identify you to the server and perform SSL authentication. To use both Java and x.509 certificates, you need to perform some one-time configuration steps before you can use the SMP/E RECEIVE ORDER command.

The following overview explains how to use x.509 certificates to establish identity and authenticity for client/server communications. It also defines the steps that you need to take to accomplish the following tasks:

- Obtain a user certificate
- Set up the z/OS Security Server to work with certificates and install the user certificate
- Define the ORDERSERVER input for the RECEIVE ORDER command
- Define the CLIENT input for the RECEIVE ORDER command, including information necessary to allow SMP/E to use Java.

Identity and authentication overview

SMP/E communicates with the remote IBM Automated Delivery Request server using the HTTP protocol, and all HTTP communications with the server are performed using Secure Sockets Layer (SSL). Both the client (SMP/E) and the server use x.509 certificates to secure communications when using SSL. When initializing an SSL connection with a server, the client requests the server's x.509 certificate to authenticate the server. The server's certificate identifies the server to the client and provides the server's public key.

SSL server authentication allows a client application to confirm the identity of the server application. The client application through SSL uses standard public-key cryptography to verify that the server’s certificate and public key are valid and that the certificate has been signed by a trusted certificate authority (CA) that is known to the client application. The client and the server then use the negotiated session keys and begin encrypted communications.

One of the most important pieces of the SSL server authentication scheme is the trusted certificate authority (CA). Certificate Authorities are trusted organizations that verify information about servers and then issue digital certificates that may be accepted by applications as authentication of server identities when used in a secure handshaking protocol such as SSL. Trusting a certificate issued by a certificate authority is analogous to accepting a passport issued by a national passport agency as proof of identity. We trust that the agency has taken proper measures to verify the identity of the bearer of the passport. In a similar manner, applications may accept certificates signed by a certificate authority.

Two types of certificates are of interest to SMP/E processing:

User certificate
: A certificate that is associated with a z/OS user ID and is used to authenticate the user’s identity. Such a certificate may also be known as a Personal, or Client certificate.

Certificate-authority certificate
: A certificate that is associated with a certificate authority and is used to verify signatures in other certificates. Such a certificate may also be known as a root certificate. DigiCert is an example of a certificate authority that provides a certificate authority certificate.

© Copyright IBM Corp. 1986, 2026

Obtaining a user certificate

Before you can use the RECEIVE ORDER command, you must register to use IBM's server and obtain a certificate that identifies you to the server. You must use ShopzSeries to register, and the registration process generates a user certificate for you. Certificates have an expiration date and you will need to obtain a certificate once per year. A single certificate that is generated for you can be used for many PTF and HOLDDATA orders, and SMP/E will notify you when the certificate is about to expire and you need to obtain a new one. See IBM Shopz (www.ibm.com/software/shopzseries/ShopzSeries_public.wss).

1. If you are not yet a ShopzSeries user, register to become one.
2. If you are, or have become a registered ShopzSeries user, logon and create a new order.
3. Select z/OS Service and then "Service certificate" in the drop down list.
4. On the next screen, select the "Certificate type," either a certificate with a 1024-bit RSA key, or a certificate with a 2048-bit RSA key, and then enter an encryption pass phrase. ShopzSeries uses this pass phrase to encrypt the PKCS12 package that contains your certificate and its associated private key. Remember this pass phrase because you will need to specify it again later when decrypting the package.

   Note: A PKCS12 package contains both a certificate and the associated private key of the certificate. Because a private key is sensitive and considered secret, the package must be encrypted to protect it. To encrypt the package, a one-time encryption key must be used. That encryption key is also known as the pass phrase. You specify a pass phrase (any phrase you will remember) in ShopzSeries when you request a certificate to be generated for you.
5. Download to your workstation the generated PKCS12 certificate file as directed by ShopzSeries.

After downloading the certificate file to your workstation, you need to upload it to your z/OS system and add the certificate to your security product data base. SMP/E obtains the certificate from your security product data base and uses it during communications with IBM's Automated Delivery Request server.

Uploading the user certificate to z/OS

After you download the certificate file to your workstation, you must upload it to your z/OS system. There are many ways to transfer files from your workstation to your z/OS system. For example, you can upload the certificate file with Personal Communications 3270 or you can use TCP/IP FTP. The important things to remember are the certificate file must be uploaded to z/OS as binary data, the certificate file must be stored in a sequential data set, and the sequential data set must have RECFM=VB and LRECL=256 or greater.

Setting up RACF

Before you can begin making changes to your security product data base, you must ensure that your user ID is authorized to manipulate certificates and key rings. Consult your system's RACF administrator and refer to *z/OS Security Server RACF Security Administrator's Guide* before modifying the security characteristics for your system.

SMP/E requires either the RACF component of z/OS Security Server or an equivalent security product on z/OS to store and manage x.509 certificates. The documentation assumes that you are using RACF. If you are using an equivalent security product, you should refer to that product's documentation to understand the equivalent actions. For more information, see "Setting up alternate security products" on page 99.

Access to the RACDCERT command

First, you need to define the necessary FACILITY class profiles to give you access to use the RACDCERT commands. RACF's control levels in increasing strength are NONE, READ, UPDATE, CONTROL, and ALTER. To use the RACDCERT command, the command issuer requires appropriate permission to the IRR.DIGTCERT.function profile under the FACILITY class. In general, READ access is required to manipulate your own certificates and key rings, UPDATE access is required to manipulate them for other users, and CONTROL access is required to manipulate CERTAUTH (certificate authority) certificates.

Therefore, you can use the following sample RACF RDEFINE and PERMIT commands to define necessary FACILITY class profiles and to give you access to use the RACDCERT commands:

```
RDEFINE FACILITY IRR.DIGTCERT.ADD      UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.ADDRING  UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.ALTER    UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.CONNECT  UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LIST     UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)

PERMIT IRR.DIGTCERT.ADD      CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT IRR.DIGTCERT.ADDRING  CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT IRR.DIGTCERT.ALTER    CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT IRR.DIGTCERT.CONNECT  CLASS(FACILITY) ID(userid) ACCESS(UPDATE)
PERMIT IRR.DIGTCERT.LIST     CLASS(FACILITY) ID(userid) ACCESS(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

Note:

1. UPDATE access is required to the IRR.DIGTCERT.CONNECT profile in the FACILITY class in order to connect a certificate authority (CA) certificate to your key ring. See “Displaying certificate authority certificates” on page 95 for details of CA certificates.
2. To use the SMP/E RECEIVE ORDER command, access is required only to the IRR.DIGTCERT.LIST and IRR.DIGTCERT.LISTRING profiles. Access to the other profiles is required to create and manipulate key rings and certificates.

When your user ID has the proper authorization, continue with “Creating key rings” on page 95.

Creating key rings

A *key ring* is a named collection of certificates associated with a specific user. A certificate is identified by its label and the key ring that it is connected to. A key ring must be created using a RACF command like the following one:

```
RACDCERT ID(ring-owner) ADDRING(keyringname)
```

where *ring-owner* is the user ID that will own the key ring and *keyringname* is a name that you chose for the key ring.

Displaying certificate authority certificates

A certificate authority (CA) certificate is used to verify signatures in other certificates such as the server certificate. The IBM Automated Delivery Request server uses a server certificate issued by the DigiCert Global Root G2 certificate. Therefore, the DigiCert Global Root G2 certificate must be accessible in the RACF database during RECEIVE ORDER command processing so the server certificate can be verified. That is, if the DigiCert Global Root G2 certificate is not already in your RACF database, then you will need to add it.

To determine if the DigiCert Global Root G2 certificate is already in your RACF database, you must search for and display the certificate using the certificate's unique serial number and issuer. The serial number and issuer remain constant no matter what label is assigned to the certificate when it is added to your RACF database. Use the following RACF command to display the DigiCert Global Root G2 certificate:

```
RACDCERT CERTAUTH LIST( +
SERIALNUMBER(033AF1E6A711A9A0BB2864B11D09FAE5) +
ISSUERSDN( +
'CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US')
```

If the certificate is found in your RACF database, you should see the certificate information, like this:

```
Label: DigiCert Global Root G2
Certificate ID: 2QiJmZmDhZmjgcSJh4nDhZmjQMeTloKBk0DZlpajQMfy
Status: TRUST
Start Date: 2013/08/01 08:00:00
End Date:   2038/01/15 08:00:00
Serial Number:
     >033AF1E6A711A9A0BB2864B11D09FAE5<
```

```
Issuer's Name:
     >CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US<
Subject's Name:
     >CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US<
Signing Algorithm: sha256RSA
Key Usage: HANDSHAKE, CERTSIGN
Key Type: RSA
Key Size: 2048
Private Key: NO
Certificate Fingerprint (SHA256):
     CB:3C:CB:B7:60:31:E5:E0:13:8F:8D:D3:9A:23:F9:DE:
     47:FF:C3:5E:43:C1:14:4C:EA:27:D4:6A:5A:B1:CB:5F
```

If the certificate is found, make note of the label for the certificate, as the label for the certificate in your RACF database can be different than “DigiCert Global Root G2”. You must use the actual label value in the subsequent RACF commands. If the certificate is found but does not have a status of TRUST, then you must use the following RACF command to trust the DigiCert Global Root G2 certificate:

```
RACDCERT CERTAUTH +
ALTER(LABEL(’DigiCert Global Root G2’)) TRUST
```

If the certificate is not found, you will need to add the DigiCert Global Root G2 certificate to your RACF database.

Adding certificate authority certificates

To add the DigiCert Global Root G2 certificate to your RACF database, perform the following steps:

1. Download to your work station the DigiCert Global Root G2 certificate file. Using your browser, go to the DigiCert Trusted Root Authority Certificates web page DigiCert Trusted Root Authority Certificates (www.digicert.com/digicert-root-certificates.htm). Find the DigiCert Global Root G2 certificate in the list of root certificates and click the "Download PEM" link for this certificate to download the certificate file to your workstation.
2. Upload the CA certificate to your z/OS system. There are many methods to transfer files from your workstation to your z/OS system. For example, you can upload the certificate file with Personal Communications 3270 or use TCP/IP FTP. Since the PEM format certificate file is text data you can also open the file in a text editor and use your workstation’s cut/paste feature. The PEM format certificate file must be uploaded to z/OS as text data, the certificate file must be stored in a sequential data set, and the sequential data set must have RECFM=VB and LRECL>=256.
3. After you store the certificate in a sequential data set, add it to your RACF database by using the following RACF command:

   ```
   RACDCERT CERTAUTH ADD('ca-cert.dataset.name') +
   WITHLABEL('DigiCert Global Root G2') TRUST
   ```

where *ca-cert.dataset.name* is the name of the sequential data set used to store the certificate received from the DigiCert website.

Adding the user certificate to your RACF data base

A user certificate is used by the SMP/E RECEIVE ORDER command to uniquely identify you to the IBM Automated Delivery Request server. As described previously, the user certificate was generated for you by ShopzSeries, downloaded to your workstation, transferred to your z/OS system as binary data, and stored as a sequential data set. From the sequential data set, the certificate can be stored in the RACF data base using the following RACF command:

```
RACDCERT ID(certificate-owner) ADD('user.certificate.dataset.name') +
WITHLABEL('SMPE Client Certificate') PASSWORD('pass phrase') TRUST
```

where *certificate-owner* is the user ID that you choose to own the certificate, *user.certificate.dataset.name* is the data set name used to store the PKCS12 certificate package obtained from ShopzSeries, *SMPE Client Certificate* is the label you choose to identify this certificate (32 characters or less), and *pass*

phrase is the encryption pass phrase you specify when generating the PKCS12 certificate package on ShopzSeries.

Note: After you issue the preceding RACDCERT command, RACF should return this message: `"certificate authority not defined to RACF. Certificate added with TRUST status."` This is the expected response and is acceptable.

Connecting the certificates to the key ring

After you add the user certificate to the RACF database and have verified the DigiCert Global Root G2 certificate is in your RACF database, you must connect both certificates to the key ring. Use the following RACF commands:

```
RACDCERT ID(ring-owner) CONNECT(LABEL('SMPE Client Certificate') +
RING(keyringname) USAGE(CERTAUTH) )

RACDCERT ID(ring-owner) CONNECT( CERTAUTH LABEL('DigiCert Global Root G2') +
RING(keyringname) USAGE(CERTAUTH) )
```

where *SMPE Client Certificate* is the label you choose in the previous step to identify this certificate, *keyringname* is the name of the key ring you choose in "Creating key rings" on page 95, and *ring-owner* is the user ID that created the key ring.

Note: To enable the user certificate to be easily shared by other user IDs without requiring unnecessarily high levels of access for those other user IDs, the user certificate must be connected to the key ring as a certificate authority (CA) certificate (USAGE CERTAUTH). Connecting a user certificate to a key ring with USAGE CERTAUTH is not typical for SSL/TLS client authentication. However, the user certificate in this case is not used for this purpose. The user certificate in this case is merely a secure container for customer identity information to be presented by SMP/E to the IBM Automated Delivery Request server. Connecting the user certificate to the key ring with USAGE CERTAUTH enables SMP/E when running under a user ID that is not the certificate or key ring owner to access the certificate, as long as the user ID has read access to key rings.

Sharing a user certificate among multiple user IDs

It is possible for multiple users to share a single user certificate obtained from ShopzSeries. To do so, you must first create a key ring, enable the CA certificate, and add the user certificate to your RACF data base as explained in the preceding topics. Assume that user ID USER1 is associated with the key ring and is the owner of the user certificate. In order to allow user ID USER2 to share the user certificate, you must give USER2 permission to read other users' key rings and certificates. You can use the following RACF commands:

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(USER2) ACCESS(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(USER2) ACCESS(UPDATE)
```

Permitting USER2 UPDATE access to the IRR.DIGTCERT.LISTRING FACILITY class is not a security exposure. It is true that USER2 will have the ability to read anyone's key ring. However, that only allows the ability to extract and use the certificates from the key ring. It does not allow use of the private keys associated with those certificates. Therefore, USER2 cannot masquerade as another user ID.

After USER2 has the appropriate permission, in order for USER2 to use the certificate for the SMP/E RECEIVE ORDER command, you must ensure SMP/E finds the certificate in the correct key ring when running the command. To do this, USER2 must specify not only the key ring name, but also the user ID associated with the key ring, USER1, on the *keyring* attribute in the ORDERSERVER data set for the RECEIVE ORDER command as follows:

```
keyring="USER1/keyringname"
```

See "Defining the ORDERSERVER input for RECEIVE ORDER" on page 99 for further information about the *keyring* attribute and the ORDERSERVER data set.

Debugging key ring and certificate issues

The following RACDCERT commands might be useful if the SMP/E RECEIVE ORDER command detects errors or failures related to your key ring or certificates. You can use the following RACDCERT commands to list the keying and certificates to verify their existence and proper attributes.

- To list the key ring:

  ```
  RACDCERT ID(ring-owner) LISTRING(keyringname)
  ```

  where *ring-owner* is the user ID that created and owns the key ring.
- To list the certificate authority (CA) certificate:

  ```
  RACDCERT CERTAUTH LIST(LABEL('DigiCert Global Root G2'))
  ```

  or

  ```
  RACDCERT CERTAUTH LIST( +
  SERIALNUMBER(033AF1E6A711A9A0BB2864B11D09FAE5) +
  ISSUERSDN( +
  'CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US'))
  ```

- To list the user certificate:

  ```
  RACDCERT ID(certificate-owner) LIST(LABEL('SMPE Client Certificate'))
  ```

  where *certificate-owner* is the user ID that owns the certificate.

Replacing a user certificate that expired

A user certificate obtained from ShopzSeries has a finite life span; it will expire after a specific time period and will no longer be valid. The SMP/E RECEIVE ORDER command warns you if the user certificate will expire within 30 days, and the command will fail if the certificate has finally expired. When this occurs, you should replace your existing certificate with a new one. The steps to replace an existing certificate with a new certificate are very much like those you performed when obtaining and adding the first user certificate.

1. Obtain a new user certificate from ShopzSeries as described in “Obtaining a user certificate” on page 94.
2. Upload the new user certificate to z/OS as described in “Uploading the user certificate to z/OS” on page 94.
3. Before adding the new certificate to your security product database, you must delete the existing certificate so that you can use the same label for the new user certificate. Use the following RACF command:

   ```
   RACDCERT ID(certificate-owner) DELETE(LABEL('SMPE Client Certificate'))
   ```

   where *certificate-owner* is the user ID that owns the certificate.
4. Follow the steps described in “Adding the user certificate to your RACF data base” on page 96 to add the new user certificate and to connect it to your key ring.

Because the new certificate uses the same label as the existing certificate, no other changes are necessary to ensure that your RECEIVE ORDER command jobs continue to run as expected.

Refreshing RACF classes

After you perform the RACF updates to add certificates and key rings, you might need to refresh the in-storage RACF profiles. That is, if you choose to RACLIST the DIGTCERT and DIGTRING classes, you must first activate them using the following RACF command:

```
SETROPTS CLASSACT(DIGTCERT DIGTRING)
```

If you have RACLISTed the DIGTCERT or DIGTRING classes, you need to refresh the in-storage profiles before the updates can take affect. You can use the following RACF command:

```
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
```

If you have not RACLISTed the DIGTCERT or DIGTRING classes, you do not need to refresh the in-storage profiles.

Setting up alternate security products

SMP/E requires either the z/OS Security Server (RACF) or an equivalent security product on z/OS to store and manage x.509 certificates. This chapter assumes that you are using RACF. If you are using an equivalent security product, you should refer to that product's documentation to understand the equivalent actions.

eTrust® CA-ACF2 security for z/OS users

Computer Associates™ has created a Hyper Notification, QI73845, to address the setup requirements associated with SMP/E Internet Service Retrieval for its eTrust CA-ACF2 Security for z/OS customers. Its purpose is to provide detailed instructions equivalent to the RACF instructions found herein. Refer to this Hyper Notification for details.

eTrust CA-Top Secret security for z/OS users

Computer Associates has created a Technical Document, ID# TEC394405, to address the setup requirements associated with SMP/E Internet service retrieval for its eTrust CA-Top Secret Security for z/OS customers. Its purpose is to provide detailed instructions equivalent to the RACF instructions found herein. Refer to this Technical Document for details.

Defining the ORDERSERVER input for RECEIVE ORDER

The ORDERSERVER data set is used by the SMP/E RECEIVE ORDER command to provide necessary information about the IBM Automated Delivery Request server. The information is described using the <ORDERSERVER> tag and attributes that are defined in detail in *z/OS SMP/E Commands*. Here is an example and brief explanations:

```
<ORDERSERVER
 url="https://eccgw01.boulder.ibm.com/services/projects/ecc/ws"
 keyring="SMPEKeyring"
 certificate="SMPE Client Certificate">
 </ORDERSERVER>
```

url
: Specifies the Uniform Resource Locator (URL) for the order server. The actual URLs for the IBM Automated Delivery Request server are `https://eccgw01.boulder.ibm.com/services/projects/ecc/ws` and `https://eccgw02.rochester.ibm.com/services/projects/ecc/ws`

keyring
: Identifies the RACF key ring that contains the user certificate required for access to the order server. If the key ring is owned by a user ID other than that used to run SMP/E, then the key ring name must be prefixed with the associated user ID. The user ID and key ring name must be separated by a forward slash. For example,`keyring="USER1/SMPEKeyring"`.

certificate
: Specifies the label to identify the user certificate used for access to the order server. This is the certificate obtained from ShopzSeries and added to your security product data base.

Defining the CLIENT input for RECEIVE ORDER

The CLIENT data set is used by the SMP/E RECEIVE ORDER command to provide information about the local z/OS system and network, as well as certain processing options. The information is described using the <CLIENT> tag and attributes that are defined in detail in *z/OS SMP/E Commands*. Here is an example:

```
<CLIENT
  javahome="/usr/lpp/java/J8.0"
  classpath="/usr/lpp/smp/classes"
  javadebugoptions="-Dcom.ibm.smp.debug=severe -showversion"
  downloadmethod="ftp">
  <HTTPPROXY host="local.httpproxy.com">
  </HTTPPROXY>
  <FTPOPTIONS>-v -f "//'USER1.FTP.DATA'"</FTPOPTIONS>
  </FIREWALL>
    <SERVER host="local.ftpproxy.com"> </SERVER>
    <FIRECMD>&REMOTE_USER;@&REMOTE_HOST;</FIRECMD>
    <FIRECMD>&REMOTE_PW;</FIRECMD>
  </FIREWALL>
</CLIENT>
```

Not all tags and attributes shown in the example are required for every user, but these are explained later in this section. The options of the CLIENT data set can be grouped into three categories: options that affect Java, options that affect HTTPS, and options that affect downloads.

Options that affect Java

To support the HTTPS communications with the IBM Automated Delivery Request server, SMP/E uses the capabilities of Java. Specifically, SMP/E requires IBM 31-bit SDK for z/OS, Java Technology Edition Version 8 (5655-DGG), or the IBM 64-bit SDK for z/OS, Java Technology Edition Version 8 (5655-DGH), or a logical successor. SMP/E supplies several Java application classes, which after SMP/E is installed, reside in the `/usr/lpp/smp/classes` directory in the UNIX file system on the z/OS system. In order for SMP/E to use these application classes, they and the Java runtime must be available in the execution environment for SMP/E. Because neither can be accessed using a STEPLIB DD statement, you specify the locations for the Java runtime and the SMP/E application classes using the `javahome` and `classpath` attributes in the CLIENT data set. The `javahome` attribute is used to specify the directory where the Java run time resides, and the `classpath` attribute is used to specify the search path for Java application classes. For example:

```
 javahome="/usr/lpp/java/J8.0"
 classpath="/usr/lpp/smp/classes"
```

If the SMP/E Java application classes are not installed on the driving z/OS system's UNIX file system, you can use appropriate mount points and directory structure to point to a target system's directories where SMP/E is installed. This is useful if you use STEPLIB to access a target system's base SMP/E programs in order to ensure the base SMP/E programs and the application classes are at the same service level. For example:

```
classpath="/TGTSYS/usr/lpp/smp/classes"
```

As an alternative to the `javahome` and `classpath` attributes in the CLIENT data set, you can also use the SMPJHOME and SMPCPATH DD statements or DDDEF entries to specify the location of the Java run time and the SMP/E application classes. If specified, the SMPJHOME and SMPCPATH DD statements override any values specified on the `javahome` and `classpath` attributes. For example:

```
//SMPJHOME DD PATH='/usr/lpp/java/J8.0'
//SMPCPATH DD PATH='/usr/lpp/smp/classes'
```

The *javadebugoptions* attribute is used to specify any Java command-line parameters, including debug and trace options. Such options determine what debug and trace information should be produced when SMP/E invokes its Java application classes. The debug and trace information is written to the PRINT file for the HFSCOPY utility (SYSPRINT is the default). Although it is not necessary all of the time, it is a good

idea when first using the RECEIVE ORDER command, to specify the following value to ensure that basic debug and trace information is produced for reference:

```
javadebugoptions="-Dcom.ibm.smp.debug=severe -showversion"
```

Although it is not necessary, it is also possible to specify other Java command-line parameters that affect the operation of the Java Virtual Machine (JVM). For example, if you encounter a Java error that indicates there is insufficient space in the Javaheap, you can specify an option to override the default maximum Java heap size:

```
javadebugoptions="-Xmx128m"
```

Options that affect HTTPS operations

SMP/E communicates with the IBM Automated Delivery Request server using the HTTP 1.1 protocol using Secure Sockets Layer (SSL), also known as HTTPS. All communications with the server are performed using the HTTPS port of 443.

Optional tags are available in the CLIENT data set for the RECEIVE ORDER command to describe local HTTP or SOCKS proxy servers. A proxy server redirects HTTP requests to the IBM Automated Delivery Request server on the Internet. The <HTTPPROXY> tag is used to identify a local HTTP proxy server, and the <HTTPSOCKSPROXY> tag is used to identify a local SOCKS proxy server. For example:

```
<HTTPPROXY host="local.httpproxy.com"> </HTTPPROXY>
```

The <HTTPPROXY> and <HTTPSOCKSPROXY> tags are optional, and should be specified only if the HTTP requests to the Internet from your z/OS system are required to pass through a specific HTTP or SOCKS proxy server. For example, if you must specify a proxy server in your Internet browser configuration to allow you access to websites on the Internet, then you might need to specify the <HTTPPROXY> or <HTTPSOCKSPROXY> tag in the CLIENT data set. If your HTTP or SOCKS proxy server requires authentication, the *user* and *pw* attributes can be used to specify the proxy server user ID and password. Also, if your HTTP or SOCKS proxy server listens on a port other than the ports 80 and 1080, the *port* attribute can be used to specify an alternate port value. For example:

```
<HTTPPROXY host="local.httpproxy.com"
           user="userid" pw="password" port="8080"> </HTTPPROXY>
```

See *z/OS SMP/E Commands* for complete details of the <HTTPPROXY> and <HTTPSOCKSPROXY> tags and attributes, and **consult your network administrator** for help determining what, if anything, you must specify for an HTTP or SOCKS proxy server.

Options that affect download operations

SMP/E uses HTTPS to communicate with the IBM Automated Delivery Request server, and has the capability to use FTP, FTPS, HTTP, or HTTPS to download package files containing PTFs and HOLDDATA, from a download server to your local z/OS® system. After the package files are staged to a download server, the Automated Delivery Request server provides SMP/E with the information it needs to authenticate with the download server and then download the package files. Specifically, it provides SMP/E with the server host name and a user ID and password for that server, which are unique for the specific package to be downloaded.

The `downloadmethod` attribute in the CLIENT data set tells SMP/E which method to use for downloading the package files from the server to your local z/OS. IBM's download server requires the use of a secure and encrypted download method, either FTPS or HTTPS, to download package files to your local z/OS system. See Chapter 5, "Preparing for secure Internet delivery," on page 105 for information on setting up to use FTPS or HTTPS and instructing SMP/E to use one of those methods for download operations.

Network configuration notes

SMP/E assumes that you have network connectivity from your z/OS system to the IBM servers through the Internet. Consult your network administrator and the *z/OS Communications Server: IP Configuration Guide* for information about how to set up your z/OS system's network configuration properly.

The HTTP(S) and FTP operations performed by SMP/E require host name to IP address resolution. This is usually accomplished using a Domain Name System (DNS) name server. A name server is defined using the NSINTERADDR or NAMESERVER statement within a resolver configuration file (TCPIP.DATA information). There are several different locations where a resolver configuration file can be found when using an application such as SMP/E. Because SMP/E uses a UNIX process for its HTTP(S) and FTP operations, the z/OS UNIX search order is used to find the resolver configuration file. See the sections titled "Resolver configuration files" and "Search orders used in the z/OS UNIX environment" in *z/OS Communications Server: IP Configuration Guide* for details on specifying the location of the resolver configuration file. However, because of the asynchronous UNIX process used by SMP/E for its HTTP(S) and FTP operations, there are two exceptions to the documented search order: neither the SYSTCPD DD statement, nor the RESOLVER_CONFIG environment variable can be used to define the location of the resolver configuration file.

You can verify your name server setup by using the following sample job to invoke the NSLOOKUP command:

```
//jobname  JOB ...
//NSLOOKUP EXEC PGM=BPXBATCH,
//         PARM='PGM /bin/nslookup eccgw01.boulder.ibm.com'
//STDOUT   DD PATH='/tmp/&SYSUID..bpxbatch.stdout',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),PATHMODE=SIRWXU
//OUTSTEP  EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//INPUT    DD PATH='/tmp/&SYSUID..bpxbatch.stdout',
//         PATHOPTS=(ORDONLY),PATHDISP=DELETE
//OUTPUT   DD SYSOUT=*,DCB=(RECFM=V,LRECL=256)
//SYSTSIN  DD *
  OCOPY INDD(INPUT) OUTDD(OUTPUT)
/*
```

Note: Although the NSLOOKUP command can be run from the OMVS shell, this sample job runs the command in an environment similar to that in which SMP/E runs.

If your name server is set up properly, the nslookup command returns the IP address for the server. The output of the command is similar to the following output:

```
Defaulting to nslookup version 4
Starting nslookup version 4
Server:  local.dns.com
Address:  9.0.2.1

  Non-authoritative answer:
Name:    eccgw01.boulder.ibm.com
Address:  207.25.252.197
```

If an IP address for the eccgw01.boulder.ibm.com server is not returned, then verify that your name server setup and resolver configuration file are proper. See the section titled "Diagnosing resolver problems" in *z/OS Communications Server: IP Diagnosis Guide* for details about using the Trace Resolver debug facility.

Summary

The steps described in this topic help you to configure your system to enable the SMP/E RECEIVE ORDER command. After you perform the described tasks, you can run the RECEIVE ORDER command to order and download PTFs and HOLDDATA from the IBM Automated Service Delivery server. Here is a summary of the steps:

1. Obtain a user certificate:

a. Go to ShopzSeries to generate a user certificate and download the certificate file to your workstation.
b. Transfer the certificate file to your z/OS system as binary data and store as a sequential data set with RECFM=VB and LRECL=256 or greater.

2. Set up z/OS security server (RACF):
   a. Ensure that your user ID has appropriate RACF access to create key rings and add certificates.
   b. Create a RACF key ring.
   c. Add the DigiCert Global Root G2 certificate to your RACF data base if it does not already exist.
   d. Connect the DigiCert Global Root G2 certificate to your key ring.
   e. Add the user certificate to the RACF data base.
   f. Connect the user certificate to your key ring.
3. Setup for using Java:
   a. Ensure Java Version 8, or a logical successor is installed on your z/OS system.
   b. Specify the Java runtime directory on the javahome attribute in the CLIENT data set for the RECEIVE ORDER command.
   c. If necessary, specify the SMP/E Java application classes directory on the classpath attribute in the CLIENT data set for the RECEIVE ORDER command.
4. Setup for network configuration:
   a. If necessary, identify your HTTP proxy server in the CLIENT data set for the RECEIVE ORDER command.
   b. If necessary, identify your FTP firewall server and navigation commands in the CLIENT data set for the RECEIVE ORDER command.

Example

After you complete the necessary steps to configure your system to enable the SMP/E RECEIVE ORDER command, you can run an SMP/E RECEIVE ORDER job like the following:

```
//jobname JOB ...,REGION=0M
//RECEIVE EXEC PGM=GIMSMP
//SMPCSI DD DSN=smpe.global.csi,DISP=SHR
//SMPNTS DD PATH='/u/smpe/smpnts/',PATHDISP=KEEP
//SMPOUT DD SYSOUT=*
//SMPRPT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
  SET BDY(GLOBAL).
  RECEIVE SYSMODS HOLDDATA
          ORDER(ORDERSERVER(ORDSRVR)
          CLIENT(MYCLIENT)
          CONTENT(RECOMMENDED))
          DELETEPKG.
/*
//ORDSRVR DD *
  <ORDERSERVER
    url="https://eccgw01.boulder.ibm.com/services/projects/ecc/ws/"
    keyring="keyringname"
    certificate="SMPE Client Certificate">
   </ORDERSERVER>
/*
//MYCLIENT DD *
   <CLIENT
    javahome="/usr/lpp/java/J8.0"
    javadebugoptions="-Dcom.ibm.smp.debug=severe -showversion"
    downloadmethod="ftp">
    <HTTPPROXY host="local.httpproxy.com">
    <FTPOPTIONS>-v -f "//'USER1.FTP.DATA'"</FTPOPTIONS>
    </HTTPPROXY>
    <FIREWALL>
      <SERVER host="local.ftpproxy.com"> </SERVER>
      <FIRECMD>&REMOTE_USER;@&REMOTE_HOST;</FIRECMD>
      <FIRECMD>&REMOTE_PW;</FIRECMD>
```

```
        <FIREWALL>
      </CLIENT>
    /*
```

Note:

- An alternative URL for the IBM Automated Delivery Request server is `https://eccgw02.rochester.ibm.com/services/projects/ecc/ws`.
- For the `javahome` attribute, specify the directory for a currently supported Java Runtime Environment (JRE).

This sample job instructs SMP/E to order and then download all recommended (RSU) PTFs that are applicable to the target zones defined in your global zone. For information about the RECEIVE ORDER command, see The RECEIVE command in *z/OS SMP/E Commands*.

Chapter 5. Preparing for secure Internet delivery

z/OS product and service offerings can be downloaded directly from IBM's servers to your z/OS system. SMP/E provides capabilities to perform these download operations using the RECEIVE command and the GIMGTPKG service routine. SMP/E supports secure and encrypted download operations using FTPS (FTP over SSL/TLS) and HTTPS (HTTP over SSL). However, using either of these download methods requires preparation and one-time setup.

This topic provides an overview of using SMP/E for secure internet download operations, in particular from IBM's secure delivery servers, and the one-time steps you need to take to prepare.

- SSL overview
- Enable certificate authority certificates
- Define CLIENT input for RECEIVE and GIMGTPKG

HTTPS Fast Path!

The quick and easy method to enable secure download operations is to instruct the SMP/E RECEIVE command and GIMGTPKG service routine to use the HTTPS download method and certificate authority (CA) certificates managed by the default z/OS Java truststore. To do so, simply specify the SMP/E <CLIENT> tag with the following attributes:

```
<CLIENT
  downloadmethod="https"
  downloadkeyring="javatruststore
  javahome="/usr/lpp/java/J8.0"
  >
</CLIENT>
```

If you want to understand the background and details of the above attributes, or if you want to explore other options such as FTPS or using CA certificates stored in your z/OS security manager database, then read on. Otherwise, you can skip the rest of this topic.

Secure Sockets Layer overview

FTPS and HTTPS are internet protocols that use Secure Sockets Layer (SSL) technology to perform secure and encrypted communications between client and server applications. When initializing an SSL connection with a server, the client requests the server's x.509 certificate to authenticate the server. The server's certificate identifies the server to the client and provides the server's public key. SSL server authentication allows a client application to confirm the identity of the server application. The client application through SSL uses standard public key cryptography to verify that the server’s certificate and public key are valid and that the certificate has been signed by a trusted certificate authority (CA) that is known to the client application. The client and the server then use negotiated session keys to begin encrypted communications.

One of the most important pieces of the SSL server authentication scheme is the trusted certificate authority (CA). Certificate authorities are trusted organizations that verify information about servers and then issue digital certificates that can be accepted by applications as authentication of server identities when used in a secure handshaking protocol such as SSL. Trusting a certificate issued by a certificate authority is analogous to accepting a passport issued by a national passport agency as proof of identity. We trust that the agency has taken proper measures to verify the identity of the bearer of the passport. In a similar manner, applications may accept certificates signed by a certificate authority.

A certificate authority certificate is associated with a certificate authority and is used to verify signatures in other certificates. Such a certificate may also be known as a root certificate. DigiCert is an example of a

© Copyright IBM Corp. 1986, 2026

certificate authority that provides certificate authority certificates. As of August 14, 2018, the IBM secure delivery servers use a server certificate signed by the DigiCert certificate authority.

Enabling certificate authority certificates

Certificate authority certificates are often managed and stored in your security manager database on z/OS. However, to display, add, and use certificate authority certificates stored in your security manager database requires proper authorization. If you do not already have proper authorization, a simpler approach is to choose the HTTPS download protocol and allow SMP/E to use the default truststore provided by the Java runtime on z/OS. The default Java truststore contains a collection of trusted certificate authority certificates, including the DigiCert Global Root G2 certificate required by IBM's secure delivery servers. This truststore can be used by SMP/E for HTTPS download operations to verify the server certificate during the SSL handshake.

Therefore, do one of the following:

1. If you choose to use HTTPS as your download protocol, and if you choose to use certificate authority certificates managed by the default Java truststore, then skip the following topic that describes how to update your security manager database. Instead, go directly to the topic "Define CLIENT input for RECEIVE and GIMGTPKG" on page 108.
2. If you choose to use your security manager to control certificate authority certificates, or if you choose to use FTPS as your download protocol, then the DigiCert Global Root G2 certificate must be in your security manager database so SMP/E can verify the server certificate during the SSL handshake. That is, if the DigiCert Global Root G2 certificate is not already in your security manager database, you will need to add it. The actions you must take to add the certificate are further described in this topic.

Note: The z/OS® Security Server (RACF) or an equivalent security manager product on z/OS are used to store and manage x.509 certificates. The remainder of this topic assumes you are using RACF. If you are using an equivalent security manager product, you should refer to that product's documentation to understand the equivalent actions.

Access to certificate authority certificates

To determine if the required CA certificate is already in your RACF database and to add the certificate if necessary, requires proper authorization. To display a CERTAUTH (certificate authority) certificate, you must have CONTROL access to FACILITY class resource IRR.DIGTCERT.LIST. To add a CERTAUTH certificate, you must have CONTROL access to FACILITY class resource IRR.DIGTCERT.ADD. Use the following RACF commands to assign proper authorization:

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
PERMIT IRR.DIGTCERT.ADD  CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
```

Displaying certificate authority certificates

Once you have access to display CERTAUTH certificates, you must search for and display the DigiCert Global Root G2 certificate to determine if it is already in your RACF database. Search for and display the certificate using the certificate's unique serial number and issuer. The serial number and issuer remain constant no matter what label is assigned to the certificate when it is added to your RACF database. Use the following RACF command to display the DigiCert Global Root G2 certificate:

```
RACDCERT CERTAUTH LIST( +
SERIALNUMBER(033AF1E6A711A9A0BB2864B11D09FAE5) +
ISSUERSDN( +
'CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US')
```

If the certificate is found in your RACF database, you should see the certificate information, like this:

```
Label: DigiCert Global Root G2
Certificate ID: 2QiJmZmDhZmjgcSJh4nDhZmjQMeTloKBk0DZlpajQMfy
Status: TRUST
Start Date: 2013/08/01 08:00:00
```

```
End Date:   2038/01/15 08:00:00
Serial Number:
     >033AF1E6A711A9A0BB2864B11D09FAE5<
Issuer's Name:
     >CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US<
Subject's Name:
     >CN=DigiCert Global Root G2.OU=www.digicert.com.O=DigiCert Inc.C=US<
Signing Algorithm: sha256RSA
Key Usage: HANDSHAKE, CERTSIGN
Key Type: RSA
Key Size: 2048
Private Key: NO
Certificate Fingerprint (SHA256):
     CB:3C:CB:B7:60:31:E5:E0:13:8F:8D:D3:9A:23:F9:DE:
     47:FF:C3:5E:43:C1:14:4C:EA:27:D4:6A:5A:B1:CB:5F
```

Make note of the label for the found certificate, as the label for the certificate in your RACF database can be different than “DigiCert Global Root G2”. You must use the actual label value in the subsequent RACF commands. If the certificate is found but does not have a Status of TRUST, then you must use the following RACF command to trust the DigiCert Global Root G2 certificate:

```
RACDCERT CERTAUTH +
ALTER(LABEL(’DigiCert Global Root G2’)) TRUST
```

If the certificate is not found, you will need to add the DigiCert Global Root G2 certificate to your RACF database.

Adding certificate authority certificates

To add the DigiCert Global Root G2 certificate to your RACF database, complete the following steps:

1. Download the DigiCert Global Root G2 certificate file to your work station. Using your browser, go to the DigiCert Trusted Root Authority Certificates (www.digicert.com/digicert-root-certificates.htm) web page DigiCert Trusted Root Authority Certificates (www.digicert.com/digicert-root-certificates.htm) and find the DigiCert Global Root G2 certificate in the list of root certificates. Click the "Download PEM" link for this certificate to download the certificate to your workstation.
2. Upload the certificate to your z/OS system. Transferring files from your workstation to your z/OS system can be done in multiple ways. For example, you can upload the certificate file with Personal Communications 3270 or use TCP/IP FTP. Because the PEM format certificate file is text data you can also open the file in a text editor and use your workstation’s cut and paste feature. The PEM format certificate file must be uploaded to z/OS as text data and the certificate file must be stored in a sequential data set. The sequential data set must have RECFM=VB and LRECL>=256.
3. After you store the certificate in a sequential data set, add it to your RACF database.

   ```
   RACDCERT CERTAUTH ADD('ca-cert.dataset.name') +
   WITHLABEL('DigiCert Global Root G2') TRUST
   ```

where *ca-cert.dataset.name* is the name of the sequential data set used to store the certificate received from the DigiCert website.

Access for SMP/E to use certificate authority certificates

Now that the appropriate CA certificate is in your RACF database, you must ensure SMP/E can access the certificate and a key ring, real or virtual, that contains the certificate. The user ID under which SMP/E runs must be properly authorized to the FACILITY class resource IRR.DIGTCERT.LISTRING or to the RDATALIB class resource *keyring-owner.keyring-name*.LST for SMP/E to use a specified key ring. READ access is required for a user ID to use its own key ring or the CERTAUTH virtual key ring. UPDATE access is required to use a key ring from another user. Use the following RACF command to ensure the user ID has access to read the CERTAUTH virtual key ring:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

If the RDATALIB class is used, for any real keyring, READ access to *keyring-owner.keyring-name*.LST is required. For CERTAUTH virtual keyring, READ access to CERTIFAUTH.IRR_VIRTUAL_KEYRING.LST is required.

Refreshing RACF classes

After you perform the RACF updates to add certificates, you might need to refresh the in-storage RACF profiles. That is, if you choose to RACLIST the DIGTCERT and DIGTRING classes, you must first activate them using the following RACF command:

```
SETROPTS CLASSACT(DIGTCERT DIGTRING)
```

If you have used RACLIST for the DIGTCERT or DIGTRING classes, you need to refresh the in-storage profiles before the updates can take affect. You can use the following RACF command:

```
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
```

If you have not used RACLIST for the DIGTCERT or DIGTRING classes, you do not need to refresh the in-storage profiles.

Define CLIENT input for RECEIVE and GIMGTPKG

The CLIENT data set is used by the SMP/E RECEIVE command and the GIMGTPKG service routine to provide SMP/E with information about the local z/OS system and network, as well as certain processing preferences. In particular, you can identify which method should be used for download operations, FTP(S) or HTTPS.

IBM recommends you use HTTPS for secure downloads because HTTPS is simple to configure and more firewall friendly than FTPS. However, if you can satisfy the configuration requirements for using FTPS and your firewall supports FTPS, then you can use FTPS instead of HTTPS.

The information in the CLIENT data set is described using the <CLIENT> tag and attributes that are defined in detail in *z/OS SMP/E Commands*.

Options that affect HTTPS operations

The following example <CLIENT> tag tells SMP/E that files should be downloaded using the HTTPS protocol, thus enabling secure and encrypted transfers.

```
<CLIENT
  downloadmethod=”https”
  downloadkeyring=”javatruststore”
  javahome="/usr/lpp/java/J8.0"
  >
  <HTTPPROXY host=”local.httpproxy.com”>
  </HTTPPROXY>
</CLIENT>
```

downloadmethod
: Identifies the network protocol to use for downloading files from the remote server to the local z/OS. The IBM secure delivery server supports values of either `https` or `ftp`. Specify a value of `https` to indicate files will be downloaded using HTTPS.

downloadkeyring
: Identifies the location of the certificate authority certificates required for the SSL handshake with the HTTPS server. The name for a security manager key ring, or the keyword `javatruststore` may be specified.

  If you choose to manage certificate authority certificates with your z/OS security manager, you can specify a real or virtual key ring. The simplest key ring specification is to use the CERTAUTH virtual key ring, *AUTH*/*. Using the CERTAUTH virtual key ring indicates all of the trusted CA certificates that are defined in the security manager database are available for use during the SSL handshake.

This includes the required DigiCert Global Root G2 certificate that is described in the topic "Enabling certificate authority certificates" on page 106.

If you specify the keyword `javatruststore`, then all of the certificate authority certificates in the default Java truststore are available for use. A Java truststore is a Java keystore file containing the collection of trusted certificate authority certificates. The default Java truststore includes the required DigiCert Global Root G2 certificate, and is located relative to the Java home directory, which is specified on the `javahome` attribute, or on the SMPJHOME DD statement.

javahome
Specifies the location for the Java runtime to be used by SMP/E. SMP/E uses the capabilities of Java for its HTTPS operations. Therefore, specify the location for the IBM 31-bit SDK for z/OS, Java Technology Edition Version 8 (5655-DGG), or the IBM 64-bit SDK for z/OS, Java Technology Edition Version 8 (5655-DGH), or a logical successor, on the `javahome` attribute.

<HTTPPROXY>
Optional tags are available to describe local HTTP or SOCKS proxy servers. A proxy server redirects HTTP requests to the remote server on the Internet. The <HTTPPROXY> tag is used to identify a local HTTP proxy server, and the <HTTPSOCKSPROXY> tag is used to identify a local SOCKS proxy server.

The <HTTPPROXY> and <HTTPSOCKSPROXY> tags are optional, and should be specified only if HTTP requests to the Internet from your z/OS system are required to pass through a specific HTTP or SOCKS proxy server. For example, if you must specify a proxy server in your Internet browser configuration to allow you access to websites on the internet, then you might need to specify the <HTTPPROXY> or <HTTPSOCKSPROXY> tag in the CLIENT data set.

If you use an HTTP or SOCKS proxy server, and it requires authentication, the `user` and pw attributes can be used to specify the proxy server user ID and password. Also, if your HTTP or SOCKS proxy server listens on a port other than the default ports 80 and 1080, the `port` attribute can be used to specify an alternate port value. For example:

```
<HTTPPROXY host="local.httpproxy.com"
          user="userid" pw="password" port="8080"></HTTPPROXY>
```

See *z/OS SMP/E Commands* for complete details of the <HTTPPROXY> and <HTTPSOCKSPROXY> tags and attributes, and consult your network administrator for help determining what, if anything, you must specify for an HTTP or SOCKS proxy server.

Options that affect FTPS operations

The following example <CLIENT> tag tells SMP/E files should be downloaded using the FTP protocol. The configuration of the z/OS FTP client program is affected by the FTP.DATA file specified in this example, and determines if the files will be downloaded in a secure and encrypted manner.

```
<CLIENT
  downloadmethod="ftp"
  >
  <FTPOPTIONS>-v -f "//'USER1.FTP.DATA'"</FTPOPTIONS>
  <FIREWALL>
    <SERVER host="local.ftpproxy.com"> </SERVER>
    <FIRECMD>&REMOTE_USER;@&REMOTE_HOST;</FIRECMD>
    <FIRECMD>&REMOTE_PW;</FIRECMD>
  </FIREWALL>
</CLIENT>
```

downloadmethod
Identifies the network protocol to use for downloading files from the remote server to the local z/OS. The IBM secure delivery server supports values of either `https` or `ftp`. Specify a value of `ftp` to indicate files will be downloaded using FTP. This is the default if the attribute is not specified at all.

Firewall servers

In order for SMP/E to login to the IBM secure delivery server using FTP, it may be necessary to navigate a local FTP firewall server. If so, optional tags are available in the CLIENT data set to describe information necessary to navigate your local FTP firewall server.

The <SERVER> tag is used to identify the FTP firewall server. You can also specify a user ID and password if your firewall server requires authentication. The <FIRECMD> tags are used to identify the FTP client commands necessary to navigate your firewall server. The commands you specify in the <FIRECMD> tags should be the same as those you use with the z/OS Communications Server FTP client (PGM=FTP). The behavior of firewall servers differ, therefore, the best method to determine what you should specify in the <FIRECMD> tags is to login to an external FTP server using the z/OS Communications Server FTP client in a JCL job, and then specify the same commands in the <FIRECMD> tags. For example:

```
//jobname  JOB …
//FTP      EXEC PGM=FTP,PARM='testcase.boulder.ibm.com'
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
anonymous      ; user ID for remote FTP server
email_address  ; password for remote FTP server
cd mvs/toibm   ; simple change directory command
quit           ; done, so log off
/*
```

This sample job logs into the IBM Testcase FTP server, and assumes that there are no special commands or login procedures required to navigate a local firewall server. If such a job works for you, then no firewall server information needs to be specified in your CLIENT data set for the RECEIVE command or GIMGTPKG service routine. However, if you have a local firewall server that requires such commands or login procedures, you must account for them. For example, if your working FTP job is like this:

```
//jobname  JOB …
//FTP      EXEC PGM=FTP,PARM='local.ftpproxy.com'
//OUTPUT   DD SYSOUT=*
//INPUT    DD *
anonymous@testcase.boulder.ibm.com
email_address  ; password for remote FTP server
cd mvs/toibm   ; simple change directory command
quit           ; done, so log off
/*
```

Then your CLIENT data set should include the following definitions:

```
<FIREWALL>
  <SERVER host="local.ftpproxy.com"> </SERVER>
  <FIRECMD>&REMOTE_USER;@&REMOTE_HOST;</FIRECMD>
  <FIRECMD>&REMOTE_PW;</FIRECMD>
</FIREWALL>
```

The substitution variables &REMOTE_HOST;, &REMOTE_USER;, and &REMOTE_PW; are replaced automatically by SMP/E during processing with the specific values for the FTP server host name, user ID, and password that are correct for the current operation. SMP/E uses the specified commands to log into the FTP server.

See *z/OS SMP/E Commands* for complete details of the <FIREWALL>, <SERVER>, and <FIRECMD> tags, as well as a list of the substitution variables that can be used within the firewall commands. Consult your network administrator for help determining what, if anything, you must specify to navigate your local FTP firewall server.

FTP.DATA configuration file

The FTP.DATA configuration file is used by the z/OS Communications Server FTP client program to define operational settings for the FTP client. Certain functions of the FTP client program, such as performing file transfers in a secure mode (FTPS) or properly navigating local SOCKS firewalls, require the use of a configuration file (FTP.DATA) for the FTP client program. You can use the -f FTP parameter to identify a specific FTP.DATA file, and you can specify the -f parameter using the <FTPOPTIONS> tag in the CLIENT data set, like this:

```
<FTPOPTIONS>-v -f "//'USER1.FTP.DATA'"</FTPOPTIONS>
```

Using the -f parameter overrides the default search order used by the FTP client program to find the FTP.DATA file. If you do not specify the -f parameter, the default search order for the FTP.DATA file is as follows:

1. $HOME/ftp.data
2. userid.FTP.DATA
3. /etc/ftp.data
4. SYS1.TCPPARMS(FTPDATA) data set
5. tcpip_hlq.FTP.DATA file

As of May 1, 2021, to download files from IBM's secure delivery server using FTPS, it is necessary to enable TLS 1.2 in the z/OS Communications Server FTP client program. To enable the FTP client program for TLS 1.2, there are several statements in the FTP.DATA file that must be considered as follows:

```
SECURE_FTP        REQUIRED
SECURE_MECHANISM  TLS
TLSRFCLEVEL       RFC4217
TLSMECHANISM      ATTLS
SECURE_DATACONN   PRIVATE
EPSV4             TRUE
```

SECURE_FTP
This statement specifies whether a security mechanism is optional or required by the FTP client. ALLOWED indicates a security mechanism is optional and the FTP client will allow both secure traffic and non-secure traffic. REQUIRED indicates a security mechanism is required and the FTP client will allow only secure traffic. Either ALLOWED or REQUIRED must be specified.

SECURE_MECHANISM
This statement specifies which security mechanism to use when a session with the server is established. The TLS parameter must be specified.

TLSRFCLEVEL
Use this statement to specify the level of RFC 4217 that FTP operations will support. RFC4217 allows the client and IBM's download server to communicate properly and must be specified.

TLSMECHANISM
Use this statement to specify whether TLS is implemented by AT-TLS or by FTP. ATTLS indicates TLS processing is performed by AT-TLS, and must be specified in order to support TLS 1.2 which is required by IBM's download server.

SECURE_DATACONN
This statement indicates the minimum level of security to be used for data connections by the FTP client. NEVER indicates data must never be enciphered during transfer. CLEAR indicates data may be transferred either with no security or may be enciphered, and is the default value. PRIVATE indicates data must be transferred enciphered. The IBM secure delivery FTP server requires that data be transferred enciphered. Therefore, you must specify PRIVATE for the SECURE_DATACONN statement.

EPSV4
This statement directs the FTP client to use the EPSV and EPRT FTP commands during an FTP session. If you have trouble establishing a secure and encrypted data connection to the secure FTP server through a Network Address Translation (NAT) firewall, specifying TRUE for the EPSV4 statement can help.

The KEYRING statement is ignored when TLSMECHANISM is ATTLS. The certificate authority to be used during the TLS handshake is defined by TCPIP AT-TLS policy, not in the FTP.DATA file. Be sure the policy indicates to use the DigiCert Global Root G2 certificate. If this certificate is not already present in the security manager database, add the certificate as described in “Enabling certificate authority certificates” on page 106.

For information about the contents of the FTP.DATA file and the `-f` parameter, see *z/OS Communications Server: IP User's Guide and Commands*

Example

Here is an example job using the SMP/E GIMGTPKG service routine to download a product or service package from the IBM secure delivery server using the HTTPS protocol:

```
//jobname  JOB ...
//step     EXEC PGM=GIMGTPKG
//SMPNTS   DD PATH='/u/smpe/smpnts/',PATHDISP=KEEP
//SMPOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCLNT  DD *
  <CLIENT
    javahome="/usr/lpp/java/J8.0"
    downloadmethod="https"
    downloadkeyring=”javatruststore”>
  </CLIENT>
/*
//SMPSRVR  DD *
  <SERVER
    host="download3.boulder.ibm.com"
    user="B161a167"
    pw="p5736b6D">
    <PACKAGE
      file="2014011762984/PROD/GIMPAF.XML"
      hash="E01777A6B0B45BCC919054DABCD0B17FBED6859B"
      id="U00563746">
    </PACKAGE>
  </SERVER>
/*
```

This sample job instructs the GIMGTPKG service routine to download the package files using the HTTPS protocol, using the certificate authority certificates in the default Java truststore. The package of files resides on an IBM server, described by the <SERVER> information, which is unique for a particular order and is provided by IBM.

Chapter 6. Preparing to verify signatures for GIMZIP packages

IBM z/OS product and service offerings consist of GIMZIP packages containing already installed software products, or SMP/E consumables such as SYSMODs, RELFILE data sets, and HOLDDATA. These GIMZIP packages are digitally signed. Signing a GIMZIP package, and then verifying the signature of that package, increases confidence in the authenticity (who produced it?) and the integrity (has it changed in transit?) of the package.

SMP/E and z/OSMF provide capabilities to verify the signatures of the GIMZIP packages for these product and service offerings. However, verifying signatures requires preparation and one-time setup.

Note: Support for signing GIMZIP packages and verifying the signatures for those packages is added to SMP/E V3.7 with APAR IO28360.

Digital signature overview

GIMZIP package signing is implemented using public/private key technology; a private key is used to generate a digital signature for package files, and the corresponding public key is used to verify the signatures. The key pair is associated with an X.509 certificate and SMP/E uses this certificate, and its associated key pair, for both the signing and signature verification operations.

A certificate used in the signing process, the signing certificate, is often issued by a well known and trusted certificate authority (CA). The certificate authority is used to establish the authenticity of the package signer (is the signer who they say they are?). If the certificate authority is trusted, so then a signing certificate issued by that certificate authority can also be trusted. Therefore, if you want to verify the signature of signed GIMZIP packages, you must tell SMP/E which trusted certificate authorities may be used to validate the signing certificate, and determine if the signer of the package is trusted.

The signing certificate for the GIMZIP packages produced for IBM's z/OS product and service offerings is issued by the IBM z/OS certificate authority, STG Code Signing Certificate Authority - G2.

Enabling certificate authority certificates

Certificate authority certificates are often managed and stored in your security manager database on z/OS. However, to display and use certificate authority certificates stored in your security manager database requires proper authorization.

The certificate authority certificate for the signing certificates for IBM's GIMZIP packages must be in your security manager database so SMP/E can validate the signing certificate. Therefore, if the STG Code Signing Certificate Authority - G2 certificate is not already in your security manager database, you will need to add it. The actions you must take to check for the certificate are further described in this topic.

Note: The z/OS Security Server (RACF) or an equivalent security manager product on z/OS are used to store and manage x.509 certificates. The remainder of this topic assumes you are using RACF. If you are using an equivalent security manager product, you should refer to that product’s documentation to understand the equivalent actions.

Access to certificate authority certificates

To determine if the required CA certificate is already in your RACF database requires proper authorization. To display a CERTAUTH (certificate authority) certificate, you must have CONTROL access to FACILITY class resource `IRR.DIGTCERT.LIST` . Use the following RACF command to assign proper authorization:

```
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(userid) ACCESS(CONTROL)
```

© Copyright IBM Corp. 1986, 2026

Displaying certificate authority certificates

Once you have access to display CERTAUTH certificates, you must search for and display the STG Code Signing Certificate Authority - G2 certificate to determine if it is in your RACF database. Search for and display the certificate using the certificate's unique serial number and issuer. The serial number and issuer remain constant no matter what label is assigned to the certificate in your RACF database. Use the following RACF command to display the STG Code Signing Certificate Authority - G2 certificate:

```
RACDCERT CERTAUTH LIST(SERIALNUMBER(00) +
 ISSUERSDN('CN=STG Code Signing CA - G2.OU=IBM Code Signing.O=IBM Corp+
 oration.C=US')
```

If the certificate is found in your RACF database, you will see the certificate information, like this:

```
Label: STG Code Signing CA - G2
Certificate ID: 2QiJmZmDhZmjgeLjx0DDloSFQOKJh5WJlYdAw8FAYEDH8kBA
Status: NOTRUST
Start Date: 2013/06/01 00:00:00
End Date: 2033/05/31 23:59:59
Serial Number:
 >00<
Issuer's Name:
 >CN=STG Code Signing CA - G2.OU=IBM Code Signing.O=IBM Corporation.C=U<
 >S<
Subject's Name:
 >CN=STG Code Signing CA - G2.OU=IBM Code Signing.O=IBM Corporation.C=U<
 >S<
Signing Algorithm: sha256RSA
Key Usage: CERTSIGN
Key Type: RSA
Key Size: 2048
Private Key: NO
Certificate Fingerprint (SHA256):
 38:AF:8E:66:CB:9B:36:97:82:EF:0A:7D:51:D5:2A:61:
 65:D9:E8:67:A8:8A:53:0B:31:FE:6B:80:59:4C:74:E5
```

Make note of the label for the found certificate, as the label for the certificate in your RACF database may be different than `STG Code Signing CA - G2`. You must use the actual label value in the subsequent RACF commands.

If the certificate is found but has a Status of NOTRUST, then you must use the following RACF command to trust the certificate:

```
RACDCERT CERTAUTH +
 ALTER(LABEL('STG Code Signing CA - G2')) TRUST
```

If the certificate is not found, RACF initialization will automatically add it during the next IPL of your z/OS system. For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

Create a key ring

A key ring is a named collection of certificates associated with a specific user. A certificate is identified by its label and the key ring that it is connected to. Connect the STG Code Signing Certificate Authority - G2 certificate to a key ring you will use when validating package signatures. Use the following RACF commands to create a key ring and connect the CA certificate to it:

```
RACDCERT ID(userid) ADDRING(IBM.package.signature.verification)
RACDCERT ID(userid) CONNECT( CERTAUTH +
 LABEL('STG Code Signing CA - G2') +
 RING(IBM.package.signature.verification) +
 USAGE(CERTAUTH) )
```

where `userid` is the user ID that will own the key ring and `IBM.package.signature.verification` is a name for the key ring. You can choose any name you desire for the key ring, as this is the key ring you will provide to SMP/E for verifying the digital signature of IBM signed GIMZIP packages.

Access for SMP/E to use certificate authority certificates

Now that the appropriate CA certificate is in your RACF database and connected to a key ring, you must ensure SMP/E can access the certificate and the key ring that contains the certificate. The user ID under which SMP/E runs must be properly authorized to the FACILITY class resource `IRR.DIGTCERT.LISTRING` or to the RDATALIB class resource `keyring-owner.keyring-name.LST` for SMP/E to use a specified keyring. READ access is required for a user ID to use its own key ring or the CERTAUTH virtual keyring. Use the following RACF command to ensure the user ID has access to read the key ring:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

If the RDATALIB class is used, for any real keyring, READ access to `keyring-owner.keyring-name.LST` is required. For the CERTAUTH virtual keyring, READ access to `CERTIFAUTH.IRR_VIRTUAL_KEYRING.LST` is required.

Refreshing RACF classes

After you perform RACF updates to add a key ring and connect the certificate to the key ring, you might need to refresh the in-storage RACF profiles. That is, if you choose to RACLIST the DIGTCERT and DIGTRING classes, you must first activate them using the following RACF command:

```
SETROPTS CLASSACT(DIGTCERT DIGTRING)
```

If you have used RACLIST for the DIGTCERT or DIGTRING classes, you need to refresh the in-storage profiles before the updates can take effect. You can use the following RACF command:

```
SETROPTS RACLIST(DIGTCERT DIGTRING) REFRESH
```

If you have not used RACLIST for the DIGTCERT or DIGTRING classes, you do not need to refresh the in-storage profiles.

Define CLIENT input for RECEIVE, GIMGTPKG and GIMUNZIP

The CLIENT data set is used by the SMP/E RECEIVE command, the GIMGTPKG and GIMUNZIP service routines to provide SMP/E with information about the local z/OS system and network, as well as certain processing preferences. You also identify the keyring for validating the digital signatures of GIMZIP packages in the CLIENT data set.

The information in the CLIENT data set is described using the <CLIENT> tag and attributes that are defined in detail in *z/OS SMP/E Commands*.

The following example <CLIENT> tag tells SMP/E that files should be downloaded using the HTTPS protocol, thus enabling secure and encrypted transfers, and that if the GIMZIP package is digitally signed, its signature will be verified using the certificate authority certificates in the identified keyring.

```
<CLIENT
 downloadmethod="https"
 downloadkeyring="javatruststore"
 signaturekeyring="userid/IBM.package.signature.verification"
 >
</CLIENT>
```

signaturekeyring
: Identifies the location of the Certificate Authority (CA) certificate required to verify the digital signature of the GIMZIP package. Specify the name for a security manager keyring. The specified name may be for a real or virtual keyring. If the keyring is associated with a userid other than that used to execute SMP/E, then the keyring name must be prefixed with the associated userid. The userid is separated from the keyring value by a forward slash (userid/keyring).

To indicate all the Certificate Authority (CA) certificates defined in the security manager may be used to verify the digital signature, use the CERTAUTH virtual keyring by specifying the userid/keyring value "*AUTH*/*".

Note: The userid under which SMP/E runs must be properly authorized to the FACILITY class resource `IRR.DIGTCERT.LISTRING` or to the RDATALIB class resource `keyring-owner.keyring-name.LST` for SMP/E to use the specified keyring. READ access is required for a userid to use its own keyring or the CERTAUTH virtual keyring. UPDATE access is required to use a keyring from another user. If the RDATALIB class is used, for any real keyring, READ access to `keyring-owner.keyring-name.LST` is required. For the CERTAUTH virtual keyring, READ access to `CERTIFAUTH.IRR_VIRTUAL_KEYRING.LST` is required.

Chapter 7. Installing a new function

This section discusses the method you use to install a new function. It describes:

- Sources of new functions and sources of installation information
- An example of installing a function

Introduction

The primary purpose of SMP/E is to help you install SYSMODs in your target and distribution libraries. To do this, SMP/E provides three basic commands: RECEIVE, APPLY, and ACCEPT.

This topic summarizes the general steps you follow to install a function. You should look at the installation materials that arrive with a function to find out about special requirements and procedures for installing the function. Table 10 on page 117 lists sources of new functions and places where you can find information for installing the new functions.

Table 10. Sources for functions and their installation information

Product delivery vehicle	Products and service you get	Installation materials you can use
CBPDO	Products and service (not integrated) on a single logical tape	Sample jobs to receive products and service Program directories for the products you ordered Installation information for the products you ordered
Independent product	Product tape	Program directory for the product Installation information for the product (if one is provided)

RECEIVE-APPLY-ACCEPT method

RECEIVE-APPLY-ACCEPT is the standard installation method. It uses SMP/E RECEIVE, APPLY, and ACCEPT commands to install functions onto a subsystem. Usually, you do not have to do any special processing outside SMP/E to install your functions. The JCLIN needed to set up the load modules for the function is sent along with the function.

In this method of installation, you:

1. Use the RECEIVE command to get the SYSMODs from the input data set and put them in the SMP/E data sets (the PTS and global zone within the CSI).
2. Use the APPLY command to install the SYSMODs in the target system libraries; then test them as required.
3. Use the ACCEPT command to install the SYSMODs into the distribution libraries.

The standard RECEIVE-APPLY-ACCEPT method

This section describes the standard process for using the RECEIVE, APPLY, and ACCEPT commands to install a function.

© Copyright IBM Corp. 1986, 2026

Note: You can use either the SMP/E dialogs or JCL jobs to receive, apply, and accept functions. The basic steps to follow are the same. If you have access to the SMP/E dialogs, you should use them. Otherwise, you can use the steps described in this chapter as examples.

Preparing your system

Before you start doing any operations on the product function or service tape you have received, there is work you should do to your system to make sure it is ready and to be certain you can recover in case of a serious failure during installation.

Following are some of these steps:

1. Read the documentation for your new product. This includes the program directory and, if provided, an installation guide. Also check the IBM Preventive Service Planning (PSP) files for the latest information about the product. This is important because there might be a PTF for the product that is not included in an ESO, or one of the PTFs may contain an error you should know about.
2. When you order a product, update the FMID list in the global zone with the FMIDs of the products you that you are receiving. (Check the program directory for this information.) After you make the update, you will receive any preventive service that is shipped between the time you order the product and the time you install it.
3. Read the program directory. It tells you which libraries are affected, whether any existing libraries must be expanded and by how much, and whether any new libraries are required.
4. Prepare the target and distribution libraries. If these libraries are properly prepared before you apply or accept a SYSMOD, little time is lost if an error occurs.

   List the VTOC of the target and distribution packs. This shows you which data sets are into their secondary extents, or are too full to contain additional elements that might be applied or accepted. If you are unsure how large a data set will grow, you may want to check full data sets against the SYSMODs you will be processing.

   Partitioned data sets with a high percentage of their space used can be compressed by use of IEBCOPY. If more space might be needed after the compression, allocate a larger data set and copy the nearly full data set into it; then delete the old data set. Rename the new one properly and, if it had to be allocated on a different pack, update any procedure necessary with the new VOLUME data.

   This preparation is time-consuming but takes less time and work than recovering from an out-of-space abend (E37, B37, and so on).

   SMP/E command operands can also help you handle out-of-space abends.

   - The COMPRESS operand tells SMP/E to compress the data sets before they are updated; this can help you avoid an *x*37 ABEND. For more information about the COMPRESS operand, see *z/OS SMP/E Commands*.
   - The RETRY(YES) operand tells SMP/E to attempt recovery after an *x*37 ABEND occurs by compressing the affected data sets and retrying the failing utility. If you still need space after SMP/E's initial retry attempt and input to the utility was batched (copy or link-edit utility only), SMP/E debatches the input and retries the utility separately for each member. For information about this retry processing, see “Recovering after errors from utility processing” on page 75.
5. Allocate any new libraries required. Determine where they are to be allocated and then allocate them. Remember that the program directory ordinarily shows how much space will be used. It does not show how much space to allocate for the libraries. Allocate the libraries with more space than required to allow for later modifications. Usually, twice the required space is recommended to allow for the replacement of every element in the library without running out of space.

   Remember to add the appropriate DDDEF entries to the target zones and DLIB zones into which you will install this function.
6. Check the SMP/E data sets to make sure that they have enough space. If necessary, compress or expand the partitioned data sets. A data set that is easily overlooked in this process is the SMPSTS, which fills rapidly when you are receiving source updates (JES2 and JES3, for example). Reorganize or expand (if necessary) the CSI data set (using access method services EXPORT and IMPORT).

7. Create a backup for the volumes affected. This is a important step that should not be overlooked. Without a current backup copy, a serious system failure during installation means not only redoing the installation in process but also means going to the last backup level and redoing all the work done since then.
8. Estimate the time required for APPLY and ACCEPT processing. Make sure that enough time is available to allow these jobs to run to completion.

   The program directory or installation guide might contain information to help you estimate the time required.

Staging the SYSMODs: The RECEIVE process

When you get a new function as part of a CBPDO, you get one logical tape that contains the function and the unintegrated service. If there is no preventive service for the function, it is not included in your order.

The first step in installing the new function is the RECEIVE process, which reads in the SYSMODs so they can be installed later:

- If you have access to the SMP/E dialogs, you can use the RECEIVE dialog to receive the function and any related service and HOLDDATA.
- You can use the RIMLIB job included on the CBPDO package to receive the function, service, and HOLDDATA shipped on the CBPDO. For more information, see *z/OS Planning for Installation* and the documentation that was included with the CBPDO.

Receiving the function SYSMOD

Function SYSMODs obtained from IBM are packaged in RELFILE format. Before any actual processing takes place, SMP/E must first determine if the SYSMODs packaged in RELFILE format are to be received from tape or DASD. If the SMPPTFIN data set is located on tape, SMP/E assumes that the RELFILEs are on tape. If the SMPPTFIN data set is on DASD, SMP/E assumes the RELFILEs are on DASD and are cataloged.

During RECEIVE processing, the contents of the RELFILEs are placed into the SMPTLIBs, which are used as temporary storage. SMPTLIBs that are uncataloged are automatically cataloged by SMP/E. When the RELFILEs are on DASD, SMP/E checks to ensure that the RELFILE and SMPTLIB names are not the same. If they are, RECEIVE processing stops.

Note: Do not delete the SMPTLIBs after the RECEIVE step; they must be retained until after the function is applied and accepted.

Updating the target libraries: The APPLY process

After preparing your target and distribution libraries and receiving the function and any related service and HOLDDATA, the next step is to update your target libraries. Review the program directory for the products you are installing.

When installing a new function, you are concerned with three groups of SYSMODs:

- The function itself
- All PTFs applicable to the function
- All PTFs applicable to other functions that are specified as requisites of the function or service applicable to the function

You might be able to apply all the required SYSMODs with one APPLY command. This method has several advantages:

- It eliminates the need to run the APPLY command several times in order to install the complete set of SYSMODs required.
- You replace elements in the target libraries less often; therefore, there is less risk of running out of space.

- Because the SMP/E overhead and the number of invocations of the system utilities are reduced, overall processing time is decreased.

Therefore, although SMP/E supports the separate installation of a new function and its service, the common installation method is preferred unless the product program directory for other unique installation requirements directs otherwise. This is the method illustrated in subsequent examples. For more information about the APPLY command operands, see *z/OS SMP/E Commands*.

When you are updating the target libraries, there are actually three distinct SMP/E jobs to be run:

Receive additional HOLDDATA
Before starting the APPLY, you should get the latest HOLDDATA by either using the RECEIVE ORDER command or manually submit a request for HOLDDATA using ShopzSeries.

Run the APPLY CHECK job.
This is a nonupdating mode of APPLY. Its purpose is to help resolve any problems that may prevent the APPLY from completing processing successfully.

Apply the SYSMOD updates
This installs the new function and service into the target libraries.

Checking the update: The APPLY CHECK process

The purpose of this step is to determine:

- Whether any errors will occur while the new function is being applied (except for errors that occur as a direct result of an update, such as a target library running out of space). This includes missing DDDEF entries.
- Whether any requisite SYSMODs are missing.
- The target libraries that will be updated.
- The SYSMODs, if any, that will be regressed.

Use the SMP/E dialogs or the following sample job to do an APPLY CHECK for the function and related SYSMODs:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(ZOSTGT1)       /* Set to target zone.      */.
APPLY    FORFMID(HXX1200)   /* Apply for this FMID      */
         FUNCTIONS          /*  the function itself     */
         PTFS               /*  and all its PTFs.       */
         GROUPEXTEND        /* Also all requisite PTFS. */
         CHECK              /* But do not update libs.  */
         BYPASS(HOLDSYSTEM  /* Bypass options           */
                HOLDCLASS(UCLREL,ERREL)
               )
         .
/*
```

Researching the APPLY CHECK reports

As a result of running the APPLY CHECK job, SMP/E produces various messages and reports that you should now use to do further research. Here are some of the errors that might have been detected:

- Some DD statements might be missing. Check the program directory or *z/OS SMP/E Reference* to determine why they are required and how they should be specified.
- Some APAR fixes or USERMODs might be regressed. If so, you must determine why. For APAR fixes, you have to get the version of the APAR fix applicable to the new product. For USERMODs, you have to rework the modification to make it applicable to the new function, or eliminate the modification if the product being installed provides the same function. When doing the actual APPLY operation, you may need to specify the BYPASS operand to inform SMP/E that you have resolved these problems.
- Some prerequisite or requisite PTFs might be missing. If so, you should determine whether they can be obtained. Some may already be on an ESO tape you have in-house but have not received; others may not have been shipped, in which case you have to get an early copy of them by contacting the IBM

Support Center. Although you can also avoid these conditions by using the BYPASS operand, you are advised not to do this because the regressions have not been resolved.

Note: Obtaining a product in a CBPDO greatly reduces the amount of work needed to find requisite PTFs, because CBPDOs include all the service for products applicable to the selected SREL.

- Some elements might not have been selected for installation. For each such element, if the current functional owner (that is, FMID) is an IBM product, there may not be a problem; this condition is common and occurs because there are multiple functions with common elements. Check the program directory or installation guide for the product you are installing to determine whether this condition is normal or if it indicates a problem.

  If the FMID is not one for an IBM product, further research is necessary. Contact the current owner of the element to determine how that product is related to the one you are installing.
- Some of the PTFs might not have been selected for installation because of exception SYSMOD conditions identified by the ++HOLD MCSs. When installing a new function, you may want to research these PTFs further. You can use the reason ID and the comments specified in the ++HOLD MCS to determine which of the following actions is most appropriate:
  - Bypass the condition using the BYPASS(HOLDERR) operand
  - Do not install the PTF
  - Obtain a fix for the APAR

Getting additional SYSMODs

After doing the research step, you may decide that additional SYSMODs are needed. If so:

1. Obtain the additional SYSMODs by using CBPDO, ESO, IBMLINK, or the IBM Support Center.
2. Receive the additional SYSMODs, using the same source ID value as used when processing the CBPDO package.
3. Rerun the APPLY CHECK job.

Repeat this process until no new errors are reported.

Updating the target library: The APPLY process

After you complete the APPLY CHECK and the associated research and the other necessary preparation, the APPLY job itself should be fairly simple. The APPLY job does the same checking as the APPLY CHECK and then continues by calling the appropriate system utilities to get all the elements installed.

Note: If a product deletes another product, you cannot use the RESTORE command to back off the applied product and bring back the deleted one.

Use the SMP/E dialogs or the following sample job to apply the function and any related SYSMODs:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLY    EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(ZOSTGT1)       /* Set to target zone.      */.
APPLY    FORFMID(HXX1200)   /* Apply for this FMID      */
         FUNCTIONS          /*   the function itself    */
         PTFS               /*   and all its PTFs.      */
         GROUPEXTEND        /* Also all requisite PTFs. */
                            /* No check this time.      */
         BYPASS(HOLDCLASS(UCLREL,ERREL)/*Bypass options.*/
                HOLDSYS(reason-id,…) .
/*
```

If you obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand, or if **all** applicable APARs and USERMODs are to be applied, specify the APARS and USERMODS operands.

Note: For most products, you do not have to do any additional processing to get the elements into executable format. However, some products may require you to run additional utilities or to perform extra steps after applying the SYSMODs. See your product documentation for more information.

Testing the new function

After installing the new function, you should perform two operations:

1. Create a backup of the updated data sets, including any SMP/E data sets affected, in case something happens to the data sets during the next phase.

   Note: If you are doing the installation on a clone of your original system, you already have a backup—your original system.

2. Do some testing before putting the new function into production. This testing should include some of the following:
   - If the product has supplied installation verification procedures (IVPs), they should be run.
   - If your installation has a test job stream, the tests should be run.
   - If the new function could at all affect your ability to IPL the system, try an IPL at this time.

Only after verifying that the new function on a noncritical test system should you put it into production.Do not consider the test phase completed until you run the new function in production mode for some period (as determined by your installation requirements). Only then, if no errors are found, are you ready to go to the next step, updating your distribution libraries.

Updating the distribution libraries: The ACCEPT process

The last major phase of installing a new function is to update the distribution libraries, using the SMP/E ACCEPT command. This is a critical step: Once the function and its service have been accepted, there is no SMP/E method for removing it from either the target or distribution libraries.

When you are ready to update your distribution libraries, you have the same set of considerations and SMP/E support as described under "Updating the target libraries: The APPLY process" on page 119, and the same three-phase operation:

Receive additional data
Before starting the ACCEPT process, you should obtain the latest HOLDDATA using the RECEIVE ORDER command or manually submitting a request for HOLDDATA using ShopzSeries.

If there is a significant time between the APPLY process and ACCEPT process, additional problems may have been reported for which ++HOLD statements have been created. If this new data is not obtained, SMP/E may install PE PTFs into the distribution libraries.

Run the ACCEPT CHECK job
The ACCEPT CHECK job is a nonupdating mode of ACCEPT. Its purpose is to help resolve any problems that may prevent the ACCEPT from completing processing successfully.

Accept the SYSMOD update
Accepting the SYSMOD update installs the new function and service into the distribution library.

Checking the update: The ACCEPT CHECK process

The ACCEPT CHECK job provides the same function for the distribution libraries that the APPLY CHECK job provides for the target libraries. See "Checking the update: The APPLY CHECK process" on page 120.

Use the SMP/E dialogs or the following sample job to do an ACCEPT CHECK for the function and related SYSMODs:

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPTCK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1)         /* Set to DLIB zone.        */.
ACCEPT   FORFMID(HXX1200)   /* Accept for this FMID     */
         FUNCTIONS          /*   the function itself    */
         PTFS               /*   and all its PTFs.      */
         GROUPEXTEND        /* Also all requisite PTFs. */
         CHECK              /* But do not update libs.  */
         BYPASS(HOLDSYSTEM  /* Bypass options           */
                HOLDCLASS(UCLREL,ERREL)
               )
```

```
/*
```

Researching the ACCEPT CHECK reports

Research the ACCEPT CHECK reports in the same manner as the APPLY CHECK reports . For more information, see “Researching the APPLY CHECK reports” on page 120.

Getting additional SYSMODs

The process of getting additional SYSMODs or APAR fixes for those PTFs being accepted is the same as during the APPLY process. For more information, see “Getting additional SYSMODs” on page 121.

Updating the distribution library: The ACCEPT process

The ACCEPT process updates the distribution libraries. Use the SMP/E dialogs or the following sample job to accept the function and any related SYSMODs:

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPT   EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(DLIB1)         /* Set to DLIB zone.        */.
ACCEPT   FORFMID(HXX1200)   /* Accept for this FMID     */
         FUNCTIONS          /*   the function itself    */
         PTFS               /*   and all its PTFs.      */
         GROUPEXTEND        /* Also all requisite PTFs. */
                            /* No check this time.      */
         BYPASS(HOLDCLASS(UCLREL,ERREL)/*Bypass options */
                HOLDSYS(reason-id,…) .
/*
```

Note: If you obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand or, if **all** applicable APARs and USERMODs are to be installed, specify the APARS and USERMODS operands.

Checking other zones for requisites: REPORT CROSSZONE

After installing a function, you may need to check other zones for conditional requisites. A conditional requisite is software (such as service) that must be installed for a given function if another function is also installed. To help automate the research for conditional requisites, the installation logic (SMP/E modification control statements) for a function uses ++IF statements to identify the requisites.

Conditional requisites may be for functions that are installed in different zones. If you have set up automatic cross-zone requisite checking, as described in “Specifying automatic cross-zone requisite checking” on page 83, SMP/E will enforce cross-zone requisites during APPLY or ACCEPT processing. Otherwise, you must use the SMP/E REPORT CROSSZONE command after a function and the related service has been installed to manually identify cross-zone requisites defined in the installation logic. To help you install the identified requisites, REPORT CROSSZONE can also write APPLY and ACCEPT commands to the SMPPUNCH data set. So, if you have **not** specified automatic cross-zone requisite checking, and the function you have installed (or any related service) specifies conditional requisites, you should run the REPORT CROSSZONE command against the target and DLIB zones containing that product, as well as against the zones containing the functions identified on the ++IF statements. For more information about the REPORT CROSSZONE command, see Chapter 15, “Identifying cross-zone requisites: The REPORT CROSSZONE command,” on page 161.

Chapter 8. Installing preventive service

This topic describes the steps for installing preventive service. After an introduction to preventive service and a summary of the preventive service process, it discusses the following topics:

- Preparing your system
- Staging the SYSMODs with the RECEIVE command
- Requesting preventive service with the RECEIVE ORDER command
- Updating the target libraries with the APPLY command
- Testing the new service level
- Updating the distribution libraries with the ACCEPT command

Introduction

You install preventive service through the use of the SMP/E preventive service process. The process uses:

- The preventive service SYSMODS received as a package downloaded from the IBM Server as a result of a RECEIVE ORDER request, or a request submitted manually on ShopzSeries.
- A well-defined set of steps that you should follow to install each service level in order to bring your system up to the current service level.

A RECEIVE ORDER request

You can order all currently available PTFs that meet your selection criteria. The PTF package that results from such an order is tailored to your SMP/E environment and contains all currently available PTFs that match your selection criteria and are not already present in your environment. There are three selection options:

Critical
: Critical service includes all available PTFs that resolve high impact pervasive (HIPER) problems or PTFs in error (PEs).

Recommended
: Recommended service includes all available PTFs identified with a Recommended Service Update SOURCEID (RSUyymm) and all available PTFs that resolve a critical problem (HIPER or PE). Recommended service includes PTFs through the most current RSU level.

All
: All service includes all available PTFs.

The package also contains the last 2-years of Enhanced HOLDDATA for the entire z/OS software platform.

Preventive service process: Summary

The preventive service process has five phases:

1. Prepare your system.
2. Stage the preventive service.
3. Update the target libraries.
4. Test the system.
5. Update the distribution libraries.

The preventive service phases are the same as those defined in Chapter 9, “Installing corrective service,” on page 133, although the steps within each phase differ. These phases are the basic SMP/E operations to install any SYSMOD.

© Copyright IBM Corp. 1986, 2026

Each of these phases consists of a series of steps (SMP/E jobs, research, or invocations of system utilities) that must be done to make sure the preventive service is installed correctly and to ensure the integrity of your system libraries.

The steps defined are the normal steps for the installation of most PTFs. Any PTF that requires special processing will contain instructions for installing it.

Generally, you should first attempt to install all the normal PTFs that you have received and then to install those having special requirements. The intention is to install the maximum number of preventive fixes on your system as soon as possible.

Note: You should let SMP/E manage PE PTFs (PTFs discovered to be in error), rather than researching and resolving them yourself.

The following sections describe, in detail, the steps necessary for each of the preventive service phases.

Note: You can use either the SMP/E dialogs or JCL jobs to receive, apply, and accept preventive service. The basic steps are the same. If you have access to the SMP/E dialogs, you should use them. Otherwise, you can use the steps described in this chapter as examples.

Preparing your system

Before you start installing preventive service, you should do the following to make sure that your system is ready and that you can recover in case of a serious failure during installation:

- Get the latest HOLDDATA. You process this HOLDDATA during the next phase. If you used the RECEIVE ORDER command to request a preventive service package, it contains the last 2-year's HOLDDATA for the z/OS software platform, so you can skip this step.
- Make sure you have the publications you need.
- Estimate the time you need for APPLY and ACCEPT processing. Make sure that there is time for these jobs to run to completion.
- Back up your disk packs.

Staging the SYSMODs: The RECEIVE process

After you prepare your target and distribution libraries, receive the preventive service SYSMODs (PTFs) and the HOLDDATA into the SMP/E database (the global zone and the SMPPTS). If you used the RECEIVE ORDER command to request a preventive service package and did not specify TRANSFERONLY, the process downloaded the SYSMODS directly into the SMP/E database (the global zone and SMPPTS), so you can skip this step.

Updating the target libraries: The APPLY process

After you prepare your target and distribution libraries and receive all the necessary PTFs and HOLDDATA, update the target libraries. Though most PTFs can be installed directly into the target libraries, some require special processing, such as a fix that must be concurrently installed on all processors in a network.

These PTFs contain a ++HOLD statement that automatically places them into HOLD for SYSTEM action status; that is, SMP/E does not allow them to be installed unless you take some direct action, such as specifying BYPASS(HOLDSYS) on the APPLY command. These PTFs should not be processed immediately; you should attempt to install all PTFs not requiring such actions and then return to process these. For additional information about these PTFs, see “Installing PTFs that need special processing” on page 130.

When installing preventive service, you are concerned with two groups of PTFs:

- All PTFs from the CBPDO, ESO, or requested service package you are installing
- Any other PTFs that are required to install these PTFs

SMP/E provides operands (SOURCEID and GROUP or GROUPEXTEND) on the APPLY command that facilitate the installation of all required PTFs by use of one APPLY command. Installing all PTFs with one APPLY command provides several advantages:

- It eliminates the need to run the APPLY command several times in order to install the complete set of PTFs required.
- It reduces the risk of running out of space, because you are replacing elements in the target libraries less often.
- It decreases overall processing time, because there is less SMP/E overhead and the system utilities are invoked less often.

When you update the target libraries, there are three distinct SMP/E jobs to be run:

1. **Receive additional HOLDDATA.** Before starting the APPLY, you should contact the IBM Support Center to obtain any additional HOLDDATA for the CBPDO or ESO you are installing. This step is required if:
   a. You did not obtain the additional HOLDDATA from the IBM Support Center during the staging phase.
   b. There was a delay between the RECEIVE and APPLY staging phase and the target update phase.

   We will not discuss this first step further here. If you need to perform this step, see "Staging the SYSMODs: The RECEIVE process" on page 126.
2. **Run the APPLY CHECK job.** This second step is a nonupdating mode of APPLY, referred to as the APPLY CHECK run. Its purpose is to assist in resolving any problems that prevent the APPLY itself from completing processing successfully.
3. **Run the APPLY job.** This third step is the updating mode of APPLY, in which the preventive service is installed into the target libraries.

The following sections describe the last two steps as well as the processing of PTFs that require special processing.

Checking the update (APPLY CHECK)

The purpose of this step is to determine:

- Whether any errors will occur when you apply a SYSMOD (except for those error conditions that occur as a direct result of an update, such as a target library running out of space)
- Whether any requisite PTFs are missing
- The target system libraries that will be updated during APPLY
- The PTFs or APARs, if any, that will be regressed during APPLY

The GROUP and GROUPEXTEND operands allow SMP/E to include any PTF that may be required to install PTFs on the current service level (PUT0703 in the example that follows). Some of the PTFs on previous tapes may not have been installed, because they were in hold status (PE PTFs) at the time the ESO containing the service level was installed. The current service level may contain fixes for the APARs that caused the original PTFs to be held. These PTFs, because they have module intersections with the PE PTF, must either be prerequisite to the old PTFs or must supersede them so SMP/E can automatically include the old PTFs when the fixing PTF is installed.

The following sample job shows how to do an APPLY CHECK for preventive service:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SOURCEID(PUT0703)  /* Apply this service level */
         GROUPEXTEND        /*  and all requisite PTFs, */
         CHECK              /*  but do not update libs. */
         SELECT(sysmod-id,…) /* Select additional     */
                            /* service if required.     */
         BYPASS(HOLDCLASS(ERREL,UCLREL)
                HOLDSYSTEM) .
/*
```

You may be able to improve SMP/E performance by including the source IDs for previous service levels within the SOURCEID operand. The following job provides an example of an APPLY CHECK job for PTFs in service level 0703:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SOURCEID(PUT0703   /* Apply this service level */
                  PUT0702   /*  and back-level tapes    */
                  PUT0701)  /*  back to some reasonable */
                            /*  level.                  */
         GROUPEXTEND        /* And all requisite PTFs.  */
         CHECK              /* But do not update libs.  */
         SELECT(sysmod-id,…) /* Select additional     */
                            /* service if required.     */
         BYPASS(HOLDCLASS(ERREL,UCLREL)
                HOLDSYSTEM) .
/*
```

Note: This form of the SOURCEID operand can also be used to group service levels initially in one APPLY command.

If you want to install preventive service only on selected functional areas of the system, you can also specify the FORFMID operand on the APPLY command, specifying either specific function identifiers (FMIDs) or the name of one or more FMIDSETs.

Researching the APPLY CHECK reports

As a result of running the APPLY CHECK job, SMP/E produces various messages and reports you should now use to perform further research. Here are some of the errors that may be detected:

- Some DD statements may be missing. Check *z/OS SMP/E Reference* to determine why they are required and how they should be specified.
- Some APARs or USERMODs may be regressed. If so, you must determine why. For APARs, obtain the version of the APAR fix applicable to the service level. For USERMODs, rework the modification to be applicable to the new service level. When performing the actual APPLY operation, you most likely need to specify the BYPASS operand in order to inform SMP/E that you have resolved these problems.
- Some requisite PTFs may be missing. If so, you should determine how they can be obtained. Some may be on service levels you have not received; others may not have been shipped, in which case you have to obtain an early copy of them by contacting the IBM Support Center. Although you can get around these conditions by using the BYPASS operand, you are advised not to do this because the regressions have not been resolved.
- Some of the PTFs are not installed because of exception SYSMOD conditions identified by the ++HOLD statements. You should ignore these PTFs until a fixing PTF is delivered in a subsequent service level.

  Note: Depending on your requirement to install such PTFs, you can use the reason ID and the comments specified in the ++HOLD statement to determine which of the following actions is most appropriate:

  – Bypass the condition using the BYPASS(HOLDERR) operand
  – Do not install the PTF
  – Obtain a fix for the APAR

A common cause of regression is user modification. When USERMODs are applied to your system, the service level information (RMID or UMID) is altered to reflect these additions. The APPLY CHECK may have flagged a SYSMOD as one that would cause regression.

This regression-handling procedure works under the assumption that you have applied, but not yet accepted, a USERMOD. This means that the USERMOD has applied service to the target libraries, but the service in your distribution library is that which the SYSMOD should be applied against.

You can follow these steps for handling regression:

1. Restore the module from the distribution library back into the target system to back off the USERMOD.

2. Apply the SYSMOD in question to the target system in order to keep SMP/E's information about the target system up to date.
3. Accept the SYSMOD into the distribution libraries.

When USERMODs are applied on a system, it is up to you to ensure that they are at the proper level.

If you reapply your USERMOD at this point, remember to exclude it when accepting the preventive service if you want to be able to restore your system to the level assumed by the next preventive service update.

The following steps describe regression handling.

1. **Restore APARs or USERMODs, if necessary.** Use the RESTORE command to remove the APAR or USERMOD from the target libraries. This places into the target library the versions of the elements that currently exist in the distribution library.

   Use the SMP/E dialogs or the following sample job to restore the SYSMODs:

```
//SMPRESTR JOB  'accounting info',MSGLEVEL=(1,1)
//RESTORE  EXEC SMPPROC
//SMPCNTL DD *
SET     BDY(TGT1)        /* Set to target zone.      */.
RESTORE                  /* Put DLIB data into       */
                         /*    target libraries.     */
        SELECT(AZ00001,  /* Must be SELECT or GROUP, */
               AZ00002)  /*     NOT by source ID.    */.
/*
```

2. **Repeat the APPLY CHECK.** This gives you an updated status report to determine that all regression conditions have been addressed.
3. **If a USERMOD or APAR is necessary,** compare the PTF just flagged by APPLY CHECK with the APAR or USERMOD that caused the regression. You may need microfiche or a dump of the module. If any changes are needed, follow the steps listed later in this section. Otherwise, continue with the APPLY step.

   - Do one of the following:
     - For a USERMOD, add the REWORK operand to the ++USERMOD MCS. The REWORK operand allows the updated SYSMOD to be automatically rereceived, as long as it is more recent than the version that has already been received. This takes the place of rejecting the SYSMOD and receiving it.
     - For an APAR or USERMOD, reject the prior copy from the SMPPTS.

       The SMP/E REJECT job removes the USERMOD or APAR from the SMPPTS. This prevents the prior copy from being applied again.

       A sample REJECT job follows:

```
//SMPREJ  JOB 'accounting info',MSGLEVEL=(1,1)
//REJECT  EXEC SMPPROC
//SMPCNTL DD *
SET      BDY(GLOBAL)     /* Set to global zone.   */.
REJECT                   /* Remove these two      */
                         /*  SYSMODS from the     */
         S(AZ00001,      /*  PTS and global zone. */
           AZ00002)      /*                       */.
/*
```

   - **Receive the USERMODs or APARs.** This loads the SYSMODs into the SMPPTS.

Getting additional SYSMODs

After doing the research step, you may decide that additional SYSMODs are needed. These should be obtained from the IBM Support Center and then received into the SMPPTS.

At this time, you should modify the APPLY command to add a SELECT operand specifying each of the PTFs obtained from the IBM Support Center. An alternative is to assign all such PTFs the same source ID value as the service level, or to assign them a unique value and then add that value to the SOURCEID operand.

This process should continue until no new errors are reported.

Updating the target library (APPLY)

If the suggested preparation and all phases of the APPLY CHECK are completed, the APPLY job itself should be fairly simple. The APPLY job performs the same checking as the APPLY CHECK and then calls the appropriate system utilities to install all the elements.

Use the SMP/E dialogs or the following sample job to apply the preventive service:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLY    EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SOURCEID(PUT0703)  /* Apply this service level */
         GROUPEXTEND        /*  and all requisite PTFs, */
         SELECT(UZ00001     /* Plus two other PTFs.     */
                UZ00002     /*                          */
                AZ12345     /* Plus two APAR fixes.     */
                AZ12346)    /*                          */
                            /*  No check this time.     */
         BYPASS(HOLDCLASS(ERREL,UCLREL)
                HOLDSYSTEM) .
/*
```

- If you have obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand or, if **all** applicable APARs and USERMODs are to be installed, specify the APARS and USERMODS operands.
- If any of the SYSMODs specified in the SELECT list have already been applied and you want to reinstall them, you must also specify the REDO operand on the APPLY command.
- If you want to install preventive service only on selected functional areas of the system, you can also specify the FORFMID operand on the APPLY command, specifying either specific function identifiers (FMIDs) or the name of one or more FMIDSETs.

Installing PTFs that need special processing

There are so many reasons a PTF may require special processing that it is impossible to document how you should handle each case. Any PTF requiring special processing should contain a ++HOLD statement (after all the ++VER statements and before the first element MCS). That ++HOLD statement should be as follows:

```
++HOLD(sysmod-id)           /* Originating SYSMOD ID.   */
      SYSTEM                /* Special processing info. */
      FMID(sysmod-id)       /* Functional owner.        */
      REASON(reason-id)
      COMMENT(….
              ….any amount of comment text
              ….
             )
      .
```

See *z/OS SMP/E Reference* for a detailed description of the ++HOLD statement syntax. The comment text within the ++HOLD statement, or in the PTF cover letter, contains a description of all the special processing necessary to install this PTF.

Testing the new service level

After having installed the new service, you should perform two operations:

1. Create a backup of the updated data sets, including the SMP/E data sets affected. This ensures that if something happens to the data sets during the next phase, you do not have to repeat all the processing done in previous steps.
2. Perform some testing before putting the service into production. This testing should include some of the following:

- Run selected product IVP jobs.
- Run test job streams, if your installation has them.
- Attempt an IPL.

Only after verifying the service on a noncritical test system should you put that service into production. The test phase should not be considered complete until you have run the service in production mode for some period (as determined by the requirements for your installation). If no errors are found, you are ready to proceed to the next step: updating your distribution libraries.

Updating the distribution libraries: The ACCEPT process

The last major phase of installing preventive service is updating the distribution libraries with the SMP/E ACCEPT command. This is a critical step. Once the service is accepted, there is no SMP/E method to remove it from either the target or distribution libraries.

When you are ready to update your distribution libraries, you have the same set of considerations and SMP/E support as described under "Updating the target libraries: The APPLY process" on page 126, and the same three-phase operation:

1. **Receive additional HOLDDATA.** Before starting the ACCEPT, you should obtain any additional HOLDDATA for the service level you are installing. This step is required if:
   - You did not obtain the additional HOLDDATA from the IBM Support Center during the staging phase.
   - There has been a delay between the RECEIVE staging phase and the ACCEPT DLIB update phase.
2. **Note:**
   a. If there is a significant time between the APPLY and ACCEPT, additional problems may have been reported for which ++HOLD statements have been created. If this new data is not obtained, SMP/E may install PE PTFs into the distribution libraries.
   b. You may want to run the REPORT ERRSYSMODS command to see whether any SYSMODs that are applied are now in error. For more information, see Chapter 16, "Identifying installed SYSMODs affected by error holds: The REPORT ERRSYSMODS command," on page 165.
3. **Run the ACCEPT CHECK job.** The second job is a nonupdating mode of ACCEPT, referred to as the ACCEPT CHECK run. Its purpose is to help resolve any problems that prevent the ACCEPT itself from successfully completing processing.
4. **Run the ACCEPT update.** The third job is the updating mode of ACCEPT, in which the preventive service is installed into the distribution libraries.

Note: Special processing may be required during the ACCEPT process. PTFs requiring this processing should be handled in the same manner as during the APPLY process.

Checking the update (ACCEPT CHECK)

The ACCEPT CHECK job provides the same function for the distribution libraries that the APPLY CHECK job provided for the target libraries. See "Checking the update (APPLY CHECK)" on page 127.

Use the SMP/E dialogs or the following sample job to do an ACCEPT CHECK for preventive service. This example is an ACCEPT CHECK job for PTFs in service level 0703:

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPTCK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1)          /* Set to DLIB zone.        */.
ACCEPT   SOURCEID(PUT0703)   /* Accept this service level*/
         GROUPEXTEND(        /*  Include requisite PTFs  */
                NOAPARS      /*  Don't include APARs or  */
                NOUSERMODS)  /*  USERMODs                */
         CHECK               /*  but do not update libs. */
         SELECT(sysmod-id,…) /* Select additional     */
                             /* service if required.     */
         BYPASS(HOLDCLASS(ERREL,UCLREL)
```

```
                HOLDSYSTEM) .
/*
```

Note: This example can be used for PTFs from either a CBPDO or an ESO.

If you want to install preventive service only on selected functional areas of the system, you can also specify the FORFMID operand on the ACCEPT command, specifying either specific function identifiers (FMIDs) or the name of one or more FMIDSETs.

Researching the ACCEPT CHECK reports

The ACCEPT CHECK reports should be researched in the same manner as the APPLY CHECK reports (see “Researching the APPLY CHECK reports” on page 128).

Getting additional SYSMODs

The procedure for getting additional SYSMODs or APAR fixes from those PTFs being accepted is the same as that followed during the APPLY process (see “Getting additional SYSMODs” on page 129).

If you obtain additional SYSMODs during the ACCEPT phase, you should process these through the APPLY phase before accepting them.

Updating the distribution library (ACCEPT)

The ACCEPT command causes SMP/E to update the distribution libraries. You can use the SMP/E dialogs or the following sample job to accept the preventive service:

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPT   EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(DLIB1)         /* Set to DLIB zone.        */.
ACCEPT   SOURCEID(PUT0703)  /* Accept this service level*/
         GROUPEXTEND(       /*  Include requisite PTFs  */
                NOAPARS     /*  Don't include APARs or  */
                NOUSERMODS) /*  USERMODs                */
                            /*  No check this time.     */
         SELECT(sysmod-id,…) /* Select additional     */
                            /* service if required.     */
         BYPASS(HOLDCLASS(ERREL,UCLREL)
                HOLDSYSTEM) .
/*
```

Note:

1. If you have obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand or, if **all** applicable APARs and USERMODs are to be installed, specify the APARS and USERMODS operands.
2. If you want to install preventive service only on selected functional areas of the system, you can also specify the FORFMID operand on the ACCEPT command, specifying either specific function identifiers (FMIDs) or the name of one or more FMIDSETs.

Installing PTFs that need special processing

During the ACCEPT process, the considerations for special processing are the same as for the APPLY process (see “Installing PTFs that need special processing” on page 130).

Chapter 9. Installing corrective service

This section describes the steps for installing corrective service. It discusses these topics:

- An introduction to corrective service
- Building and checking a corrective service fix
- Preparing your system
- Staging the SYSMODs with the RECEIVE command
- Requesting corrective service with the RECEIVE ORDER command
- Updating the target libraries with the APPLY command
- Testing the corrective service
- Updating the distribution libraries with the ACCEPT command

Introduction

Corrective service is the process of installing a SYSMOD to resolve a specific problem in your system. The problem has usually been brought to your attention because the system has not functioned as expected (for example, an abend has occurred, or jobs are not running as expected).

The first task is to investigate the problem, so that the failing component and module can be identified. *z/OS SMP/E Messages, Codes, and Diagnosis* provides helpful information about diagnosing and handling SMP/E problems. This can be done in conjunction with the IBM Support Center. SMP/E can help you work with the IBM Support Center to isolate and obtain a fix for the problem. Useful information includes:

- The function and service level of the module involved
- The service level of your system—that is, the specific SYSMODs that have been installed
- Any USERMODs involved
- The affected load modules

After determining the cause of the error, you want a fix for the problem. There are several possibilities:

1. The problem has already been reported, and a PTF has been created to fix the module. If you do not already have a copy of the PTF received, you can use ShopzSeries or the RECEIVE ORDER command to order the PTF from the Automated Service Delivery server.
2. The PTF identified by the Support Center may have been received but not yet applied. Use the LIST command or the SMP/E Query dialog to check the status of the PTF.
3. The problem has been previously reported. No PTF has been created, but an APAR fix is available either to fix or to bypass the problem.
4. The problem is a new one, and no fix is available. In this case, you have to work with the IBM Support Center to construct a fix for your system.

No matter where you obtain the fix, the installation of that fix is said to be in corrective mode.

Note: You can use either the SMP/E dialogs or JCL jobs to build, receive, apply, and accept corrective service. The basic steps to follow are the same. If you have access to the SMP/E dialogs, you should use them. Otherwise, you can use the steps described in this chapter as examples.

Building or checking the fix

If the fix is a PTF, you can assume that the construction of that fix is accurate and the material in this section is not applicable.

© Copyright IBM Corp. 1986, 2026

If the fix obtained is not a PTF, you should make sure it was constructed accurately. This is true even if the fix obtained from the IBM Support Center is already in SMP/E format (that is, you received a ++APAR type SYSMOD).

If you have to build the fix yourself, see Standard Packaging Rules for z/OS-based Products (publibz.boulder.ibm.com/epubs/pdf/gimpkg80.pdf) for rules for constructing APAR SYSMODs and *z/OS SMP/E Reference* for details on MCS syntax. (To get started, you might find Chapter 19, “Building a user modification,” on page 171 helpful.) The general format of all ++APAR fixes is:

```
++APAR(xxxxxxx)             /* APAR identifier          */.
++VER (srel)                /* System identifier        */
      FMID(aaaaaaa)         /* Functional area          */
      PRE (                 /* PRE some SYSMODs.        */
           bbbbbbb          /* PRE RMID of element      */
           ccccccc cccccc   /*  and any UMIDs present.  */
           ccccccc cccccc   /*                          */
          )                 /*                          */
      SUP (                 /* SUP some SYSMODs.        */
           ddddddd dddddd   /* Fixes incorporated into  */
           ddddddd dddddd   /*  this fix                */
          )                 /*                          */.
++ZAP (modname)             /*                          */
      DISTLIB(eeeeeeee)     /* DLIB name                */.
… Some superzap cards here
   or
++MACUPD (macname)          /*                          */
      DISTLIB(eeeeeeee)     /* DLIB name                */.
… Some IEBUPDTE cards here
   or
++SRCUPD (srcname)          /*                          */
      DISTLIB(eeeeeeee)     /* DLIB name                */.
… Some IEBUPDTE cards here
```

You should perform the following checks to ensure the accuracy of the fix:

1. Make sure the value *xxxxxxx* in the ++APAR statement is equal to the APAR number associated with the problem you are fixing.
2. Make sure the system release value (SREL) in the ++VER statement matches one of those defined as an SREL subentry in the TARGETZONE entry for your target zone.
3. Make sure the FMID value *aaaaaaa* in the ++VER statement matches the FMID value in the appropriate element entry in your target zone. You can determine this value by listing the appropriate entry.
4. If the element entry in your target zone has an RMID value different from its FMID value, make sure it is a prerequisite of the APAR fix (that is, make sure the *bbbbbbb* value is accurate). If the RMID and FMID values are equal, the *bbbbbbb* value need not be specified.
5. If the element entry in your target zone has any UMID values, you should first make sure the fix itself was constructed so that it works correctly in that environment.

   You should then make sure each of the UMID values is specified in the PRE operand in place of the *ccccccc* values shown in the example. This is not absolutely required, but if it is not done, SMP/E issues warning messages during installation indicating that these SYSMODs may have an intersection with the one you are installing, and therefore may be regressed. Putting the UMID values in the PRE list suppresses these messages.
6. If this SYSMOD is to fix multiple problems, each of the additional APARs that are being fixed should be specified in the SUP operand (*ddddddd* values in the example).
7. Make sure the name in the ++ZAP, ++MACUPD, or ++SRCUPD statement is correct.
8. Make sure the value *eeeeeeee* specified in the DISTLIB operand matches the DISTLIB value in the target zone entry.

   Note: The DISTLIB value is optional, but it is a good idea to specify it to make sure there is no mistake about which element you are dealing with.

Once you have made sure the SYSMOD is accurate, you are ready to start the actual installation process.

Preparing your system

Corrective service is different from the installation of a new function or preventive service.

- It usually affects a limited area of the system.
- It is usually done because a severe problem is affecting the system.
- There is a need for an immediate fix.
- The fix usually takes little time to install.

Thus, there is usually no need or time to prepare the system by backing up packs, compressing libraries, and so on. If possible, it is a good idea to have a backup system available in case a problem does occur.

Staging the SYSMODs: the RECEIVE process

After verifying that the corrective SYSMOD is syntactically correct and specifies the proper set of functions and PTFs, receive that SYSMOD (APAR or PTF) into the SMP/E database so you can install it into the target libraries.

Because corrective service requires a very small number of SYSMODs—often only one—the job of receiving it is simple. You can use the SMP/E dialogs or the following sample job:

```
//RECEIVE  JOB 'accounting info',MSGLEVEL=(1,1)
//RECEIVE  EXEC SMPPROC
//SMPPTFIN DD …points to input with SYSMOD
//*              If you put the SYSMOD in a data set
//*                 refer to that data set.
//*              If the SYSMOD is in card format
//*                 use "DD *" followed by the cards.
//SMPCNTL  DD *
SET      BDY(GLOBAL)        /* Set to global zone.      */.
RECEIVE  SELECT(            /* Receive selected SYSMODs.*/
                xxxxxxx     /* Specify SYSMOD number.   */
               )            /*                          */
         SYSMODS            /* Only process SMPPTFIN -
                               do not look at SMPHOLD.  */.
/*
```

Note: No source ID was assigned. This is because the SYSMOD is installed selectively in the APPLY step. If you want to assign a common value or tag the SYSMOD with some sort of identifier (such as programmer initials), you can use the SOURCEID operand.

If the input data set contains only the SYSMODs you are installing for this corrective service problem, you can omit the SELECT operand. SMP/E then attempts to process all the SYSMODs in the SMPPTFIN input data set.

Generating a service request using the RECEIVE ORDER Command

You can order specific PTFs by name and you can order PTFs that resolve specific APARs. The package that results from such an order is tailored to your SMP/E environment and contains the PTFs you requested, plus any requisite PTFs if those requisites are not already present in your environment.

You can use the SMP/E dialogs or the following sample job:

```
//jobname JOB ...
//RECEIVE EXEC PGM=GIMSMP
//SMPCSI DD DSN=SMPE.GLOBAL.CSI,DISP=SHR
//SMPNTS DD PATH='/u/smpe/smpnts/',PATHDISP=KEEP
//SMPOUT DD SYSOUT=*
//SMPRPT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
  SET BOUNDARY(GLOBAL).
  RECEIVE SYSMODS HOLDDATA
          ORDER(                          /* Place an order for service */
            ORDERSERVER(ORDRSRVR)
            CONTENT(
              PTFS(UQ12345,UQ98765)       /* Get these PTFs, and any..  */
```

```
                    )                          /* ..requisites..             */
               FORTGTZONES(ZOS14)              /* ..for this target zone     */
                 ).
   /*
   //ORDRSRVR DD *
     <ORDERSERVER
       url="https://eccgw01.boulder.ibm.com/services/projects/ecc/ws/"
       keyring="MRWKYRNG"
       certificate="SMPE Client Certificate">
     </ORDERSERVER>
/*
```

Note: IBM Automated Delivery Request server can be found at `https://eccgw02.rochester.ibm.com/services/projects/ecc/ws`.

Updating the target libraries: the APPLY process

After receiving the corrective service, you are ready to install it into the target libraries. Do not attempt to install the SYSMODs without first verifying them. If you have already done all the proper checking, the SYSMODs should be installed correctly. However, if you overlooked something, the direct installation might cause unexpected results.

Checking the update (APPLY CHECK)

The purpose of this job is to verify that the SYSMODs can be installed correctly and that you understand what libraries and load modules in the system are affected. You can use the SMP/E dialogs or the following sample job to do an APPLY CHECK for corrective service:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SELECT(            /* Install selected SYSMOD. */
                xxxxxxxx    /* Specify SYSMOD name here.*/
               )            /*                          */
         CHECK              /* In check mode only.      */.
```

Researching the APPLY CHECK reports

Review the reports from the check, looking for the following types of information:

- Were any error messages produced? If so, determine the cause and fix the problem.
- Will any SYSMODs be regressed? If so, determine how to resolve the problems.
- Are any other areas of the system affected?

Using HOLDDATA to assist in identifying fixes

If the SYSMOD you are installing is a PTF (obtained from a CBPDO, an ESO, or directly from the IBM Support Center), SMP/E might have some HOLDDATA stored applying to that PTF. If so, the reports will indicate all the reason IDs preventing PTF installation. Use these reason IDs to determine what the errors are. For example:

1. List the SYSMOD and MCS entries for the PTF.
2. Look at the ++HOLD statements that are listed.
3. Use the COMMENT field in the ++HOLD statement to determine the cause of the error. If the COMMENT field is not present or does not describe the problem adequately, ask the IBM Support Center for further information.
4. Determine whether the error in the PTF is critical enough to stop it from being installed. Remember that you are trying to fix an existing problem; you may decide to install the PTF to fix that problem because the exposure is minimal.

5. If necessary, contact the IBM Support Center to obtain a corrective fix for the PTF. If, in the preceding step, you decided that the PTF should be installed immediately to fix your problem, you should perform this step at some later date.

Getting additional SYSMODs

If the research of the APPLY CHECK reports discloses that additional SYSMODs are required, these should be obtained in the same manner as the original corrective SYSMOD.

Updating the target library (APPLY)

Once the APPLY CHECK has run to your satisfaction, you are ready to install the fix using the SMP/E dialogs or the following sample job to apply the corrective service:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLY    EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SELECT(            /* Install selected SYSMOD. */
               xxxxxxxx     /* Specify SYSMOD name here.*/
              )             /*                          */
                            /* Note no check operand.   */.
```

Testing the corrective service

The testing needed after you install a corrective fix depends on the type of problem you encountered. It can range from no testing to running the job in which the error was detected.

Updating the distribution libraries: the ACCEPT process

After you install corrective service into the target libraries, you must decide whether you want to update the distribution libraries. Base this decision on the products involved and on your processing requirements.

The following is a consideration for not accepting corrective service:

> Corrective service has not been tested and, therefore, may be found to be in error at some later date. After you accept the SYSMODs, there is no RESTORE capability.

The following are some of the considerations for accepting corrective service:

- If you do not accept the SYSMOD and you perform a system generation, all that service is lost and must be reinstalled.
- If you must restore a SYSMOD, the work required increases with the number of SYSMODs that have been applied but not accepted. All intersecting SYSMODs must be restored, and then all but the desired SYSMOD must be reapplied. This is especially true for source-modified products.

The following sections describe the steps to follow, assuming that you are going to accept corrective service.

Checking the update (ACCEPT CHECK)

The ACCEPT CHECK job provides the same function for the distribution libraries that the APPLY CHECK job provided for the target libraries. It is important because the function and service level of the modules in the distribution libraries may be different from that in the target libraries. You can use the SMP/E dialogs or the following sample job to do an ACCEPT CHECK for corrective service:

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPTCK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1)         /* Set to DLIB zone.        */.
ACCEPT   SELECT(            /* Install selected SYSMOD. */
               xxxxxxxx     /* Specify SYSMOD name here.*/
```

```
                )            /*                          */
          CHECK              /* In check mode only.      */.
```

Researching the ACCEPT CHECK reports

You should research the ACCEPT CHECK reports in the same way as for the APPLY process (see “Researching the APPLY CHECK reports” on page 136).

Using HOLDDATA to assist in identifying fixes

If SMP/E reported any exception SYSMOD data during the APPLY CHECK process, you should expect to see the same information during the ACCEPT CHECK process. If you have processed any HOLDDATA between the APPLY and ACCEPT, additional information may be reported. This information should be handled in the same manner as the APPLY information.

Getting additional SYSMODs

If additional SYSMODs are required to ACCEPT the corrective service, you should obtain them in the same manner as the original corrective service SYSMOD.

Note: If you obtain additional SYSMODs, make sure you process them through the APPLY and test phases before accepting them.

Updating the distribution library (ACCEPT)

Once the ACCEPT CHECK runs to your satisfaction, you are ready to accept the fix. Use the SMP/E dialogs or the following sample job to accept the corrective service:

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPT   EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1)         /* Set to DLIB zone.        */.
ACCEPT   SELECT(            /* Install selected SYSMOD. */
                xxxxxxxx    /* Specify SYSMOD name here.*/
               )            /*                          */
                            /* Note no check operand.   */.
```

Chapter 10. Installing a user modification

This section describes the steps for installing a user modification (USERMOD). After an introduction to USERMODs, it describes the following processes:

- Preparing your system
- Staging the USERMOD with the RECEIVE command
- Updating the target libraries with the APPLY command
- Testing the USERMOD
- Updating the distribution libraries with the ACCEPT command

Introduction

A USERMOD is a SYSMOD used to make a modification to some IBM-supplied software element (module, macro, source, or data element) to implement a new function or to provide a hook into a user program that provides that function.

A USERMOD should not be confused with an APAR SYSMOD (corrective fix), even if you built the initial version of that fix to fix a problem immediately. For a description of how to construct USERMODs, see Chapter 19, “Building a user modification,” on page 171. For details on the syntax of the MCS statements used in constructing USERMODs, see *z/OS SMP/E Reference*. The Standard Packaging Rules for z/OS-based Products (publibz.boulder.ibm.com/epubs/pdf/gimpkg80.pdf) contains additional information about when a USERMOD should be used. The rest of this chapter assumes that you have properly constructed the USERMOD and are now ready to install it.

Note: You can use either the SMP/E dialogs or JCL jobs to receive, apply, and accept USERMODs. The basic steps to follow are the same. If you have access to the SMP/E dialogs, you should use them. Otherwise, you can use the steps described in this chapter as examples.

Preparing your system

You must determine the amount of system preparation necessary for your USERMOD. If it is extensive and affects critical components of the system, you should perform the same tasks as defined under “Preparing your system” on page 118 or “Preparing your system” on page 126. If it is a minor change, affecting few modules and not critical to the operation of the system, no preparation is needed.

Staging the SYSMODs: The RECEIVE process

Because a USERMOD is generally processed as a single SYSMOD, processing is similar to that for corrective service; that is, it is received by use of the SELECT option. Use the SMP/E dialogs or the following sample job to receive the USERMOD:

```
//RECEIVE  JOB 'accounting info',MSGLEVEL=(1,1)
//RECEIVE  EXEC SMPPROC
//SMPPTFIN DD …points to input with your USERMOD
//*              If you put the USERMOD in a data set
//*                 refer to that data set.
//*              If the USERMOD is in card format
//*                 use "DD *" followed by the cards.
//*              Create your data set in LRECL=80,
//*                 FB format.
//SMPCNTL  DD *
SET      BDY(GLOBAL)        /* Set to global zone.      */.
RECEIVE  SELECT(            /* Receive selected SYSMODs.*/
                xxxxxxxx    /* Specify USERMOD number.  */
               )            /*                          */
         SYSMODS            /* Only process SMPPTFIN -
                               do not look at SMPHOLD.  */.
/*
```

© Copyright IBM Corp. 1986, 2026

Note: No source ID was assigned, because the SYSMOD is installed selectively in the APPLY step. If you want to assign a common value to all the USERMODs or tag each of them with some sort of identifier (such as programmer initials), you can use the SOURCEID operand.

If the input data set contains only USERMODs that you want to receive now, you can omit the SELECT operand. SMP/E then attempts to process all SYSMODs in the SMPPTFIN input data set.

Updating the target libraries: The APPLY process

After receiving the USERMOD, you are ready to install it into the target libraries. You may be tempted to install the SYSMODs without first performing the verification pass. If you have constructed your USERMOD correctly, it should install correctly. However, if you have overlooked something, the direct installation may cause unexpected results. Thus, it is advisable to perform the verification pass.

Checking the update (APPLY CHECK)

The purpose of this job is to verify that the SYSMODs are installed correctly and that you understand which libraries and load modules in the system are affected. Use the SMP/E dialogs or the following sample job to do an APPLY CHECK for the USERMOD:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SELECT(            /* Install selected SYSMOD. */
                xxxxxxxx    /* Specify SYSMOD name here.*/
               )            /*                          */
         CHECK              /* In check mode only.      */.
```

Note: At times, it may be necessary to reinstall a USERMOD; for example, after the installation of a PTF. If you are reinstalling it, the APPLY REDO operand is necessary. You may also have to specify one of the BYPASS operands; that depends on the relationship between your USERMOD and the PTF that was installed.

Researching the APPLY CHECK reports

Review the reports from the APPLY CHECK process, looking at the following types of information:

- Were any error messages produced? If so, determine the cause and fix the problem.

  A common error here is that the FMID specified on the ++VER modification control statement did not match the FMID value in the element entry; therefore, SMP/E does not select the element to be installed. This condition does not stop the USERMOD from being installed. However, messages are issued to say which elements were not selected.
- Will any SYSMODs be regressed? If so, determine how to resolve the problems.

Updating the target library (APPLY)

Once the APPLY CHECK runs to your satisfaction, you are ready to install the USERMOD. Use the SMP/E dialogs or the following sample job to apply it:

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLY    EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1)          /* Set to target zone.      */.
APPLY    SELECT(            /* Install selected SYSMOD. */
                xxxxxxxx    /* Specify SYSMOD name here.*/
               )            /*                          */
                            /* Note no check operand.   */.
```

Testing the USERMOD

The amount of testing needed after the installation of a USERMOD depends on the changes you are making. You may want to review the recommendations found under "Testing the new function" on page 122 and "Testing the new service level" on page 130.

When originally constructing your USERMOD, you may want to provide a document similar to a program directory, containing some of the following information:

- The elements affected.
- The areas within each element.
- Externals of the change.
- An IVP job that can be used to ensure that the USERMOD is working correctly. This can be used after subsequent preventive service is applied or if the USERMOD must be reinstalled because a new release of the IBM product is installed.

This information may be helpful to the next system programmer responsible for installing and maintaining your USERMOD.

Updating the distribution libraries: The ACCEPT process

Once you have installed the USERMOD into the target libraries, you must decide whether you want to update the distribution libraries. While this decision is up to you, IBM generally does **not** recommend the accepting USERMODs, because if a problem is encountered in the modified modules, you may be asked to re-create the problem using an unmodified version. If you have accepted the USERMOD, you cannot create an unmodified version of the module unless you are also maintaining a separate set of distribution libraries without the USERMODs.

Note: You can use ++HOLD statements to prevent USERMODs from being accepted. For each USERMOD that you want to keep from being accepted, follow these steps after applying the USERMOD:

1. Create a ++HOLD statement with a user reason ID that you plan to use only for USERMODs that are not supposed to be accepted. Here is an example:

   ```
   ++HOLD(usermod) USER
   REASON(NOUSERM)
   COMMENT(do not accept this usermod).
   ```

2. Run the SMP/E RECEIVE command to read in the ++HOLD statement. Use the SMPHOLD DD statement to point to the file or data set containing the ++HOLD statement.

Because of the user hold, this USERMOD can be accepted only if you bypass the specific user reason ID. The SYSMOD will not be automatically accepted if you specify USERMOD or the specific SYSMOD ID on the ACCEPT command without bypassing the user reason ID.

Be aware that if you receive the ++HOLD statement before applying the USERMOD, you must bypass the user hold reason ID in order to apply the USERMOD.

Chapter 11. Managing exception SYSMODs

This section explains how SMP/E manages SYSMODs that require special processing. It discusses these topics:

- An introduction to exception SYSMODs
- What SMP/E does with HOLDDATA
- Sources of HOLDDATA
- Steps for processing the data

Introduction

Most SYSMODs you receive from IBM can be installed without additional considerations; you can simply receive, apply, and then accept them. For some SYSMODs, however, this is not possible. Examples of such SYSMODs are:

- SYSMODs that were sent out to correct a problem but that either did not fix the problem or introduced a new problem. These are called *PTFs in error*, or PE PTFs.
- SYSMODs that require special installation processing, such as a fix that must be concurrently installed on all processors in a network.
- SYSMODs that introduce changes into the system that you should be made aware of, such as changes to operator messages or critical documentation changes.

In SMP/E terms, these SYSMODs are called *exception SYSMODs*. SMP/E supplies a function to automate the management of exception SYSMODs. SMP/E supports three categories of exception SYSMODs:

Error
PTFs in error (PE PTFs).

System
SYSMODs identified by IBM as requiring special processing or notification.

User
Any SYSMODs that you identify as requiring special processing.

Two MCSs are used to manage exception SYSMODs:

- ++HOLD puts a SYSMOD into exception status.
- ++RELEASE removes a PE PTF from exception status when it has been determined that the PTF was held erroneously.

++HOLD statements for system holds are usually built as part of the held PTF. ++RELEASE statements and ++HOLD statements for error or user holds must be in SMPHOLD. ++HOLD and ++RELEASE statements that are provided by SMPHOLD (external holds) identify the following:

- The SYSMOD ID of the exception SYSMOD (that is, the SYSMOD being held).
- The exception SYSMOD category.
- The FMID to which that ++HOLD applies.
- The reason the SYSMOD is being put into or was in exception status. This is a 1- to 7-character alphanumeric string called the *reason ID*.
  - For error-category exception SYSMODs, SMP/E expects the reason ID to be the SYSMOD ID of the APAR reporting the problem.
  - For system-category exception SYSMODs, SMP/E expects the reason ID to be a short description of the action required.

© Copyright IBM Corp. 1986, 2026

  - For user-category exception SYSMODs, SMP/E makes no assumption about what the reason ID represents.

  For more information about reason IDs, see *z/OS SMP/E Reference*.

- Text describing why the SYSMOD is being put into exception status. This field is only for ++HOLD statements.
- An alternative way to release the exception SYSMOD. This field is only for ++HOLD statements.

  Every ++HOLD statement specifies a HOLD category of ERROR, SYSTEM, or USER. In addition to one of these categories, a ++HOLD statement may include a HOLD CLASS, which is an alternative way to release a held SYSMOD. For example, an exception SYSMOD may fix a problem more severe than the problem it introduces. The ++HOLD statement for that SYSMOD would have an ERROR reason ID that matches an APAR ID and a CLASS of ERREL.

++HOLD statements provided within a SYSMOD identify the same information. However, even though these internal holds are effective against the containing SYSMOD, the SYSMOD ID specified on the hold may be different from that of the containing SYSMOD, as long as the SYSMOD ID specified on the hold is superseded by the containing SYSMOD.

SMP/E then manages exception SYSMODs by actually managing the resolution of the problems described by the reason ID specified on the ++HOLD statement.

Subsequent topics describe how SMP/E uses HOLDDATA during the installation of a SYSMOD, where the exception SYSMOD statements come from, and how to process them. The chapters on the RECEIVE command, the APPLY command, and the ACCEPT command in *z/OS SMP/E Commands* contain a much more detailed explanation of the material covered here.

What SMP/E does with the HOLDDATA

This section describes what SMP/E does with the HOLDDATA when processing the various commands associated with installing and removing SYSMODs.

Note: You must provide SMP/E with the most current HOLDDATA possible to get the most benefit from this support.

Initial entry into staging data sets: RECEIVE

The RECEIVE command tells SMP/E to take the HOLDDATA from the input data set on which it was delivered and store it in the SMP/E database.

The two operands that control input processes are:

- SYSMOD, which tells SMP/E to process the SYSMODs from the data set specified by the SMPPTFIN DD statement.
- HOLDDATA, which tells SMP/E to process the HOLDDATA (++HOLD and ++RELEASE statements) from the data set or file in a UNIX file system specified by the SMPHOLD DD statement.

You can specify one or both operands on the RECEIVE command. If neither operand is specified, SMP/E attempts to receive the SYSMODs from SMPPTFIN, and HOLDDATA from SMPHOLD.

When receiving a SYSMOD, SMP/E creates two entries:

1. An MCS entry is created on the SMPPTS. This entry is a copy (possibly compacted) of the SYSMOD as it appeared in the SMPPTFIN data set.
2. A SYSMOD entry is created in the global zone. This entry contains information that describes the installation requirements and element content of the SYSMOD.

When receiving the HOLDDATA, SMP/E also creates (or modifies) two entries:

1. A HOLDDATA entry is created (or modified) in the global zone. This entry is an exact copy of the ++HOLD statements as they appeared in SMPHOLD. The name of the entry is the ID of the SYSMOD affected by this ++HOLD statement. The HOLDDATA entry for a single SYSMOD can contain multiple ++HOLD statements.

Note: When a ++RELEASE statement is processed, SMP/E removes the corresponding ++HOLD statement from the HOLDDATA entry. When all ++HOLD statements are removed, the HOLDDATA entry is automatically deleted.

2. A SYSMOD entry is created (or modified) in the global zone. This entry contains information that describes the exception SYSMOD conditions.

   For each ++HOLD statement processed, SMP/E updates the global zone SYSMOD entry to add a HOLD reason ID subentry. There are three types of HOLD reason ID subentries, HOLDERROR, HOLDSYSTEM, and HOLDUSER, corresponding to the three categories of exception SYSMODs.

   Note: When a ++RELEASE statement is processed, SMP/E removes the corresponding reason ID from the global zone SYSMOD entry. Do not use the ++RELEASE statement to install a SYSMOD with an unresolved reason ID. Use the appropriate BYPASS operand instead.

Updating target libraries: APPLY

When SMP/E applies a SYSMOD, SMP/E checks to see if that SYSMOD is currently in exception SYSMOD status by seeing if there are any HOLD reason ID subentries in the global zone SYSMOD entry. If so, SMP/E makes sure each reason ID is resolved before allowing the SYSMOD to be installed.

For an error reason ID to be resolved, **at least one** of the following conditions must be met:

- The reason ID must be superseded by another SYSMOD being installed.
- The reason ID must already exist as a SYSMOD entry in the target zone.
- You must specify BYPASS(HOLDERROR) on the APPLY command to show that you are aware that an unresolved exception SYSMOD is being installed.
- If there is a HOLD CLASS associated with the reason ID, you can specify BYPASS(HOLDCLASS) on APPLY to indicate that you are using an alternative way to resolve the reason ID.

  For more information about the BYPASS operand, see the APPLY command section in i *z/OS SMP/E Commands*.

Internal holds are considered resolved if any of the following conditions are met:

- The SYSMOD ID specified on the ++HOLD defining the exception is found as a SYSMOD entry in the target zone
- The SYSMOD ID specified on the ++HOLD defining the exception is being superseded by a SYSMOD being applied concurrently
- The applicable BYPASS operand is specified.

You can resolve external system and user reason IDs by specifying BYPASS(HOLDSYSTEM) or BYPASS(HOLDUSER) on the APPLY command. If there is a HOLD CLASS associated with the reason ID, you can specify BYPASS(HOLDCLASS) on APPLY to indicate that you are using an alternative way to resolve the reason ID.

If you choose to resolve a reason ID by using the BYPASS operand, you must do any required processing at the appropriate time. Otherwise, errors related to the undone processing may occur, even though the reason ID was considered resolved.

Note: You may use the ++RELEASE statement for user category reason IDs if you want to unconditionally release the SYSMOD for all systems. Remember that, unlike BYPASS, ++RELEASE actually deletes the ++HOLD statement. If you plan to use the user category ++HOLD statement, see *z/OS SMP/E Reference* for more information about the naming conventions for reason IDs.

If all reason IDs are resolved, SMP/E allows the SYSMOD to be applied. If any remain unresolved, SMP/E prevents the SYSMOD, and any other SYSMODs dependent on this one, from being installed.

SMP/E leaves the reason IDs in the global zone SYSMOD entry when the SYSMOD is applied, so if the SYSMOD is applied on another system later, the same checking is done on that system. If the information had been deleted during the first APPLY, SMP/E would not recognize the problem when the SYSMOD

is applied to subsequent systems. Therefore, the ++RELEASE statement **should not be used** to install an exception SYSMOD with an unresolved reason ID. The appropriate BYPASS operand should be used instead.

In summary, SMP/E ensures that no known problems are introduced into your system by managing those problems at the level of the individual problem, rather than the SYSMOD level. It is, therefore, important that SMP/E have the most current information about exception SYSMODs. For more information about the importance of having current HOLDDATA and what you must do to provide that information to SMP/E, see "How to process HOLDDATA" on page 147.

Updating distribution libraries: ACCEPT

Exception SYSMOD processing is the same when accepting a SYSMOD as when applying one, except that the appropriate distribution zone is used to determine whether the fixes for the reason IDs have been installed.

Removing HOLDDATA from SMP/E data sets

There are various ways to remove HOLDDATA from SMP/E data sets.

During RESTORE processing

HOLDDATA is never deleted during RESTORE processing. The assumption is that you may later want to reapply the SYSMODs you are restoring.

With the REJECT command

If you are using the SMPPTS as a history file of all SYSMODs, you may eventually want to purge some of those SYSMODs. To do this, use the REJECT command. You can use the HOLDDATA operand to have SMP/E delete not only the global zone SYSMOD and SMPPTS MCS entries, but also any associated HOLDDATA.

Sources of HOLDDATA

Besides the data you create, these are the main sources of HOLDDATA provided by IBM:

- CBPDO packages
- IBM service web pages
- Automated Service Delivery server as a result of a RECEIVE ORDER command

CBPDO packages

One of the primary means of obtaining HOLDDATA is CBPDO packages. IBM custom-builds these tapes to provide the products and service you request, taking into consideration whether this is your first CBPDO order.

- If you select a particular service level, you get HOLDDATA for all service from that level to the current level.
- If you do not select a service level and this is your first CBPDO order, you get HOLDDATA for all the service shown on the order checklist.
- If you do not select a service level and you have ordered a CBPDO before, you get HOLDDATA for service following the last service level that was shipped in your previous CBPDO.

The HOLDDATA on a CBPDO package has been customized to your product set. That is, it contains only data applicable to PTFs for those products **within a given feature** for which you are licensed under a single customer number. However, it does not reflect the contents of any specific system within the establishment defined by that customer number.

The HOLDDATA on the CBPDO package should be processed immediately on receipt of the tape. You can use either the SMP/E dialogs or the RIMLIB jobs provided with the CBPDO package to receive the HOLDDATA. For more information, see the documentation that came with the CBPDO package.

Automated service delivery package

Another method of obtaining HOLDDATA is from an Automated Service Delivery server by issuing a RECEIVE ORDER command. Whether you use the RECEIVE ORDER command to obtain preventive service or corrective service, you automatically receive the last 2-years of Enhanced HOLDDATA for the entire z/OS software platform.

How to process HOLDDATA

The management of exception SYSMODs is an important part of SMP/E. SMP/E's ability to manage exception SYSMODs, however, is limited by the quality and timeliness of the HOLDDATA made available to it. To gain the full advantage of this function, you must understand how SMP/E expects the three HOLDDATA input sources to be used and the times during which SMP/E expects them to be used.

The following steps summarize the process for managing exception SYSMOD data:

1. Receive all new products as you get them, or use UCLIN to add the FMIDs of the new products to the global zone. This allows you to process exception SYSMODs for them. Then receive any associated HOLDDATA shipped with the product.
2. Receive HOLDDATA from subsequent CBPDO orders **in service level order**. Remember to do the following:
   - Receive all new products as you get them so you can process exception SYSMODs for them.
   - Receive HOLDDATA as soon as you get your CBPDO.
3. Before doing preventive service, list and review HOLDDATA for SYSTEM HOLDs and, if possible, handle the required special conditions. Then apply and accept these processed SYSMODs by specifying BYPASS(HOLDSYS) and listing the individual SYSMOD IDs on the SELECT operand. This step helps to make sure that all available service is installed when you do preventive service.

Processing HOLDDATA from a CBPDO package

When you receive a CBPDO package, the HOLDDATA it contains is based on the service level you selected and on whether this is your first CBPDO. This HOLDDATA pertains both to the PTFs actually on that tape and to PTFs shipped on previous tapes. Follow these steps to process the HOLDDATA:

1. Receive the HOLDDATA from the CBPDO package **as soon as you get the package**. Use the SMP/E dialogs or the RIMLIB job provided with the CBPDO package.

   By receiving the HOLDDATA as soon as possible, you make sure SMP/E has the most current information available. If you try to install any PTF in response to a problem on your system and that PTF is in error, SMP/E knows this and warns you so you can weigh the effect of installing the known problem against the effect of fixing the problem you have encountered.
2. Receive the SYSMODs from the CBPDO package **as soon as you get the package**. Use the SMP/E dialogs or the RIMLIB job provided with the CBPDO package.

   This makes sure that all available PTFs are ready to be installed. If you find a problem in your system and determine that a PTF must be installed in corrective mode, you have a better chance of having that PTF and all its requisites readily available on the SMPPTS.

Note: You can receive the SYSMODs and HOLDDATA separately or in the same job.

By following these procedures, you are essentially making a trade-off: system resources as increased DASD space for the SMPPTS against the time the system programmer would spend on searching for the service level with the required PTF and on fixing problems caused by installing PE PTFs.

One important part of this procedure is that the HOLDDATA on each CBPDO **must be received in chronological order**. SMP/E processes the ++HOLD and ++RELEASE statements in the order in which

they are encountered. Therefore, there can be an exposure if you receive the data out of sequence. For instance, the tapes may be set up so that one contains a ++HOLD for a PTF and a subsequent one contains a ++RELEASE for the same PTF. If the tapes are processed in the wrong order, the RELEASE statement is processed first, and then the HOLD statement. As a result, the PTF remains held.

Chapter 12. Creating cross-product, cross-zone load modules: The LINK MODULE command

This section discusses the LINK MODULE command, with an emphasis on when and how to use it.

When to use LINK MODULE

Products sometimes contain modules from other products. For example, a product may need to:

- Include another product's modules in its load modules. In this case, as long as the two products are in the same zone, SMP/E can automatically include the required modules in the load modules that need them (if the modules reside in the target library as single-CSECT load modules). SMP/E also tracks the inclusion of these cross-product modules in the load modules.
- Update another product's load module with one of its modules. In this case, as long as the two products are in the same zone, SMP/E can automatically relink the load module and include the supplied module. SMP/E also tracks the inclusion of the modules in the cross-product load module.

However, when such products reside in different zones, SMP/E cannot automatically perform the cross-zone link-edits. The LINK MODULE command can be used to perform these cross-zone link-edits as postinstallation steps within SMP/E control. The LINK MODULE command causes the required load modules in one zone to be linked with modules residing in another zone, and tracks this inclusion so that subsequent APPLY and RESTORE processing can automatically maintain the affected load modules.

Note:

1. When SMP/E processes the LINK MODULE command, it assumes that adding the modules to the load modules does not require any changes to the load module definition (that is, the linkage editor control statements or linkage editor attributes). If any such changes are needed, make them through JCLIN before using the LINK MODULE command.
2. For the LINK MODULE command, the SET BOUNDARY command must specify the target zone that contains the LMOD entries for the load modules to be link-edited.
3. There are times when the LINK MODULE command is **not** appropriate to use. That situation typically happens , for products that are written in a high-level language and, as a result, include modules from libraries (such as compiler libraries) owned by a different product. Your options for installing such a product depend on how the product was packaged.
   - SYSLIB DD statements are used in link-edit steps in order to implicitly include the necessary modules.

     In this case, when you install the product, the modules that are impoliticly included are automatically linked into the load modules. If the libraries containing those modules are updated, you can use the LINK LMODS command to rebuild the affected load modules. For more information, see the LINK LMODS chapter in *z/OS SMP/E Commands*.
   - No SYSLIB DD statements are used in link-edit steps in order to implicitly include the necessary modules. In this case, you must use postinstallation link-edit steps outside of SMP/E.

How to use LINK MODULE

Assume you have installed GDDM and CICS, and some of the GDDM modules must be linked into CICS load modules. GDDM resides in zone GDDTZN, and the zone controlling CICS is CICTZN. Because GDDM and CICS are controlled by different zones, SMP/E does not automatically link the GDDM modules into the CICS load modules when GDDM is installed. The LINK MODULE command can be used instead.

In this example, GDDM module ADMABCD needs to be linked into CICS load module DFHWXYZ. Module ADMABCD is installed in a target library as a single-CSECT load module when GDDM is installed.

© Copyright IBM Corp. 1986, 2026

Therefore, SMP/E can use the target library version of ADMABCD to update CICS load module DFHWXYZ. (If a module does not reside in a target library as a single-CSECT load module, SMP/E uses the related distribution zone copy of the module to update the load module.)

The following commands can be used to have SMP/E install and track the installation of GDDM module ADMABCD in the CICS load module:

```
SET   BDY(CICTZN)         /* Target zone for CICS.      */.
LINK  MODULE(ADMABCD)     /* Link module ADMABCD        */
      FROMZONE(GDDTZN)    /* residing in zone GDDTZN    */
      INTOLMOD(DFHWXYZ)   /* into load module DFHWXYZ.  */.
```

These commands cause GDDM module ADMABCD to be linked into CICS load module DFHWXYZ. SMP/E also adds cross-zone subentries to the affected entries:

- An XZLMOD subentry is added to the ADMABCD MOD entry in target zone GDDTZN so that if ADMABCD is updated, it can be automatically replaced in the CICS load module.

  Note: The CICS load module is automatically updated **only** if the XZLINK subentry was previously set to AUTOMATIC in the TZONE entry for zone CICTZN. Here is an example of the commands that can be used to do this:

  ```
  SET  BDY(CICTZN)        /* Target zone for CICS.    */.
  UCLIN.
  ADD  TZONE(CICTZN)      /* Update TZONE entry       */
       XZLINK(AUTOMATIC). /* to do automatic links.   */.
  ENDUCL.
  ```

- An XZMOD subentry is added to the CICS DFHWXYZ LMOD entry in target zone CICTZN to indicate that:
  - DFHWXYZ now contains ADMABCD.
  - Any updates for ADMABCD should be accepted **only** from zone GDDTZN.
- TIEDTO subentries are added to the TZONE entries for CICTZN and GDDTZN to indicate that there is a relationship between modules and load modules in these zones.

For more information about the LINK MODULE command and cross-zone updating during APPLY and RESTORE processing, see the LINK MODULE chapter in *z/OS SMP/E Commands*.

Chapter 13. Displaying the data managed by SMP/E: The LIST command

This section discusses the LIST command. After an introduction, it discusses these topics:

- Listing all the SMP/E data
- Listing by specific entry type
- Listing specific entries
- Listing by FMID or FMIDSET
- Listing to compare two zones

Introduction

The SMP/E database contains much information that is useful to you at certain times. For instance, when a problem is encountered in your system:

- You need to know the functional and service level of the module with the error.
- You may also want to know when that module was last changed: a recent change may have caused the problem.
- After reporting the problem, you can start working with the IBM Support Center to debug the problem. They may want to know the status of other specific PTFs: have you installed them; are they available on your system; is anything stopping them from being installed?
- After identifying the problem in the module, you may want to know whether any other parts of the system are affected by that module.

All of this information is available in the SMP/E database. You can obtain the information by using the SMP/E dialogs as you debug the problem. You can also use the SMP/E LIST command to create hardcopy listings of the information.

With the LIST command, you can display all entries of a specified type or specific entries. The following sections demonstrate the flexibility of the LIST command. For a complete description of all the LIST command operands, go to *z/OS SMP/E Commands*.

Listing all the SMP/E data

If you encounter a problem with your system, the SMP/E data describing that system can be very important in diagnosing the problem. This information can be obtained by using the SMP/E dialogs during the debugging process. However, if the system is not running, the information is not available unless you have periodically listed the SMP/E data for the system.

Therefore, it is advisable to list all the data for each of your systems and save the hardcopy listing. The data can be listed by individual zone. The following is an example of a job for listing all the entries in zone TGT01.

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT01)         /* Set to desired zone.     */.
LIST                        /* List all entries         */
         XREF               /* with XREF option to show
                               additional relationships
                               between entries.         */.
/*
```

© Copyright IBM Corp. 1986, 2026

Because the global zone contains data used to process each of the target zones and distribution zones, you may want to list that data more often. The following job lists all the data in the global zone, including the SYSMOD entries and the MCS entries:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(GLOBAL)        /* Set to global zone.      */.
LIST                        /* List all entries.        */.
/*
```

If you do not require a listing of the SYSMOD and MCS entries, you can use the LIST operands that enable you to list only specific entry types. For additional information, see “Listing by specific entry type” on page 152.

SMP/E also provides support to list all the entries for all the zones defined in the GLOBALZONE entry. This enables you to display all data for all systems with one SMP/E command. Here is an example of this option:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(GLOBAL)        /* Set to global zone.      */.
LIST                        /* List all entries         */
         ALLZONES           /* for all zones.           */.
/*
```

Note:

1. The ALLZONES operand should be used with caution; it may produce a large amount of output.
2. This function can also be qualified by other LIST operands to limit the entries listed from each zone. For example, the following job will list only the SYSMOD entries from all zones:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(GLOBAL)        /* Set to global zone.      */.
LIST     SYSMOD             /* List the SYSMOD entries  */
         ALLZONES           /* for all zones.           */.
/*
```

Listing by specific entry type

At times, you may need to have a listing of all entries of a certain type. For example:

- You may want to display all the DDDEF entries for a particular target zone or distribution zone.
- You may want to list all the OPTIONS and UTILITY entries in the global zone so you do not duplicate an existing entry.
- You may want to list all ORDER entries in the global zone.

SMP/E supports various operands on the LIST command that you can use to list all the entries for one or more entry types. The following job shows how to use the DDDEF, OPTIONS, UTILITY, and ORDER operands on the LIST command to do this:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT1)                      /* Set to target zone       */.
LIST     DDDEF                          /* List all DDDEF entries.  */
                                        /*                          */.
SET      BDY(DLIB1)                     /* Set to DLIB zone         */.
LIST     DDDEF                          /* List all DDDEF entries.  */
                                        /*                          */.
SET      BDY(GLOBAL)                    /* Set to global zone.      */.
LIST     OPTIONS                        /* List all options         */
         UTILITY                        /* and utility entries.     */
         ORDER(ORD000034,ORD000035)     /* List details for orders  */
                                        /* ORD000034 and ORD000035  */
```

```
.
/*
```

For a complete list of all the operands corresponding to the various entry types, see the LIST command section in *z/OS SMP/E Commands*.

Note: Not all entry types are valid for each zone type. For example, requesting a listing of the OPTIONS entries in a target zone results in an error, because OPTIONS entries exist only in the global zone. For a summary of the types of entries contained in each type of zone, see Table 3 on page 67 and Table 4 on page 67.

Sometimes, the various entry-type operands can be further qualified to process only a subset of all existing entries. The most common type for which this can be done is the SYSMOD entries. There are numerous operands to qualify SYSMOD entries. The LIST command section in *z/OS SMP/E Commands* describes all of them in detail.

One of the most common uses of this function is to determine the functions (or FMIDs) that have been installed. The following job can be used to list:

- The function SYSMODs installed on TGT1
- Those PTFs that have been applied to TGT1 but not yet accepted to DLIB1

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT1)          /* Set to target zone.      */.
LIST     SYSMOD             /* List the SYSMOD entries  */
         FUNCTIONS          /* for function SYSMODs.    */
                            /*                          */.
LIST     SYSMOD             /* List the SYSMOD entries  */
         PTFS               /* for PTF type SYSMODs     */
         NOACCEPT(DLIB1)    /* not yet accepted.        */
                            /*                          */.
/*
```

Listing specific entries

When you encounter a problem in your system and contact the IBM Support Center to resolve the problem, you may be asked to provide very specific information. For example:

- What is the service level of the module in which the problem was reported?
- Are there any USERMODs for the module?
- Do you have a specific PTF installed?

If you have a complete, current listing of all the entries in your system, you can get this information from that listing. You can also get it through the SMP/E dialogs while you are talking to the IBM Support Center.

SMP/E also provides additional LIST functions that you can use to display only specified entries. This is done by allowing you to specify a list of entry names (in parentheses) after each of the entry-type operands. For example, assume that you need to know the function and service level for modules GIMMPDRV and GIMMPIO and if SYSMOD UR12345 has been installed. The following job can be used:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT1)          /* Set to target zone.      */.
LIST     MOD(GIMMPDRV       /* List these two modules   */
             GIMMPIO)       /*                          */
         SYSMOD(UR12345)    /* and this SYSMOD.         */
                            /*                          */.
/*
```

You receive a listing of the required information for the two modules. If SYSMOD UR12345 was installed, it should be listed; otherwise, you receive a message saying that the entry was not found (meaning it has not been installed).

Another common use for this function is to list the cover letters for specific PTFs. The following job shows an example of a job for listing the cover letters for PTFs UR00001, UR00002, and UR00003:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(GLOBAL)        /* Set to global zone.      */.
LIST     MCS(               /* List cover letters       */
             UR00001        /* for these three PTFs.    */
             UR00002
             UR00003
            )
         .
/*
```

Listing by FMID or FMIDSET

Frequently, you deal with one area of the system at a time and would like to see all the information relating to that one area. You can use the FORFMID operand in conjunction with the various entry-type operands to limit the entries processed. Here is an example listing:

- All the entries associated with function HXY1100
- All the MAC entries associated with function HXZ2100
- All the SYSMOD and module entries associated with either function JXY1123 or JXY1124

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT1)          /* Set to target zone.      */.
LIST                        /* List all entries         */
         FORFMID(HXY1100)   /* for this FMID.           */
                            /*                          */.
LIST     MAC                /* List only the macros     */
         FORFMID(HXZ2100)   /* for this FMID.           */
                            /*                          */.
LIST     SYSMOD             /* List SYSMOD entries      */
         MOD                /* and MOD entries          */
         FORFMID(JXY1123    /* for this FMID            */
                 JXY1124)   /* or this FMID.            */
                            /*                          */.
/*
```

Note: The names within the FORFMID operand can be names of FMIDSET entries. In this case, SMP/E lists all the entries associated with any of the FMIDs defined in the FMIDSET entry.

Listing to compare two zones

If you have multiple zones, you may sometimes want to determine the functional and service differences between them. The LIST command provides you with this capability.

Note: You can also use the REPORT SYSMODS command to compare the SYSMOD content of two zones. Besides telling you which SYSMODs are installed in one zone but not in a second, REPORT SYSMODS also indicates which of the uninstalled SYSMODs are applicable to the second zone and generates commands you can run to install the SYSMODs in the second zone. For more information, see Chapter 18, “Comparing the SYSMODs installed in two zones: The REPORT SYSMODS command,” on page 169.

One possibility might be that you have two products at different service levels. The product at the lower service level works, and the product at the higher service level does not work. You might use LIST to compare the zones for the two systems and to determine what is causing the problem.

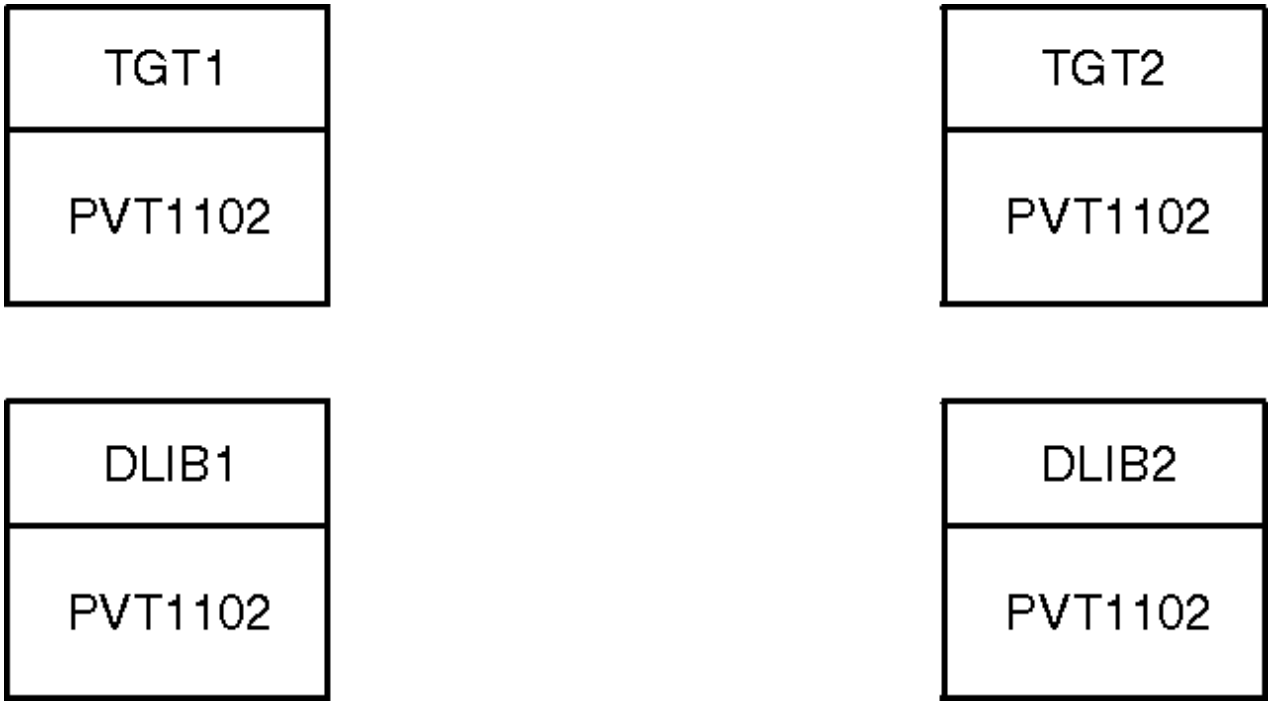
This example compares two target zones, TGT1 and TGT2. The commands in the following example perform the comparison for you:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(TGT1)          /* Set to target zone TGT1. */.
LIST     SYSMODS            /* List the SYSMODs         */
```

```
                                /* in zone TGT1             */
         NOAPPLY(TGT2)          /* that have not been       */
                                /* applied to TGT2.         */
                                /*                          */.
SET      BDY(TGT2)             /* Set to target zone TGT2. */.
LIST     SYSMODS               /* List the SYSMODs         */
                                /* in zone TGT2             */
         NOAPPLY(TGT1)          /* that have not been       */
                                /* applied to TGT1.         */
                                /*                          */.
/*
```

By comparing the two resulting listings, you can see the differences between the two zones.

In this second example, the same product is installed in different zones. You want to compare the service to make sure both copies of the product are at the same level. For example, assume that product PVT1102 is installed in two target zones and two distribution zones:



You want to make sure that PVT1102 is at the same service level in all the zones. To do this, you can use the LIST command and compare which SYSMODs are installed in which zones.

To compare the service levels of product PVT1102 in the two distribution zones, you can use the commands in the following example:

```
//LIST     JOB 'accounting info',MSGLEVEL=(1,1)
//LIST     EXEC SMPPROC
//SMPCNTL  DD  *
SET    BDY(DLIB1)          /* Set to DLIB1.        */.
LIST   SYSMOD              /* List SYSMODs         */
       FORFMID (PVT1102)   /* for PVT1102          */
       NOACCEPT (DLIB2)    /* in DLIB1, not DLIB2. */.
SET    BDY(DLIB2)          /* Set to DLIB2.        */.
LIST   SYSMOD              /* List SYSMODs         */
       FORFMID (PVT1102)   /* for PVT1102          */
       NOACCEPT (DLIB1)    /* in DLIB2, not DLIB1. */.
```

Similarly, to compare the service records for the target zone copies of PVT1102, you can use LIST with the NOAPPLY operand.

Summary

The LIST command enables you to:

- Compare two target zones using the NOAPPLY operand.

- Compare two distribution zones using the NOACCEPT operand in place of the NOAPPLY operand.
- Compare a target zone and a distribution zone using both the NOAPPLY and NOACCEPT operands.

This gives you the ability to compare all combinations of zone types, keeping in mind that the zone for the NOAPPLY operand must be a target zone, and that the zone for the NOACCEPT operand must be a distribution zone.

Chapter 14. Changing the data SMP/E manages: The UCLIN command

This section discusses the UCLIN command, with an emphasis on how and when to use it.

Introduction

The SMPCSI and associated data sets contain basically two types of information:

- Information added as a result of installing SYSMODs. Generally, this type is managed completely by SMP/E; that is, appropriate entries are added, changed, or deleted as SYSMODs are installed. You need not make any modification to the database to record this information. You may, however, need to make changes to do the following:
  - Record changes made outside SMP/E
  - Delete information no longer required
  - Recover from an SMP/E or system error
- Information added by you to control the installation of SYSMODs. You must manually add this information to the database before processing any SYSMODs. You may later need to modify the information to reflect new processing information.

The UCLIN command helps you to make these changes.

To use UCLIN effectively, you should have a detailed understanding of how it works and what it can do. *z/OS SMP/E Commands* has information about UCLIN. *z/OS SMP/E Reference* provides details about SMP/E data set entries. Read them before trying to run any UCLIN commands. The following sections describe, at a very high level, what UCLIN is.

When to use UCLIN

UCLIN is a very powerful function that must be used with **extreme caution**. You can use UCLIN to modify almost all the data in the SMP/E database. When you are modifying an entry, SMP/E makes sure the data within that one entry is consistent—that is, that the result could have occurred during normal SMP/E processing. However, no checking is done to make sure the resulting entry is consistent with other related entries in the database.

For example, you can use UCLIN to delete a UTILITY entry in the global zone without SMP/E detecting any error condition. However, if there is an OPTIONS entry within the global zone that refers to the deleted UTILITY entry, an error occurs when you attempt to use that OPTIONS entry. This is a very simple example of inconsistent data across entries that does not result in a serious error. UCLIN modifications to other entries, such as element, LMOD, or SYSMOD, may not be detected as error conditions during processing, but may cause incorrect processing, such as failing to link a module, updating the wrong library, or installing a SYSMOD that should not be installed.

In general, consider the following before making the UCLIN update:

1. Determine whether there is a better method of obtaining the same result. Table 11 on page 157 shows where to find more information about alternatives to UCLIN.

Table 11. Alternatives to UCLIN

To do this without UCLIN:	Look here for more information:
Change a common subentry in several DDDEF or UTILITY entries in the same zone	The ZONEEDIT command in *z/OS SMP/E Commands*

© Copyright IBM Corp. 1986, 2026

Table 11. Alternatives to UCLIN (continued)	
To do this without UCLIN:	**Look here for more information:**
Update the cross-zone subentries of the MOD, LMOD, and TARGETZONE entry	The ZONEEDIT command in *z/OS SMP/E Commands*
Add, rename, or delete a zone	Documentation on the following commands: • To add a zone, you can use the following commands, depending on the particular situation: ZONECOPY and ZONERNAME ZONEEXPORT, UCLIN for the ZONEINDEX and ZONEIMPORT • To rename a zone, you must use the following command: ZONERENAME • To delete a zone, you can use the following commands, depending on the particular situation: ZONEDELETE or ZONEEXPORT or
Change the structure of your system (For example, to add, delete, move, or rename elements or their aliases)	Standard Packaging Rules for z/OS-based Products (publibz.boulder.ibm.com/epubs/pdf/gimpkg80.pdf): section on how to avoid UCLIN by using the appropriate MCSs and operands

2. If you are not the originator of the UCLIN, make sure you understand exactly what is being done and why. If you are not sure, find out before making the update.
3. Make sure the UCLIN is being done in the correct sequence in the process—before or after the installation of the SYSMOD.
4. Make sure all the data is correct.
5. List the entry before changing it. This makes sure you know what the original entry looked like in case an error is reported during the UCLIN or the modification causes an error.
6. After you have done all of these steps, if you have been given directions for installing the UCLIN (for example, in the PTF cover letter or in the program directory for a new function), follow those directions.

How to use UCLIN

UCLIN is used to update the entries in the SMP/E database, just as the SPZAP utility is used to update the system libraries. You can do the following tasks:

- Add new information
- Delete existing information
- Replace existing information with new information

Some UCLIN functions will not work for certain entries or data sets. The chapters on the UCLIN command in *z/OS SMP/E Commands* and SMP/E data set entries in *z/OS SMP/E Reference* provide detailed information about which entries may be modified for each data set, what data within each entry may be modified, and the exact syntax for each entry and data item.

The general format for UCLIN statements is:

```
SET     BDY(xxxxxxxx)         /* Set to correct zone.     */.
UCLIN                         /* Marks start of UCLIN.    */.
```

```
  …                          /*   UCL                     */
  UCL statements               /*   statements              */
  …                          /*   to make modifications. */
ENDUCL                         /* Marks end of UCLIN.      */.
```

The general format of each UCL statement is as follows:

```
ADD|REP|DEL                    /* Type modification        */
   type(name)                  /* Entry type and name      */
   operand                     /*                          */
   operand                     /*     Optional             */
   operand                     /*     operands             */
   operand                     /*                          */
                               /* End of one UCL statement */.
```

Where:

ADD|REP|DEL
Specifies the action to be taken on the entry or operands specified.

In general, ADD means add the entry or operand only if it is not already present; REP means replace the operand if it is already present, otherwise add it; and DEL means delete the entry or operand if it exists. For a more detailed description of the functions provided by ADD, REP, and DEL, see the chapter on the UCLIN command in *z/OS SMP/E Commands*.

type(name)
Specifies the entry type (such as MOD, MAC, ORDER, SRC, data element type, SYSMOD) and name of the entry (such as GIMMPDRV, HELP, ORD000034, MYSRCMOD, MYCLIST, UR12345).

operand
Specifies the individual data items in the entry that are to be modified.

The data items can be:

- Single operands, such as RENT or REUS or COPY
- Single value subentries, such as DISTLIB(AOS12), where only one value can be placed within the parentheses
- Multiple value subentries, such as LMOD(LMOD01,LMOD02,LMOD3) or MAC(MAC01,MAC02), where more than one value can be specified within the parentheses

Chapter 15. Identifying cross-zone requisites: The REPORT CROSSZONE command

This chapter contains information about using the REPORT CROSSZONE command to check for requisites between zones. It discusses these topics:

- How to define the zones to be processed
- How to run the REPORT CROSSZONE command
- How to install the SYSMODs, using the output from the REPORT CROSSZONE command

Introduction

Your system may contain products that are packaged and installed separately, but which have service level or interface dependencies. For example, an interface error in one product may require a change to another product that used the interface. When this happens, a unique PTF is generated for each product. The relationship between the PTFs can be specified in a conditional requisite (++IF) modification control statement in the PTFs. If you have completed the steps listed in "Specifying automatic cross-zone requisite checking" on page 83, SMP/E automatically checks the requisites during APPLY, ACCEPT, and RESTORE processing, whether the products are in the same zone or in separate zones. However, if you have not completed these steps and the products are in separate zones, the requisites are not automatically checked in any other zones. In this case, to make sure these requisites are installed where they are needed, you must:

1. Identify a set of zones to be checked for conditional requisites. These zones may be defined in the same global zone or in different global zones. You may also define groups of zones by creating ZONESET entries in the global zone.

   When defining a ZONESET, all the zones in the ZONESET must be defined by ZONEINDEX subentries in the same global zone as the ZONESET entry.
2. Run the REPORT CROSSZONE command to get a list of the SYSMODs that must be installed and the zones where they are needed.
3. Install the SYSMODs in the indicated zones.

If you just want to check service levels for products, you should use the REPORT SYSMODS command or the LIST command. See Chapter 18, "Comparing the SYSMODs installed in two zones: The REPORT SYSMODS command," on page 169 and "Listing to compare two zones" on page 154 for more information.

Identifying zones to be processed

When identifying zones to be processed, you may group the zones by defining a global zone ZONESET entry. You may have one or more ZONESETs to describe groups of products that might have dependencies on each other or which may be defined in different global zones.

For example, assume that you have a system that supports two products, ABC and XYZ, that have dependencies on one another. You might have one zone, BASEABC, for the base ABC function, and another zone, PRODABC, for a dependent function. Likewise, you might have a zone BASEXYZ for the base XYZ function, and another zone, PRODXYZ, for a dependent function. Let's also assume that all four zones are defined in the same global zone. The dependent functions are different versions of the same product, and they must be synchronized with each other and with their base functions. You can set up two ZONESETs to help keep these products at the same service level.

These are the commands you can use to define the ZONESETs:

© Copyright IBM Corp. 1986, 2026

```
//UCL      JOB 'accounting info',MSGLEVEL=(1,1)
//UCL      EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(GLOBAL)      /* Set to global zone.   */.
UCLIN                     /*                       */.
ADD      ZONESET(ZONEA)   /* Create ZONESET ZONEA. */
         ZONE(BASEABC,    /* Include these zones.  */
              PRODABC,    /*                       */
              PRODXYZ)    /*                       */
         .
ADD      ZONESET(ZONEX)   /* Create ZONESET ZONEX. */
         ZONE(BASEXYZ,    /* Include these zones.  */
              PRODXYZ,    /*                       */
              PRODABC)    /*                       */
         .
ENDUCL                    /*                       */.
/*
```

As a second example, assume that you have the same four zones, but BASEABC and PRODABC are defined in one global zone and BASEXYZ and PRODXYZ are defined in a second global zone. Because each zone in a ZONESET must be defined by a ZONEINDEX subentry in the same global zone as the ZONESET entry, the following ZONESETs may each be defined in their respective global zone:

```
//UCL      JOB 'accounting info',MSGLEVEL=(1,1)
//UCL1      EXEC SMPPROC
//SMPCNTL   DD  *
SET         BDY(GLOBAL)    /* Set to global zone ABC */.
UCLIN                      /*                        */.
ADD         ZONESET(ZONEA) /* Create ZONESET ZONEA   */
            ZONE(BASEABC,  /* Include these zones    */
                 PRODABC)  /*                        */
        .
ENDUCL                     /*                        */.
/*
//UCL2     EXEC SMPPROC
//SMPCNTL  DD *
SET        BDY(GLOBAL)      /* Set to global zone XYZ */.
UCLIN                       /*                        */.
ADD        ZONESET(ZONEX)   /* Create ZONESET ZONEX   */
           ZONE(BASEXYZ,    /* Include these zones    */
                PRODXYZ)    /*                        */
       .
ENDUCL                      /*                        */.
/*
```

When you define a ZONESET, remember:

- Each zone in a ZONESET must also be defined in the global zone.
- Each zone in a ZONESET must be defined in the same global zone. They cannot be defined in global zones that are in different CSI data sets. If you have zones defined in two different global zones, and you wish to use ZONESETs to identify the zones to the REPORT CROSSZONE command for processing, you must create a ZONESET in each global zone to contain the zones that are defined in each global zone.
- A zone can be part of more than one ZONESET.
- A ZONESET can contain both target and distribution zones.

For more information about defining the ZONESET entry, see *z/OS SMP/E Reference*.

Running the REPORT CROSSZONE command

After you identify the zones to be processed and created any desired ZONESET entries, you can run the REPORT CROSSZONE command to get a list of all the SYSMODs that must be installed and which zones need them. This list is the Cross-Zone Requisite SYSMOD report. It identifies which of the needed SYSMODs must be received and which SYSMODs caused the needed SYSMODs to be listed. Besides the report, SMP/E writes commands to the SMPPUNCH data set. You can use them to install the SYSMODs in the appropriate zones.

If all the zones being processed are of the same type, SMP/E determines the type of report to be generated. For example, identifying only target zones to be processed results in a Cross-Zone Requisite SYSMOD report for APPLY processing. On the other hand, the set of zones to be processed can be mixed;

that is, both target and distribution zones are identified. If this is the case, you need to specify which type of Cross-Zone report you want generated.

You can limit which SYSMODs SMP/E reports on by specifying any combination of these operands on the REPORT CROSSZONE command:

- **FORZONE:** to list only SYSMODs needed in specific zones being processed.
- **FORFMID:** to list only SYSMODs needed for specific FMIDs in the set of zones to be processed.
- **DLIBZONE** or **TARGETZONE:** to tell SMP/E which zones you want a report on (if the zones being processed are mixed)

  Note:

  1. DLIBZONE and TARGETZONE are mutually exclusive operands.
  2. FORZONE is mutually exclusive with the ZONES operand.

For more information about the REPORT CROSSZONE command, see *z/OS SMP/E Commands*.

To continue the example where all four zones are defined in the same global zone, assume that you applied a number of SYSMODs to the zones in ZONESET ZONEA. Some of these SYSMODs named conditional requisites. To find out which zones are affected by these requisites, you can use the following commands:

```
//REPORT   JOB 'accounting info',MSGLEVEL=(1,1)
//REPORT   EXEC SMPPROC
//SMPCNTL  DD  *
SET    BDY(GLOBAL)      /* Set to global zone.  */.
REPORT CROSSZONE        /* Report on            */
       ZONESET(ZONEA)   /* ZONESET ZONEA.       */.
/*
```

To continue the example where the zones are defined in two separate global zones, assume that you applied a number of SYSMODs to the zones in ZONESET ZONEA defined in global zone 1 and also to PRODXYZ defined in global zone 2. Some of these SYSMODs named conditional requisites. To find out which zones are affected by these requisites, you can use the following commands:

```
//REPORT  JOB 'accounting info',MSGLEVEL=(1,1)
//REPORT  EXEC SMPPROC
//SMPCNTL DD *
SET    BDY(GLOBAL)        /*Set to global zone.       */.
REPORT CROSSZONE          /* Report on                */
       ZONES(ZONEA,(gzonedsn2,PRODXYZ)) /*
                         ZONESET ZONEA from
                         global zone 1 and PRODXYZ
                         from global zone 2            */.
/*
```

Installing the SYSMODs

After you have the Cross-Zone Requisite SYSMOD report, you can install the missing SYSMODs in the zones where they are needed. Follow these steps:

1. Receive any SYSMODs that have not yet been received into the appropriate global zone.
2. Install the SYSMODs in the zones where they are needed. If you have the SMPPUNCH output from the REPORT CROSSZONE command, you can use that. If the zones were defined in different global zones, SMPCSI and SMPCNTL DD statements would have been generated in the SMPPUNCH output. These DD statements must be copied to your job before the SMPPUNCH can be used.
3. Rerun the REPORT command using the same set of zones to check the results of installing the new SYSMODs.
4. Receive and install any additional SYSMODs that are needed.

For the example in this chapter where all four zones are defined in the same global zone, follow these steps for ZONESET ZONEA and ZONESET ZONEX, because both contain zone PRODXYZ. You can continue

to run the REPORT CROSSZONE command and install SYSMODs until all the needed SYSMODs have been installed in both ZONESETs.

For the example in this chapter where the zones are defined in two different global zones, follow these steps for ZONESET ZONEA and PRODXYZ, and also for ZONESET ZONEX and PRODABC because both contain zone PRODXYZ. You can continue to run the REPORT CROSSZONE command and install SYSMODs until all the needed SYSMODs have been installed in the required zones.

Chapter 16. Identifying installed SYSMODs affected by error holds: The REPORT ERRSYSMODS command

This topic contains information about using the REPORT ERRSYSMODS command to check for installed SYSMODs affected by error holds that were later received. It discusses these topics:

- How to run the REPORT ERRSYSMODS command
- How to use output from the REPORT ERRSYSMODS command for installing the SYSMODs

Introduction

In the course of maintaining your system, you can apply or accept service and later receive HOLDDATA that affects that service. If any of that HOLDDATA was for an error reason ID, you should install a resolving SYSMOD so that the error does not occur on your system. However, SMP/E does not automatically write any reports during RECEIVE processing to help you do this. To see if any installed SYSMODs are affected by error holds that were later received, you must:

1. Run the REPORT ERRSYSMODS command to get a list of the SYSMODs that must be installed and the zones where they are needed.
2. Install the resolving SYSMODs in the indicated zones.
3. Determine whether any resolving SYSMODs are available for held SYSMODs.

Running the REPORT ERRSYSMODS command

After you have received HOLDDATA, you can run the REPORT ERRSYSMODS command to get a list of all the SYSMODs that are affected by any unresolved error holds. This list is the Exception SYSMOD report, which also tells whether any resolving SYSMODs have been received for these holds and whether any of the resolving SYSMODs have error holds against them. Besides the report, SMP/E writes commands to the SMPPUNCH data set. You can use them to install the resolving SYSMODs in the appropriate zones.

You can limit which HOLDDATA SMP/E reports on by specifying any combination of these operands on the REPORT ERRSYSMODS command:

BEGINDATE
To check HOLDDATA entries for error reason IDs that were received by SMP/E on or after the specified date.

ENDDATE
To check HOLDDATA entries for error reason IDs that were received by SMP/E on or before the specified date.

FORFMID
To list only SYSMODs owned by specific FMIDs.

For more information about the REPORT ERRSYSMODS command, see *z/OS SMP/E Commands*.

Here is an example of when you might want to use the REPORT ERRSYSMODS command. Assume that you have just received some HOLDDATA, and you need to know whether it affects any of the SYSMODs you have already accepted into distribution zone DZONE1. You can use the following commands:

```
//REPORT   JOB 'accounting info',MSGLEVEL=(1,1)
//REPORT   EXEC SMPPROC
//SMPCNTL  DD  *
SET    BDY(GLOBAL)         /* process global zone     */.
REPORT ERRSYSMODS          /* report on exception     */
       ZONES(DZONE1)       /* SYSMODs in this zone    */
       BEGINDATE(01 01 07) /* for HOLDDATA received   */
       ENDDATE(02 01 07)   /* between these dates     */.
```

© Copyright IBM Corp. 1986, 2026

Installing the SYSMODs

Once you have the Exception SYSMOD report, you can install the resolving SYSMODs in the zones where they are needed. Follow these steps:

1. If any resolving SYSMODs are held, run REPORT ERRSYSMODS, specifying the global zone, to see if any SYSMODs have been received that resolve the additional holds for the resolving SYSMODs.
2. If the Exception SYSMOD report for the global zone shows resolving SYSMODs for the additional holds, edit the SMPPUNCH output to add the new resolving SYSMODs and to update the selection list so the held resolving SYSMODs will be processed.
3. Use the SMPPUNCH output to install the resolving SYSMODs in DZONE1.

Chapter 17. Listing the source IDs in a zone: The REPORT SOURCEID command

This chapter contains information about using the REPORT SOURCEID command to list the source IDs assigned to SYSMODs in a given zone or ZONESET. It discusses these topics:

- How to run the REPORT SOURCEID command
- How to list SYSMODs using the output from the REPORT SOURCEID command

Introduction

In the course of maintaining your system, you may need to find out which source IDs are assigned to SYSMODs in a given zone. For example, assume you install service using CBPDOs, which assign source IDs to the service SYSMODs they contain. You can use the REPORT SOURCEID command to determine the latest service level you have installed in a particular zone. To determine the service level based on source IDs, follow these steps:

1. Run the REPORT SOURCEID command to get a list of which source IDs are assigned to SYSMODs in a given zone.
2. If desired, list the SYSMOD entries for the SYSMODs with those source IDs.

Running the REPORT SOURCEID command

You can use the REPORT SOURCEID command to get a list of all the source IDs assigned to SYSMODs in a given zone. This list is the SOURCEID report, which may also indicate which SYSMODs these source IDs are assigned to. Besides the report, SMP/E writes commands to the SMPPUNCH data set, which you can use to list the SYSMODs. For information about the REPORT SOURCEID command, see *z/OS SMP/E Commands*.

Here is an example of when you might want to use the REPORT SOURCEID command. Assume you want to find out which source IDs are associated with SYSMODs in target zone TGT1, and you want to know which SYSMODs each source ID is assigned to. You can use the following commands:

```
SET BDY(GLOBAL).
REPORT SOURCEID
       ZONES(TGT1)
       SYSMODIDS.
```

Listing the SYSMODs

If you want more information about the SYSMODs that are assigned the source IDs shown in the SOURCEID report, you can list the related SYSMOD entries. The SMPPUNCH output produced by the REPORT SOURCEID command contains the LIST SYSMOD SOURCEID(...) commands needed to list the SYSMODs for the source IDs in the SOURCEID report. You can tailor the SMPPUNCH output to list the SYSMODs in which you are interested, and run the commands to list the desired SYSMODs.

© Copyright IBM Corp. 1986, 2026

Chapter 18. Comparing the SYSMODs installed in two zones: The REPORT SYSMODS command

This chapter contains information about using the REPORT SYSMODS command to check if SYSMODs installed in one zone are applicable to and installed in a second zone. It discusses these topics:

- How to run the REPORT SYSMODS command
- How to install SYSMODs using the output from the REPORT SYSMODS command

Introduction

In the course of maintaining your system, you may need to compare the function and service level of two zones. For example, if you have installed the same products in different zones, you may want to make sure that both copies of these products are at the same level. SMP/E does not do this kind of checking automatically. To compare the SYSMODs installed in two zones, you must:

1. Run the REPORT SYSMODS command to get a list of the SYSMODs that are installed in the first zone and that are applicable to the second zone but are not yet installed there.
2. Install the applicable SYSMODs in the second zone.

Running the REPORT SYSMODS command

You can run the REPORT SYSMODS command to get a list of all the SYSMODs that are installed in one zone and not in a second. This list is the SYSMOD Comparison report, which also tells which of these SYSMODs are applicable to the second zone, shows if the SYSMODs have been received, and lists any source IDs associated with the SYSMODs. Besides the report, SMP/E writes commands to the SMPPUNCH data set, which you can use to install the SYSMODs. For information about the REPORT SYSMODS command, see *z/OS SMP/E Commands*.

Here is an example of when you might want to use the REPORT SYSMODS command. Assume you have two z/OS systems. The target zones that control these systems are TGZONE1 and TGZONE2, and they are serviced from the same global zone. You want to determine which SYSMODs are installed in TGZONE1 and are not installed in, but are applicable to, TGZONE2. You can use the following commands:

```
SET    BDY(GLOBAL)          /* process global zone     */.
REPORT SYSMODS              /* report on SYSMODs       */
       INZONE(TGZONE1)      /* input zone TGZONE1      */
       COMPAREDTO(TGZONE2)  /* comparison zone TGZONE2 */.
```

Suppose we modify this example slightly and assume that TGZONE1 and TGZONE2 are serviced by different global zones. TGZONE1 is serviced by the global zone in SYS1.GLOBAL1.CSI and TGZONE2 is serviced by the global zone in SYS1.GLOBAL2.CSI. You can use the following commands:

```
SET BDY(GLOBAL) /* process global zone */.
REPORT SYSMODS /* report on SYSMODs */
       INZONE(TGZONE1) /* input zone TGTZONE1 */
       COMPAREDTO(SYS1.GLOBAL2.CSI,TGZONE2) /* comparison zone TGZONE2 */.
```

Because TGZONE1 and TGZONE2 are serviced by different global zones, an SMPCSI DD statement will be generated at the top of the SMPPUNCH output. This DD statement must be moved to your job before the SMPPUNCH output can be used.

Installing the SYSMODs

Once you have the SYSMOD Comparison report, you can install the applicable SYSMODs in the zone where they are needed. Follow these steps:

© Copyright IBM Corp. 1986, 2026

1. Research the report to determine which of the identified SYSMODs you want to install into the comparison zone.
2. Find and receive any applicable SYSMODs that were not available and that you want to install. The source ID in the report identifies some possible sources for obtaining the SYSMODs.
3. Tailor the SMPPUNCH output to install the set of SYSMODs that you deem appropriate; then run the commands to install the desired SYSMODs.

Chapter 19. Building a user modification

This topic discusses steps and considerations for building a user modification (USERMOD). It provides the following information:

- How to choose between building a SYSMOD as a USERMOD and building it as a function
- How to create modification control statements
- Examples of USERMODs

Choosing between a USERMOD and a function SYSMOD

Software products available from IBM provide you with many functions. However, these functions may not always exactly meet your processing requirements. Many of these products provide interfaces, such as user exit routines or dummy modules, that you can use to customize the functions to your needs. Sometimes, however, you may need to change a function substantially. You can do this by either:

- Constructing a user modification as USERMODs to an existing function
- Constructing an additional function SYSMOD

Although you might be able to make these changes without SMP/E, there are advantages to creating them either as USERMODs or as function SYSMODs so that SMP/E can install them.

When SMP/E installs the changes, it does the following:

- Keeps a record of the changes
- Reports any intersections with SYSMODs provided by IBM
- Ensures that the changes are not regressed
- Ensures that the changes are installed properly in the correct libraries
- Lets you remove the changes if there are problems

Before creating your changes, you must decide whether to build a USERMOD type SYSMOD or a function SYSMOD. Table 12 on page 171 lists considerations that will help you make this decision.

Table 12. Comparison of USERMODs and function SYSMODs

USERMOD	Function SYSMOD
Provides changes for elements owned by an existing function. May provide new elements for an existing function.	Provides new elements or new element versions for a new function.
SMP/E reports on changes attempted by PTFs, APARs, or USERMODs.	SMP/E does not report on changes attempted by PTFs, APARs, or USERMODs.
Other SYSMODs for the same function can update the elements.	Because the SYSMOD owns the element, SYSMODs for other functions cannot update the element without also changing the owner of the element.
Better for small changes that affect only a few elements.	Better for major additions to the system.

Creating the MCSs

This section describes some of the considerations for building the MCSs for a USERMOD SYSMOD. See Standard Packaging Rules for z/OS-based Products (publibz.boulder.ibm.com/epubs/pdf/gimpkg80.pdf)

© Copyright IBM Corp. 1986, 2026

for guidelines on when to use USERMODs. See the section on SMP/E modification control statements in *z/OS SMP/E Reference* for more information about MCS syntax.

The ++USERMOD MCS

The ++USERMOD statement identifies this SYSMOD as a USERMOD and assigns a 7-character identifier to the SYSMOD.

The format of the ++USERMOD statement is:

```
++USERMOD(sysmod_id)          /*                        */.
```

The ++VER MCS

The ++VER statement is necessary in all SYSMODs. It describes the environment necessary for installing the SYSMOD.

The general format of the ++VER statement follows:

```
++VER                        /* Environment MCS          */
          (srel)             /* System and release ID    */
      FMID(aaaaaaa)          /* Functional area          */
      PRE (                  /*                          */
           bbbbbbb bbbbbbb   /* Prerequisite PTFs        */
           bbbbbbb bbbbbbb   /*                          */
          )                  /*                          */
      REQ (                  /*                          */
           ccccccc ccccccc   /* Other related USERMODs   */
           ccccccc ccccccc   /*                          */
          )                  /*                          */
      SUP (                  /*                          */
            ddddddd dddddddd /* Other USERMODs incorp-   */
            ddddddd dddddddd /* orated into this one     */
          )                  /*                          */.
```

Specifying the proper system release

The SREL value (*srel*) must be one of those defined as an SREL subentry in the TARGETZONE entry. If the USERMOD is a change to an IBM product, the SREL should correspond to the SREL value specified in the IBM product that currently owns the elements within this SYSMOD.

Specifying the FMID value

If any element is owned by an FMID different from that specified in the ++VER statement, that element is not selected for installation during APPLY or ACCEPT processing, and message GIM45401W is issued. This condition is a SYSMOD construction error. SMP/E supports a SYSMOD construction that assumes that this condition occurs regularly and that the SYSMOD contains an ++IF statement specifying another SYSMOD that supplies the proper functional version of the element.

Note: As was explained earlier, it is a good idea to construct a USERMOD so that each SYSMOD contains only one element. This construction method eliminates this problem.

Specifying the proper requisites

When you specify requisite SYSMODs, you are defining two kinds of relationships:

- The relationship of your SYSMOD to previous versions of the element
- The relationship of your SYSMOD to other SYSMODs currently on the system

The following text describes how you can define these relationships.

Relationships to earlier versions of the elements

1. If the element entry in your target zone has an RMID value different from its FMID value, ensure it is a prerequisite of the USERMOD fix; that is, make sure the *bbbbbbb* value shown in the example is accurate. If the RMID and FMID values are equal, the *bbbbbbb* value need not be specified.
2. If the element entry in your target zone has any UMID values, you should first check to make sure the USERMOD itself was constructed so it works correctly in that environment. You should then make sure each of the UMID values is specified in the PRE operand in place of the *ccccccc* values shown in the example. This is not an absolute requirement, but if it is omitted, SMP/E issues warning messages during installation identifying that these SYSMODs may have an intersection with the one you are installing and, therefore, may be regressed. Putting the UMID values in the PRE list suppresses these messages.
3. If you do not specify the requisites that previously replaced an element, SMP/E does not allow your USERMODs to be installed unless you specify BYPASS(ID) on the APPLY or ACCEPT commands.
4. If you want this SYSMOD installed without any warning or error messages, you must specify all the current UMID values of each element in the SYSMOD in the PRE operand. This indicates to SMP/E that the SYSMOD was designed to be installed on the current function and service level of the element, including update level.

   If you do not specify the requisites that previously updated an element, the following occurs:

   - If your SYSMOD contains an element replacement, SMP/E does not allow the SYSMOD to be installed unless BYPASS(ID) is specified on the APPLY and ACCEPT command.
   - If your SYSMOD contained an element update, SMP/E allows the SYSMOD to be installed, but issues a warning message for each requisite not specified in the PRE list.

   This means SMP/E is unable to determine if there is an intersection between your update and those already on the element. It assumes that there is none, and you should investigate both updates to verify this.

Relationships to Other SYSMODs

Your SYSMOD may depend on another USERMOD or IBM PTF being installed, because you depend on the function provided by that SYSMOD. You may want to indicate that this SYSMOD is part of a set of USERMODs designed to provide some user function. Because each SYSMOD contains only one element, you want to tell SMP/E that they should all be installed together. This is done with the REQ operand (the *ccccccc* values in the example).

Specifying superseded SYSMODs

If this SYSMOD is to fix multiple problems, each of the additional APARs that are being fixed should be specified in the SUP operand (*ddddddd* values in the example). This notifies SMP/E that it is not necessary to install those SYSMODs after the current SYSMOD is installed.

The ++JCLIN MCS

The ++JCLIN statement is necessary in all SYSMODs that add new link-edited load modules or change the link-edit characteristics or link-edit control cards for existing link-edited load modules. The data following the ++JCLIN statement consists of the jobs necessary for installing the new module or link-editing the affected load modules, as if that JCL were actually going to build the load modules from scratch from members of libraries.

The general format of the ++JCLIN statement is:

```
++JCLIN                     /* Installation data        */.
```

++MOD and ++ZAP MCSs

The ++MOD and ++ZAP statements are used to identify a replacement and update to a module in the distribution library and associated load modules.

The data following the ++MOD statement is an object deck. The general format of the ++MOD statement is:

```
++MOD(modname)              /* Distribution name        */
     DISTLIB(dlibname)      /* DLIB ddname              */
…
… Object deck for module
…
```

The data following the ++ZAP statement is a set of superzap control statements. The general format of the ++ZAP statement is:

```
++ZAP(modname)              /* Distribution name        */
     DISTLIB(dlibname)      /* DLIB ddname              */.
…
… Superzap control statement
…
```

The superzap control statements are the same as you would use if you were calling the superzap utility. The only exception is that on the NAME statement, you should specify only the CSECT name within the distribution library module, rather than the load module name and the CSECT name. When SMP/E installs the SYSMOD, it determines all the load modules into which this distribution module was link-edited and then calls the superzap utility for each of these load modules, modifying the NAME statement as appropriate.

++MAC and ++MACUPD MCSs

The ++MAC and ++MACUPD statements are used to identify a replacement and update to a macro in the distribution library and associated target libraries.

The data following the ++MAC statement is the macro replacement. The general format of the ++MAC statement is:

```
++MAC(macname)              /* Macro name               */
     DISTLIB(dlibname)      /* DLIB ddname              */.
…
… Macro replacement
…
```

The data following the ++MACUPD statement is the control statements that would have been used if you called the IEBUPDTE utility. The general format of the ++MACUPD statement is:

```
++MACUPD(macname)           /* Macro name               */
     DISTLIB(dlibname)      /* DLIB ddname              */.
…
… Update control statements
…
```

The following restrictions are enforced by SMP/E:

1. The first statement must be the "./ *change name=macname*" control statement.
2. The name specified on the change statement must be the same as on the ++MACUPD statement.
3. No insert or delete statement can be used. Inserting must be done by manually assigning each line a number. Deleting must be done by commenting out the line. This restriction enables SMP/E to merge the update control statement when multiple SYSMODs modify the same macro.

++SRC and ++SRCUPD MCSs

The ++SRC and ++SRCUPD statements are used to identify a replacement and update to source in the distribution library and associated target libraries. The format of the MCSs and the data format and restrictions are similar to those for macros.

The ++PROGRAM MCS

The ++PROGRAM statement is used to define replacements for program elements, which are pre-built load modules or program objects. (For a complete description of program elements, see the chapter on MCSs in *z/OS SMP/E Reference*.) You can add a program element by packaging it in a USERMOD.

The ++PROGRAM MCS is immediately followed by the load module or program object. The general format of the ++PROGRAM MCS is:

```
++PROGRAM(name)             /* Data element type, name */
     DISTLIB(dlibname)      /* DLIB ddname             */.
…
… Program element replacement
…
```

To be packaged inline, a program element must be unloaded along with its aliases into a sequential data set and then transformed with GIMDTS into the required fixed-block-80 record format before it is packaged. Later, when SMP/E installs the element, it is changed back to its original format. For more information about using GIMDTS, , see *z/OS SMP/E Reference*.

Data element MCSs

Data element MCSs are used to define replacements for data elements, which are elements other than macros, modules, program elements, and source code. These include TSO CLISTs, help panels, ISPF dialog panels, and online publications libraries. For a complete description of data elements, see the section on SMP/E modification control statements in *z/OS SMP/E Reference*. You can add a data element by packaging it in a USERMOD.

The data element MCS is immediately followed by the data element itself. The general format of the data element MCS is:

```
++element(name)             /* Data element type, name */
     DISTLIB(dlibname)      /* DLIB ddname             */.
…
… Data element replacement
…
```

To be packaged inline, a data element must contain fixed-block 80 records. If the original format of the element is not fixed-block 80 records, you can use GIMDTS, a service routine provided with SMP/E, to transform the element into the required format before packaging it. Later, when SMP/E installs the element, it is reconverted to its original format. For more information about using GIMDTS, see *z/OS SMP/E Reference*.

Hierarchical file system element MCSs

The hierarchical file system element MCSs are used to define a replacement for an element residing in a UNIX file system. You can add a hierarchical file system element by packaging it in a USERMOD.

The hierarchical file system element MCS is immediately followed by the hierarchical file system element itself. The general format of a hierarchical file system element MCS is:

```
++hfs_element(hfsname)      /* HFS  name               */
     DISTLIB(dlibname)      /* DLIB ddname             */.
…
… Hierarchical file system element replacement
…
```

To be packaged inline, a hierarchical file system element must contain fixed-block 80 records. If the original format of the element is not fixed-block 80 records, you can use GIMDTS, a service routine provided with SMP/E, to transform the element into the required format before packaging it. Later, when SMP/E installs the element, it is reconverted to its original format. For more information about using GIMDTS, see *z/OS SMP/E Reference*.

Examples of USERMODs

This section contains examples of USERMODs that:

- Update a module
- Replace a module
- Add new modules
- Replace a macro or source code
- Update a macro or source code
- Add new source code
- Add source code that uses an IBM-supplied macro
- Add a module that uses an IBM-supplied macro

Example 1: Updating a module

This is an example of a USERMOD that updates module MODULEA.

```
++USERMOD(USMP001)         /* SYSMOD ID of USERMOD.    */.
++VER(Z038)                 /* SREL for MVS.            */
     FMID(HMP1900)          /* Applicable to SMPE.      */
     .                      /*                          */
++ZAP(MODULEA)             /* Name of module.          */
     DISTLIB(AOS12)         /* ddname of DLIB.          */
                            /*                          */.
  NAME MODULEA
* Verify existing ddname.
  VER  0050  404040404040404000
* Verify existing SYSOUT class.
  VER  0058  40
* Verify existing terminal assignment.
  VER  0059  00
* Replace with new data.
*            M Y P R I N T  = ddname
  REP  0050  D4E8D7D9C9D5E3
*            A              = SYSOUT class
  REP  0058  C1
```

Note:

1. You should verify each location that you are going to update.
2. You can specify only one name in the NAME statement.
3. The changes made by the REP statements are explained by the preceding comment lines.

Example 2: Replacing a module

This is an example of a USERMOD that replaces module MODULEB. After writing and assembling MODULEB, package it in a USERMOD, as shown later in this section:

```
++USERMOD(USMP002)         /* SYSMOD is for USERMOD.   */.
++VER(Z038)                 /* SREL for MVS.            */
     FMID(HMP1900)          /* Applicable to SMPE.      */
     .                      /*                          */
++MOD(MODULEB)             /* Name of module.          */
     DISTLIB(AOS12)         /* ddname of DLIB.          */
                            /*                          */.
...
... object module for MODULEB
...
```

Note: There are no PRE operands on the ++VER statement, because this module has not been modified since its initial installation. Therefore, only the ++VER FMID operand is required.

Example 3: Adding new modules

This is an example of a USERMOD that adds new modules to create a new load module. In this example, the new modules are SMPMOD01, SMPMOD02, and SMPMOD03. These modules are link-edited to create load module SMPEXT01, whose entry point is SMPMOD01. The target library for SMPEXT01 is SYS1.LINKLIB. The following example shows how the new modules and load modules can be packaged in a USERMOD.

```
++USERMOD(USMP003)         /* SYSMOD ID of USERMOD.    */.
++VER(Z038)                 /* SREL for MVS.            */
     FMID(HMP1900)          /* Applicable to SMPE.      */
                            /*                          */.
++JCLIN                     /* JCLIN to install routine.*/.
//JOB1     JOB 'accounting info',MSGLEVEL=(1,1)
//STEP1    EXEC PGM=IEWL
//PVTDLIB1 DD DSN=SYS1.PVTDLIB1,DISP=OLD
//SYSLMOD  DD DSN=SYS1.LINKLIB,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSLIN   DD *
  INCLUDE  PVTDLIB1(SMPMOD01)
  INCLUDE  PVTDLIB1(SMPMOD02,SMPMOD03)
  ENTRY    SMPMOD01
  NAME     SMPEXT01(R)
/*
++MOD(SMPMOD01)            /* Customized exit routine. */
     DISTLIB(PVTDLIB1)      /* ddname of DLIB.          */
                            /*                          */.
...
... object module for SMPMOD01
...
++MOD(SMPMOD02)            /* Customized exit routine. */
     DISTLIB(PVTDLIB1)      /* ddname of DLIB.          */
                            /*                          */.
...
... object module for SMPMOD02
...
++MOD(SMPMOD03)            /* Customized exit routine. */
     DISTLIB(PVTDLIB1)      /* ddname of DLIB.          */
                            /*                          */.
...
... object module for SMPMOD03
...
```

Note:

1. The ++JCLIN statement is required because the SYSMOD provides new modules and a new load module.
2. This SYSMOD could have been packaged as a ++FUNCTION, because each of the modules is user-written.
3. This SYSMOD could have been packaged as three separate SYSMODs, each containing one module. In that case, each SYSMOD would then need to specify the ++VER REQ operand so the SYSMODs would be installed as a set.

Example 4: Replacing a macro or source code

This is an example of a USERMOD that adds a new macro. In this example, the macro is installed in a target and a distribution library. In addition, source code element USRSRC01 is to be assembled when the macro is installed.

```
++USERMOD(UJES004)          /* SYSMOD ID of USERMOD.    */.
++VER(Z038)                 /* SREL for MVS.            */
     FMID(HJE2102)          /* Applicable to JES.       */
                            /*                          */.
++MAC(USRMAC01)             /* Name of new macro.       */
     DISTLIB(USRMACLB)      /* ddname of DLIB.          */
     SYSLIB (MACLIB)        /* ddname of system MACLIB. */
     ASSEM  (USRSRC01)      /* Reassemble this source.  */
```

```
                              /*                          */.
...
... macro replacement
...
```

Note:

1. No JCLIN is required to install a new macro; all necessary information can be specified on the ++MAC statement.
2. You can follow this example to add new source code, with the following exceptions:
   - The ASSEM operand is not required for source code. If SMP/E understands where source code is to be installed, it automatically tries to assemble the new version of the source code after replacing the old version.
   - To define how the source code should be installed, you should include a ++JCLIN statement followed by JCLIN data. The JCLIN data should be similar to that in the example of adding a new module.

Example 5: Updating a macro or source code

The following is an example of a USERMOD to update a macro that was added by a previous USERMOD.

```
++USERMOD(UJES005)          /* SYSMOD ID of USERMOD.    */.
++VER(Z038)                 /* SREL for MVS.            */
     FMID(HJE2102)          /* Applicable to JES.       */
     PRE (UJES004)          /* Previous replacement.    */
                            /*                          */.
++MACUPD(USRMAC01)          /* Update customized macro. */
     DISTLIB(USRMACLB)      /* ddname of DLIB.          */
                            /* SMPE knows SYSLIB.       */
     ASSEM(USRSRC01)        /* Reassemble this source.  */
                            /*                          */.
./ CHANGE NAME=USRMAC01
...
... macro changed lines here with
... sequence numbers in columns 73-80
...
```

Note:

1. The ++VER PRE operand specifies the previous SYSMOD that replaced the macro.
2. You can follow this example to update source code, with the following exceptions:
   - The ASSEM operand is not used for source code.
   - Because the SYSMOD adding the source code defined how the module should be installed, there is no need to repeat that information about a ++JCLIN statement in the USERMOD.

Example 6: Adding new source code

Assume that you have written new source code to be added to an existing product. It is a member in one of your partitioned data sets. To be installed, it must be assembled, then copied as a single-module load module into its target library. Table 13 on page 178 shows the information you need to specify and where to specify it.

Table 13. Information needed to add new source code

Information to provide:	Where to specify it:
1 Name of source code element: IFBSRC01	• Element name on ++SRC MCS: `++SRC(IFBSRC01)`

Table 13. Information needed to add new source code (continued)

Information to provide:	Where to specify it:
2 Name of object module produced by assembly: IFBSRC01 (same as source code element)	• JCLIN data, M= value on COPY SELECT statement: `COPY SELECT M=(IFBSRC01)`

3 Where input source code is provided:
PDS member
	• TXLIB value on ++SRC MCS:
	`++SRC(IFBSRC01) TXLIB(REPLACE)`
	• DDDEF entry or DD statement for APPLY and ACCEPT processing:
	`REPLACE DD DSN=…`

4 Target library for source:
SYS1.IFBSRC
	• SYSLIB value on ++SRC MCS:
	`++SRC(IFBSRC01) SYSLIB(IFBSRC)`
	• JCLIN data, OUTDD value on COPY statement:
	`COPY INDD=`*`inddval`*`,OUTDD=IFBSRC TYPE=SRC`
	• DDDEF entry or DD statement for APPLY processing:
	`IFBSRC DD DSN=SYS1.IFBSRC`

5 Target library for load module:
SYS1.LPALIB
	• JCLIN data, OUTDD value on COPY statement:
	`COPY INDD=`*`inddval`*`,OUTDD=LPALIB TYPE=MOD`
	• DD statement in JCLIN data and in SMP/E job for APPLY processing:
	`LPALIB DD DSN=SYS1.LPALIB`

6 Distribution library for source code:
SYS1.AIFBSRC
	• DISTLIB value on ++SRC MCS:
	`++SRC(IFBSRC01) DISTLIB(AIFBSRC)`
	• JCLIN data, INDD value on COPY statement:
	`COPY INDD=AIFBSRC,OUTDD=IFBSRC TYPE=SRC`
	• DD statement or DDDEF entry for ACCEPT processing:
	`AIFBSRC DD DSN=SYS1.AIFBSRC`

Table 13. Information needed to add new source code (continued)

Information to provide:	Where to specify it:

7 Distribution library for assembled object module:
SYS1.AOS23
	• DISTMOD value on ++SRC MCS:
	`++SRC(IFBSRC01) DISTMOD(AOS23)`
	• JCLIN data, INDD value on COPY statement:
	`COPY INDD=AOS23,OUTDD=LPALIB  TYPE=MOD`
	• DD statement in JCLIN data and in SMP/E job during APPLY and ACCEPT processing:
	`AOS23 DD DSN=SYS1.AOS23`

8 How to install the source code:
Assemble, then copy
	• ++SRC MCS automatically notifies SMP/E to assemble the module.
	• JCLIN data, COPY steps:
	`//JOB1  JOB …`
	`//STEP1  EXEC PGM=IEBCOPY`
	`//AIFBSRC DD DSN=SYS1.AIFBSRC,…`
	`//IFBSRC  DD DSN=SYS1.IFBSRC,…`
	`//AOS23  DD DSN=SYS1.AOS23,…`
	`//LPALIB  DD DSN=SYS1.LPALIB,…`
	`//SYSIN  DD *`
	`  COPY INDD=AIFBSRC,OUTDD=IFBSRC  TYPE=SRC`
	`  SELECT M=(IFBSRC01)`
	`  COPY INDD=AOS23,OUTDD=LPALIB  TYPE=MOD`
	`  SELECT M=(IFBSRC01)`
	`/*`
	• The OPTIONS and UTILITY entries in effect for the APPLY or ACCEPT command being processed specify the names of the assembler and copy utilities to use and parameters to be passed to them.

The following is a sample USERMOD that can be used to package the source code. The numbers associate items in the SYSMOD with the information listed in Table 13 on page 178.

```
                ++USERMOD(USR0001)          /* My USERMOD.         */.
                ++VER(Z038) FMID(MYFMID1).  /* For my function.    */.
  8             ++JCLIN.                    /* Link object module. */.
                //JOB1    JOB …
                //STEP1   EXEC PGM=IEBCOPY
  6             //AIFBSRC DD DSN=SYS1.AIFBSRC,DISP=SHR
  4             //IFBSRC  DD DSN=SYS1.IFBSRC,DISP=SHR
  7             //AOS23   DD DSN=SYS1.AOS23,DISP=SHR
  5             //LPALIB  DD DSN=SYS1.LPALIB,DISP=SHR
                //SYSIN   DD *
  4 , 6           COPY INDD=AIFBSRC,OUTDD=IFBSRC  TYPE=SRC
  2               SELECT M=(IFBSRC01)
  5 , 7           COPY INDD=AOS23,OUTDD=LPALIB  TYPE=MOD
  2               SELECT M=(IFBSRC01)
                /*
1 , 2 , 8       ++SRC(IFBSRC01)             /* My source module.   */
  3                 TXLIB(REPLACE)          /* Where source is.    */
  6                 DISTLIB(AIFBSRC)        /* DISTLIB for source. */
  7                 DISTMOD(AOS23)          /* DISTLIB for object. */
  4                 SYSLIB(IFBSRC)          /* SYSLIB for source.  */.
```

Example 7: Adding new source code that uses an IBM-supplied macro

Assume you are packaging one of your user-written routines as a USERMOD (UMOD001). Your USERMOD includes assembler source (SRCPART), which is to be included in load module LOADMOD1 and is packaged as a SRC element with a ++SRC statement, as described in previous USERMOD examples.

SRCPART refers to an IBM-supplied macro (IBMMAC), which was packaged in its owning product with a ++MAC statement. You want SRCPART to be automatically reassembled every time IBMMAC is updated or replaced with service. To accomplish this, your USERMOD must contain a JCLIN assembly step in addition to the necessary SMP/E MCS statements. (This means you need to supply the same SRCPART definition as part of the JCLIN input, as well as after the ++SRC statement.) Your USERMOD might look something like this:

```
++USERMOD(UMOD001).
++VER(srel) FMID(fmid).
++JCLIN.
//STEP1    EXEC PGM=IEUASM
//SYSPUNCH DD DSN=&&PUNCH(SRCPART),DISP=(PASS);
//SYSIN    DD *
SRCPART CSECT
        IBMMAC
        BR   14
        END
//LINK     EXEC PGM=IEWL
//SYSLMOD  DD DSN=SYS1.LINKLIB,DISP=SHR
//SYSLIN   DD *
  INCLUDE  SYSPUNCH(SRCPART)
  NAME LOADMOD1
/*
++SRC(SRCPART) SYSLIB(tgtlib) DISTLIB(dlib).
SRCPART CSECT
        IBMMAC
        BR   14
        END
```

When the assembly step in the JCLIN is processed, a GENASM subentry is added to the MAC entry for IBMMAC indicating SRCPART is to be assembled whenever IBMMAC is updated. So, if you install a SYSMOD that updates IBMMAC, SMP/E automatically reassembles SRCPART and link-edits the new copy into LOADMOD1.

Note: Remember, when a link-edit step contains a SYSLMOD DD statement, SMP/E uses the low-level qualifier of the data set name to determine the ddname of the load module's target library (and, by extension, the name of the DDDEF entry to use when allocating this library). In this USERMOD, for example, SMP/E looks for a DDDEF entry named LINKLIB in order to allocate the SYSLMOD data set.

Example 8: Adding a new module that uses an IBM-Supplied macro

Assume you are packaging one of your user-written routines as a USERMOD (UMOD001). Your USERMOD includes an object module (SRCPART), which is to be included in load module LOADMOD1 and is packaged as a MOD element with a ++MOD statement, as described in previous USERMOD examples. SRCPART refers to an IBM-supplied macro (IBMMAC), which was packaged in its owning product with a ++MAC statement. You want to be notified whenever IBMMAC is updated or replaced with service so you can update your module and reapply your USERMOD. To accomplish this, your USERMOD must include the macro, and must specify the last SYSMOD that replaced the macro (the RMID value in the MAC entry) and all the SYSMODs that have updated the macro (the UMID values in the MAC entry) as prerequisites. Your USERMOD might look something like this:

```
++USERMOD(UMOD001).
++VER(srel) FMID(fmid) PRE(macrmid).
++JCLIN.
//LINK     EXEC PGM=IEWL
//SYSLMOD  DD DSN=SYS1.LINKLIB,DISP=SHR
//SYSLIN   DD *
  INCLUDE  DLIB1(SRCPART)
  NAME LOADMOD1
/*
++MAC(IBMMAC).
  …
  macro source
  …
++MOD(SRCPART) DISTLIB(dlib).
  …
  module object deck
  …
```

Because you specified the macro's RMID and UMIDs, when IBMMAC is serviced, the APPLY will get a MODID error for the USERMOD. You will have to restore the USERMOD to successfully apply the service. Then you can rework the USERMOD and apply it again.

Note: Remember, when a link-edit step contains a SYSLMOD DD statement, SMP/E uses the low-level qualifier of the data set name to determine the ddname of the load module's target library (and, by extension, the name of the DDDEF entry to use when allocating this library). In this USERMOD, for example, SMP/E looks for a DDDEF entry named LINKLIB in order to allocate the SYSLMOD data set.

Chapter 20. Determining which SYSMODs led others to fail: The causer SYSMOD summary report

This topic describes root cause analysis and provides examples of how to use the causer SYSMOD information in the Causer SYSMOD Summary report and the SYSMOD Status report to determine the cause of SYSMOD failure.

Introduction

A *causer SYSMOD* is a SYSMOD whose failure led to the failure of other SYSMODs. To identify causer SYSMODs, SMP/E performs root cause analysis for the ACCEPT, APPLY, and RESTORE commands. The types of errors SMP/E analyzes to determine causer SYSMODs include the following:

- Held SYSMODs
- Missing requisite SYSMODs
- Utility program failures: copy, update, assembler, link, zap
- Out-of-space conditions: *x*37 ABENDs
- Missing DD statements and other allocation errors
- ID errors (a SYSMOD does not supersede or specify as a prerequisite an RMID or a UMID of an element)
- JCLIN errors (syntax errors)

The results of SMP/E's root cause analysis are presented in two reports:

- SYSMOD Status report

  This report summarizes the processing done for each eligible SYSMOD. For SYSMODs that failed, the report lists the causer SYSMODs.

  After you check the SYSMOD Status report to determine the results of processing, use the Causer SYSMOD Summary report to determine which errors need to be corrected.
- Causer SYSMOD Summary report

  This report lists the causer SYSMODs along with a brief summary of the related messages, descriptions of the errors that caused the SYSMODs to fail, and, when feasible, some causes for those errors.

Using causer SYSMOD information

How you use the causer SYSMOD information provided by the Causer SYSMOD Summary report and the SYSMOD Status report depends on what you need to install—all the SYSMODs specified on the command you are processing, or only specific ones. This section describes some general scenarios and includes an example of using these reports.

Resolving errors for all SYSMODs that failed

Suppose you are installing a CBPDO, but the reports show that several of the SYSMODs failed. You want to resolve all the reported errors and install all the SYSMODs in the CBPDO.

1. Go to the Causer SYSMOD Summary report to identify the causer SYSMODs and determine how to resolve the errors.
2. Resolve the errors.
3. Rerun the command that failed. If you find more errors on this pass, repeat the process.

© Copyright IBM Corp. 1986, 2026

Resolving errors for a single SYSMOD that failed

Suppose you are installing a group of SYSMODs, but the reports show that several of them have failed. You want to resolve the errors for one specific SYSMOD and install it.

1. Use the SYSMOD Status report to determine the causer SYSMOD for the SYSMOD that you need to install.
2. Go to the Causer SYSMOD Summary report and determine how to resolve the errors for the causer SYSMOD.
3. Resolve the errors.
4. Rerun the command to install the SYSMOD that previously failed. If you find more errors on this pass, repeat the process.

Example

Suppose you ran the following command:

```
APPLY  S(UR00001) GEXT CHECK.
```

and the SYSMOD Status report included the results shown in Figure 36 on page 184.

```
Page nnnn  - NOW SET TO TARGET ZONE nnnnnnn  DATE mm/dd/y
y  TIME hh:mm:ss SMP/E 36.nn SMPRPT OUTPUT


 SYSMOD STATUS REPORT FOR APPLY PROCESSING     SYSMODS APPLIED - 0

 NOTE:  '-' INDICATES THE REQUISITE SYSMOD OR HOLD CONDITION IS NOT SATISFIED
        '*' INDICATES THE NON SATISFIED REQUISITE SYSMOD OR HOLD CONDITION IS BYPASSED
        '#' INDICATES THE SUPERSEDING SYSMOD WAS NOT PROCESSED

 SYSMOD   STATUS    TYPE      FMID      REQUISITE SYSMODS, SUPBY SYSMODS, HOLD REASON-IDS, AND CAUSER
SYSMODS

 AR00001  NOGO                          SUPBY  #UR00002

 UR00001  HELD      PTF       HROOTC1   HOLDE  -AR00001
                                        CAUSER  UR00002

 UR00002  HELD      PTF       HROOTC1   HOLDE  -AR00003
                                        CAUSER  UR00002
```

Figure 36. SYSMOD Status Report: Sample Report for APPLY

In this case, the report indicates that APPLY processing failed for SYSMODs UR00001 and UR00002 because of unresolved error holds. The causer SYSMOD for both PTFs is UR00002. Next, you look up UR00002 in the Causer SYSMOD Summary report, shown in Figure 37 on page 184.

```
Page nnnn  - NOW SET TO TARGET ZONE nnnnnnnn  DATE mm/dd/
yy  TIME hh:mm:ss SMP/E 36.nn SMPRPT OUTPUT

CAUSER SYSMOD SUMMARY REPORT FOR APPLY PROCESSING

CAUSER     FMID     MESSAGE ID  PAGE    ERROR DESCRIPTION AND POSSIBLE CAUSES

UR00002    HROOTC1  GIM35901I      2    ERROR HOLD AR00003 WAS NOT RESOLVED.
```

Figure 37. Causer SYSMOD summary report: sample report for APPLY

That report shows that APPLY processing failed because error hold AR00003 was not resolved for SYSMOD UR00002. By resolving this hold, you will fix the errors listed in the SYSMOD Status report.

Chapter 21. Java archive update exploiter's guide

This chapter is a guide to packaging Java Archive files (JAR files) and updates for JAR files in SMP/E. The intended audience for this section is anyone who will be exploiting this function, including product packagers and service teams. This guide provides information and examples about packaging a product release using the ++JAR MCS and building subsequent PTFs to service that release using the ++JARUPD MCS.

JAR replacements in FMIDs

Suppose your product has a simple TicTacToe applet. The complete TicTacToe applet (or Java application) is composed of a class file and audio and image files, and is stored in a directory structure as follows:

```
/u/pezk/TicTacToe/TicTacToe.class
                 /audio/beep.au
                       /ding.au
                       /yahoo.au
                 /images/cross.gif
                        /not.gif
```

To build a JAR file element for the complete TicTacToe applet, you would first make your working directory the TicTacToe directory where the applet files reside, and then run the following jar command:

```
cd /u/pezk/TicTacToe/
jar cvf ABCTTT.jar *
```

Specified in this way, the jar command takes all files in the working directory (/u/pezk/TicTacToe), and all files within subdirectories of the working directory, and creates a JAR file named ABCTTT.jar.

Your product release, or FMID, can then contain this single JAR file, which is the complete TicTacToe applet. You can do this using the ++JAR MCS as seen in the following example:

```
++JAR(ABCTTT) DISTLIB(AABCBIN) SYSLIB(SABCBIN) RELFILE(2)
  PARM(PATHMODE(0,6,4,4))
  LINK('../TicTacToe.jar')
  SYMLINK('../../../../../usr/lib/TicTacToe.jar')
  SYMPATH('../../usr/lpp/abc/bin/TicTacToe.jar').
```

JAR updates in PTFs

Suppose some time after your FMID has been made available and is being used, a defect is discovered (an APAR). The TicTacToe applet requires a change because of the APAR. Specifically, you need to add another image file (images/new.gif), and replace the class file (TicTacToe.class) with an updated copy of the class file. Further, suppose that the change team has placed the new image file and updated class file in directories, as follows:

```
/u/apars/ow12345/TicTacToe/TicTacToe.class
/images/new.gif
```

You can instruct SMP/E to add new files to a JAR file, and replace files in an existing JAR file by using the ++JARUPD MCS. If files TicTacToe.class and new.gif were to be packaged and archived together in a JAR file of their own, then that file, in SMP/E terms, would be considered a JAR update file. For example, given the preceding directory structure for the new and updated files, the jar command could be used to create the JAR update file as follows:

```
cd /u/apars/ow12345/TicTacToe/
jar cvf ABCTTT.jarupd *
```

© Copyright IBM Corp. 1986, 2026

The resultant JAR file ABCTTT.jarupd would be packaged as a ++JARUPD in a PTF as in the following example:

```
++PTF(UW12345).
++VER(Z038) FMID(fmid).
++JARUPD(ABCTTT)
    PARM(PATHMODE(0,6,4,4)) JARPARM(0M)
    LINK('../TicTacToe.jar')
    SYMLINK('../../../../../usr/lib/TicTacToe.jar')
    SYMPATH('../../usr/lpp/abc/bin/TicTacToe.jar').
```

Suppose now another defect (APAR) is discovered against the TicTacToe applet, and you must update the /audio/beep.au file within the archive. Again, you can use the ++JARUPD MCS to describe an update to the archive. Assuming the replacement audio file resides in a directory as follows:

```
/u/apars/ow54321/TicTacToe/audio/beep.au
```

The following jar command could create the necessary JAR update:

```
cd /u/apars/ow54321/TicTacToe/
jar cvf ABCTTT.jarupd *
```

The resulting JAR file ABCTTT.jarupd would be packaged as a ++JARUPD in a PTF as in the following example:

```
++PTF(UW54321).
++VER(Z038) FMID(fmid) PRE(UW12345).
++JARUPD(ABCTTT)
    PARM(PATHMODE(0,6,4,4)) JARPARM(0M)
    LINK('../TicTacToe.jar')
    SYMLINK('../../../../../usr/lib/TicTacToe.jar')
    SYMPATH('../../usr/lpp/abc/bin/TicTacToe.jar').
```

Notice the second PTF has a prerequisite for the first PTF. Such a relationship is required by SMP/E because both PTFs update the same JAR file.

JAR replacements in PTFs

Just as replacements for JAR files can be delivered in an FMID, so can JAR file replacements be delivered in PTFs. Suppose that you decide to create a "level-set" PTF, which incorporates the updates from several previous APARs and PTFs. In this case, you want to replace the entire TicTacToe applet JAR file, rather then add or replace files within the archive.

You create the JAR file for the applet using the jar command as discussed earlier and then package it as a ++JAR element in a PTF, as in the following example:

```
++PTF(UW87654).
++VER(Z038) FMID(fmid) SUP(UW12345 UW54321).
++JAR(ABCTTT) DISTLIB(AABCBIN) SYSLIB(SABCBIN)
     PARM(PATHMODE(0,6,4,4)) JARPARM(0M)
     LINK('../TicTacToe.jar')
     SYMLINK('../../../../../usr/lib/TicTacToe.jar')
     SYMPATH('../../usr/lpp/abc/bin/TicTacToe.jar').
```

Notice this PTF supersedes both PTFs that supplied updates for the JAR file. Again, such relationships (either supersede or prerequisite) are required by SMP/E, because both PTFs provided updates for the JAR file contained in the current PTF.

Appendix A. Migration

Migration overview

Your plan for migrating to the new level of SMP/E should include information from various sources. These sources of information describe topics such as coexistence, service, migration procedures, and interface changes.

The following documentation, which is supplied with your product order, provides information about installing your z/OS system. In addition to specific information about SMP/E, this documentation contains information about all of the z/OS elements.

- *z/OS Planning for Installation*

  This content outlines the installation requirements for z/OS at both the system and element levels. It includes hardware, software, and service requirements for both the driving and target systems. It also describes any coexistence considerations and actions.

- *SMP/E Program Directory*

  This information, which is provided with your SMP/E product order, leads you through the specific installation steps for SMP/E.

- *ServerPac Installing Your Order*

  This is the order-customized installation information for using the ServerPac installation method. Be sure to review "Appendix A. Product Information", which describes data sets supplied, jobs or procedures that have been completed for you, and product status. IBM may have run jobs or made updates to PARMLIB or other system control data sets. These updates could affect your migration.

Within this information, you can find information about the specific updates and considerations that apply to this release of SMP/E.

Terms you need to know

This section describes some terms you may need to know as you use this section.

Migration
: Activities that relate to the installation of a new version or release of a program to replace an earlier level. Completion of these activities ensures that the applications and resources on your system will function correctly at the new level.

Coexistence
: Two or more systems at different levels (for example, software, service, or operational levels) that share resources. Coexistence includes the ability of a system to respond in the following ways to a new function that was introduced on another system with which it shares resources: ignore a new function, terminate gracefully, support a new function.

  New releases of SMP/E generally add to or enhance the information stored in SMP/E data sets. This new or enhanced data is not always compatible with previous releases of SMP/E. Results are unpredictable when a prior release of SMP/E encounters data introduced by a later release. Therefore, IBM provides toleration PTFs that allow selected prior releases to ignore changes introduced by a later release. In some cases, a compatibility PTF or small programming enhancement (SPE) is available to add a new function to selected prior releases. *z/OS Planning for Installation*. provides a list of the required toleration and compatibility PTFs for each prior release of SMP/E.

SMP/E release levels

The SMP/E element of z/OS is not updated with every release of z/OS. For example, the level of SMP/E that was shipped with z/OS Version 1 Release 8.0 is identical to that shipped with z/OS V1R9.0.

© Copyright IBM Corp. 1986, 2026

Developing a migration strategy

The recommended steps for migrating to a new release of SMP/E are:

1. Become familiar with the supporting migration and installation documentation for the release.

   Determine what updates are needed for products that are supplied by IBM, system libraries, and non-IBM products. Review *z/OS Planning for Installation* and *z/OS Introduction and Release Guide* for information about SMP/E and other z/OS elements.
2. Develop a migration plan for your installation.

   When planning to migrate to a new release of SMP/E, you must consider high-level support requirements, such as machine and programming restrictions, migration paths, and program compatibility.
3. Install the product using the *SMP/E Program Directory* or the *ServerPac Installing Your Order* documentation.
4. Verify that any application programs that use the SMP/E API will continue to run and, if necessary, make changes to ensure compatibility with the new release.
5. Use the new release before initializing major new function.
6. If necessary, customize the new function for your installation.
7. Exercise the new functions.

Reviewing changes to SMP/E processing

As you define your installation's migration plan, consider how the new and changed SMP/E functions might affect the following areas of SMP/E processing. For each item described in “SMP/E V3R4 overview” on page 195, review the "Migration procedures" section to determine how, or if, the change affects the tasks that are performed at your installation.

Customization
You can customize some SMP/E functions to take advantage of new support after the product is installed. For example, you can tailor SMP/E through the use of installation exit routines or SMPPARM options. This section lists changes to SMP/E that might require you to tailor SMP/E to ensure that it runs as before.

Operations
The new SMP/E release might introduce changes to its operating characteristics, such as changed commands, new or changed messages, or in the methods of implementing new functions. This book identifies those changes for which you should provide user education before using this release of SMP/E.

Programming
To ensure that existing programs run as before, programmers who develop programs that use the GIMAPI, library change file records, or UNIX shell scripts must be aware of any changes to those interfaces introduced in a new release of SMP/E.

Reviewing changes to SMP/E interfaces

When defining your installation's migration plan, also consider that SMP/E interfaces may also be affected by the new or changed functions that are introduced in this release. These interfaces include:

- Commands
- Exits
- Macros
- Messages
- Panels
- SYS1.SAMPLIB members

SMP/E V3R7 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R7. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

Deprecate SHA-1 hash validation (IO29474)

The following updates were made to SMP/E because the SHA-1 hash algorithm for GIMZIP package verification is deprecated.

SMP/E requires digital signature verification for digitally signed GIMZIP packages.

GIMZIP packages may be either digitally signed or unsigned. When SMP/E downloads or processes a GIMZIP package, it verifies the package contents by using hash values that are embedded within the package. Digitally signed packages include a digital signature, SHA-256, and SHA-1 hashes, while unsigned packages contain only SHA-1 hashes. Verification using the digital signature and SHA-256 was optional and depended on the presence of the `signaturekeyring` attribute in the CLIENT XML of your SMP/E RECEIVE jobs. SMP/E previously verified both signed and unsigned packages using SHA-1 by default. However, since SHA-1 is a deprecated hash algorithm, SMP/E was updated to disable SHA-1 verification by default.

The SMP/E commands RECEIVE FROMNETWORK, RECEIVE FROMNTS, RECEIVE ORDER, and the GIMGTPKG program no longer support SHA-1 hash verification for signed GIMZIP packages. Instead, digital signature verification is now mandatory for signed packages. This behavior cannot be overridden. All GIMZIP packages for IBM software products and PTF delivery offerings are digitally signed. This verification process applies to PTF and HOLDDATA requests initiated by the SMP/E RECEIVE ORDER command, as well as to all product and PTF orders submitted through Shopz.

Therefore, jobs that execute the SMP/E commands RECEIVE ORDER, RECEIVE FROMNETWORK, RECEIVE FROMNTS, or program GIMGTPKG must specify the `signaturekeyring` attribute in the CLIENT XML. For information about defining the signature validation key ring and updating the <CLIENT> XML tag, see Chapter 6, “Preparing to verify signatures for GIMZIP packages,” on page 113.

SMP/E default processing does not allow GIMZIP packages that are not digitally signed.

By default, SMP/E does not verify GIMZIP packages using the deprecated SHA-1 hash algorithm and does not download or process unsigned GIMZIP packages because these packages contain only SHA-1 hashes. However, this behavior can be overridden based on access to a new SAF resource. If a user executing the SMP/E commands RECEIVE FROMNETWORK, RECEIVE FROMNTS, RECEIVE ORDER, or the GIMGTPKG program has UPDATE access to the SAF FACILITY class resource GIM.ACTION.ALLOW_UNSIGNED_GIMZIP (introduced by APAR IO29474), SMP/E will allow and verify unsigned GIMZIP packages using SHA-1. Because unsigned packages lack the more secure SHA-256 hashes and digital signatures, it is important to determine whether your software vendors provide digitally signed GIMZIP packages. If any vendor offers only unsigned packages, you must ensure that the user identities performing SMP/E RECEIVE operations have the appropriate access to the new SAF resource. For more information about authorizing the use of unsigned GIMZIP packages, see “Authorizing use of unsigned GIMZIP packages” on page 54.

A new option is added for program GIMZIP to produce digitally signed packages that do not contain SHA-1 hashes.

The `securitylevel` attribute for the <GIMZIP> tag indicates the security level for the GIMZIP package and is described in GIMZIP package control tags in *z/OS SMP/E Reference*.

HTTPS download support (IO20858)

The SMP/E RECEIVE FROMNETWORK and RECEIVE ORDER commands, as well as the GIMGTPKG service routine, were updated to allow a user to choose FTP, HTTP, or HTTPS as the protocol for downloading files.

If the identified remote server supports the chosen protocol, then SMP/E will download GIMZIP packages to the local z/OS using the selected protocol.

Two new attributes were added to the <CLIENT> tag in the CLIENT data set.

- `downloadmethod="ftp | http | https"`
- `downloadkeyring="keyring-name" | "javatruststore"`

For more information about these attributes, see The RECEIVE command in *z/OS SMP/E Commands*.

Alternate software inventory format (IO22234)

The SMP/E RECEIVE ORDER command sends a request for PTFs or HOLDDATA, or both, to an Automated Service Request server. The server satisfies the request and returns information to SMP/E about a package to be downloaded that contains the requested PTFs or HOLDDATA, or both. To satisfy the request, the server must be given a software inventory that describes the existing installed software. The software inventory identifies the FMIDs and PTFs that are already installed. The server uses the inventory to determine which PTFs can be selected for the order, and to determine which PTFs are already installed and therefore need not be included in the order.

The SMP/E GIMXSID service routine is used as part of the ShopzSeries offering. Just as in the RECEIVE ORDER command processing, GIMXSID creates a software inventory required by ShopzSeries to place customized software product and service orders.

The software inventory produced by the SMP/E RECEIVE ORDER command and the SMP/E GIMXSID service routine restricts the PTFs described in the inventory to only those whose IDs follow the IBM naming convention. This means vendor software that uses PTF IDs that do not follow the IBM naming convention are not reflected in the software inventory. Therefore, the software inventory cannot be used by software vendors that want to provide their own Automated Service Request server.

The SMP/E RECEIVE ORDER command and the SMP/E GIMXSID service routine were modified to optionally create a new software inventory format that will describe all PTFs regardless of the form of the IDs. The decision whether or not to produce the software inventory in this new format is based on the following options:

- A new optional attribute on the <ORDERSERVER> tag in the ORDERSERVER data set for the RECEIVE ORDER command (`inventory="ibm | all"`)
- A new option on the PARM operand of the EXEC statement used to call the GIMXSID service routine (INVENTORY=IBM | ALL).

For the RECEIVE ORDER command:

- `inventory="ibm"` on the <ORDERSERVER> tag indicates the software inventory included in the request sent to the server will be produced in the form expected by the IBM Automated Service Request server. This form of the software inventory identifies all installed FMIDs and only PTFs whose IDs match the naming convention used by IBM. This is the default value.
- inventory="all" on the <ORDERSERVER> tag indicates the software inventory included in the request sent to the server will be produced in a generic form that identifies all installed FMIDs and PTFs regardless of their naming convention.

GIMUNZIP extensions (IO23270)

The GIMUNZIP service routine is used to extract the data sets, files, and directories from the archive files that compose a GIMZIP package. The data set, file, or directory into which the archive file is to be extracted may already exist, or GIMUNZIP can create a new data set, file or directory. Previously, new

sequential and partitioned data sets created by GIMUNZIP were always cataloged, and new data sets could not be created using specific SMS constructs.

GIMUNZIP now supports the following:

- The creation of new, uncataloged sequential or partitioned data sets
- The creation of new data sets using a specific device unit, storage class, management class or data class
- The identification of a volume and device unit for the allocation of temporary work data sets to be used when extracting partitioned or VSAM data sets.

To have GIMUNZIP create a new uncataloged data set, or create a new data set using a specific device unit, storage class, management class or data class, the following optional attributes may be included on the <ARCHDEF> tag in the GIMUNZIP SYSIN:

```
unit=unit-type
dataclas=data-class
mgmtclas=management-class
storclas=storage-class
catalog=YES | NO
```

To identify a volume and/or device unit for the allocation of temporary work data sets, the following optional tag may be included in the GIMUNZIP SYSIN:

```
<TEMPDS
volume=volser
unit=unit-type >
</TEMPDS>
```

See the GIMZIP service in the *z/OS SMP/E Reference* for further information about these new attributes and tags.

ZONEMERGE command extensions (IO23466)

The SMP/E ZONEMERGE command can be used to copy one zone into another (merge an existing zone into an empty zone) or merge two existing zones together. Previously, the ZONEMERGE command did no logical analysis of the SYSMOD, element, and LMOD content of the zones before attempting the merge operation. Thus the functional content of the merged zones might be incorrect.

The ZONEMERGE command supports a new CHECK operand, preserves CIFREQ subentries in the merged zone, and adds a SYSMOD validation phase to ensure the SYSMOD content of the merged zone will be functionally usable and correct. The following operands can be specified on the ZONEMERGE command:

- BYPASS(IFREQ)
- CHECK
- VERIFY(YES | NO)

For more information about the operands, see The ZONEMERGE command in *z/OS SMP/E Commands*.

8-character userids (IO24768)

SMP/E now fully supports the use of 8-character TSO/E userids.

Binder RMODE=64 and LONGPARM options (IO25475)

SMP/E recognizes the RMODE(64), RMODEX(NO), RMODEX(64TRUE) and LONGPARM link edit parameters in JCLIN input, and on the LEPARM operand of the++MOD statement. The parameters are saved in the LMOD and MOD entries in the target and dlib zones, and passed to the link edit utility whenever the LMOD or MOD is processed by that utility.

SMP/E V3R6 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R6. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

Multitasking using GIMDDALC SYSPRINT allocation

SMP/E processing is modified to support multitasking using SYSPRINT definition in the GIMDDALC data set.

Adding SAF checks to SMP/E processing (IO11698)

As an authorized program, SMP/E must ensure that any programs it calls that reside in an authorized library are called only in expected environments with expected parameters. SMP/E added SAF checks to its processing to ensure that only users with sufficient access authority are allowed to invoke certain SMP/E functions.

Cross Global Zone Reporting

The REPORT SYSMODS and REPORT CROSSZONE commands are enhanced to allow specifying target or DLIB zones that are defined in different global zones.

SYSMOD Comparison HOLDDATA Report

The REPORT SYSMODS command can be used during the deployment of a new release or maintenance level to identify missing SYSMODs. This command compares the SYSMOD content of two target or DLIB zones. A SYSMOD Comparison Report is generated identifying the SYSMODs that exist in the input zone, but are not found in the comparison zone.

The REPORT SYSMODS command is modified to identify SYSTEM and USER HOLDs that must be resolved before the SYSMODs identified in the SYSMOD Comparison Report can be installed in the comparison zone.

Retention of HOLDDATA (IO13643)

Before SMP/E 3.6, HOLDDATA was deleted from the global zone when the associated SYSMOD was deleted. Processing of the following commands was modified to retain HOLDDATA when the associated SYSMOD is deleted from the global zone:

- After the SYSMOD is successfully accepted.
- After the SYSMOD is successfully restored.
- During REJECT PURGE and REJECT mass mode processing.
- During REJECT NOFMID and REJECT select mode processing if the HOLDDATA operand is not specified.

SMP/E V3R5 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R5. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected

- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

Enhanced utility input

SMP/E V3R5 allows the specification of a standard UNIX file name as utility input on an INCLUDE statement found in a link-edit step of a JCLIN input stream. This increases the allowable length of the utility input name from 8 characters to 1023 characters. It also allows characters other than uppercase alphabetic (A-Z), numeric (0-9) and national (@,#,$). The utility input name can now contain any nonblank character X'41' through X'FE'.

Coexistence considerations

SMP/E releases before SMP/E V3R5 cannot process a JCLIN input stream containing an INCLUDE statement that has a utility input value that is longer than 8 characters or contains a character other than uppercase alphabetic (A-Z), numeric (0-9) and national (@,#,$). An error message is issued if an unsupported utility input value is encountered.

Additionally, unless the required PTF is installed, SMP/E releases before SMP/E V3R5 cannot process an LMOD entry containing an unsupported UTIN subentry. If installed, the PTF updates the SMP/E Query Dialogs and the GIMAPI to retrieve and display information about the UTIN subentries created with SMP/E V3R5 or higher. For the LIST and UNLOAD commands, if the zone contains an unsupported UTIN value, the PTF causes SMP/E to issue a warning message. For all other commands, if the zone contains an unsupported UTIN value, the PTF causes SMP/E to issue an error message.

Migration tasks

An application program that uses the GIMAPI to query the UTIN subentry of an LMOD entry might need to be updated to allow for the longer length for these UTIN values. Additionally, because the comma is used as the separator between the ddname and the file name in the UTIN value and because it is also a valid character in the UTIN file name, the application program might have to change the manner in which it extracts the ddname from the value.

Long SOURCEID support

SMP/E V3R5 allows SOURCEID values that can be up to 64 characters in length and that can contain any nonblank character (X'41' through X'FE') except a single quotation mark ('), an asterisk (*), a percent (%), a comma (,), a left parenthesis (), or a right parenthesis ()).

Coexistence considerations

SMP/E releases before V3R5 cannot process the MCS input that is developed specifically for V3R5, nor can they process data in the SMPCSI data set.

In SMP/E V3R4, if your SOURCEID value meets either of the following conditions, a message is shown to indicate that your operand contains a value that cannot be processed by the current release of SMP/E:

- Your SOURCEID value is 9 to 64 characters in length.
- Your SOURCEID value contains a character other than uppercase alphabetic (A-Z), numeric (0-9), or national (@,#,$).

ZONEMERGE command

SMP/E V3R5 has modified the ZONEMERGE command to automatically check for incompatible changes between the originating and destination zones before allowing the merge. This requires that a zone definition entry be present in the destination zone. If a zone entry is not present in the destination zone, or if it is present, but incompatible changes exist between the originating zone and the destination zone,

an error message is issued. Additionally, during CONTENT processing, the SREL subentry of the zone definition entry is merged.

Coexistence considerations

Unless the required PTF is installed, SMP/E releases before SMP/E V3R5 will not merge the SREL subentry of the zone definition entry, nor will they automatically check for incompatible changes between the originating and destination zones. If installed, the PTF causes SMP/E to merge the SREL data during CONTENT processing and to check for incompatible changes between the two zones.

Migration tasks

If the zone entry does not exist in the destination zone, it must be created using either UCLIN or the SMP/E dialogs. If the zone entry is present, but incompatible changes exist between the two zones, the user must run the UPGRADE command against the destination zone. The user must use the level of SMP/E that is equal to or higher than the UPGLEVEL subentry found in the zone entry of the originating zone. That is, if the UPGLEVEL subentry of the zone entry in the originating zone is 34.00, the UPGRADE command must be run against the destination zone using SMP/E 3.4 or higher.

HTTPS and FTP enhancements

- SMP/E uses HTTPS and FTP to communicate with servers for the RECEIVE ORDER and RECEIVE FROMNETWORK commands, or the GIMGTPKG service routine. As a result, SMP/E relies on an available and operational network, and operations might fail when network outages occur. SMP/E has added retry capabilities for HTTPS and FTP operations that fail because of an apparent network outage. This helps prevent SMP/E operations from failing because of short lived network outages.
- SMP/E allows any value that can be specified as a parameter for the FTP client to be specified during the RECEIVE ORDER command, RECEIVE FROMNETWORK command and GIMGTPKG service processing. You can specify an FTP.DATA file override (-f) parameter. You can specify the FTP client parameters in the FTPOPTIONS tag in the CLIENT or SMPCLNT data set.

HOLDDATA report changes

The following changes were made for HOLDDATA reports:

- A new report destination, SMPHRPT, was introduced to SMP/E. If the SMPHRPT ddname is allocated (either by DD statement or DDDEF entry), the HOLDDATA reports produced by the RECEIVE, APPLY, and ACCEPT commands are written to this ddname while other reports are written to the SMPRPT ddname. If the SMPHRPT ddname is not allocated, all reports are written to SMPRPT.
- The Bypassed HOLD Reason Report for the APPLY and ACCEPT commands includes only bypassed HOLDDATA for SYSMODs that are applied or accepted successfully. Bypassed HOLDDATA for SYSMODs that fail command processing will no longer appear in the report.
- In the APPLY and ACCEPT command HOLDDATA reports, suppressed reason IDs are consolidated and reported only once per reason ID per FMID. Formerly, reason IDs were reported once per SYSMOD. Reason IDs to be suppressed are identified by the SUPPHOLD subentry of the active OPTIONS entry.

BYPASS(HOLDSYS) message severity changes

During APPLY and ACCEPT command processing, SMP/E writes messages to identify SYSMOD HOLD conditions that were bypassed. In prior SMP/E releases, these messages had a severity of Warning and resulted in a minimum overall return code of 4 for the command. With SMP/E 3.5, Informational messages are written for bypassed SYSTEM HOLD conditions instead of Warning messages. This change to use Informational messages results in a minimum overall return code for the command of 0 instead of 4 when SYSTEM HOLDs are bypassed.

Migration tasks

To change APPLY and ACCEPT command processing to write Warning messages for bypassed SYSTEM HOLD conditions as was done in prior SMP/E releases, use the new COMPAT(WARNBYPASS) execution parameter for program GIMSMP. For example:

```
//SMPSTEP EXEC PGM=GIMSMP,PARM='COMPAT(WARNBYPASS)'
```

ZONEEDIT enhancement

The ZONEEDIT command now allows subentries to be added as well as changed. You can add the UNIT, VOLUME, and WAITFORDSN subentries of the DDDEF entry, and the PRINT subentry of the UTILITY entry using the ZONEEDIT command.

RECEIVE ORDER processing enhancements

- When an order results in an empty package, SMP/E resubmits the order to obtain the most recent HOLDDATA.
- When an order requests CONTENT PTFs and some of the requested PTFs cannot be found by the server, SMP/E resubmits the order to obtain those requested PTFs that can be found by the server.
- When an order requests CONTENT APARs and fixing PTFs cannot be found for some of the requested APARs, SMP/E resubmits the order to obtain the fixing PTFs that can be found.

Fix Category HOLDDATA

A new type of ++HOLD statement is used to identify SYSMODs that are associated with specific categories of fixes. Such categories identify support for new hardware devices and new releases of software, or enable selected new functions. The ++HOLD FIXCAT statement identifies those SYSMODs and their fix categories.

When FIXCAT HOLDs are received, the fix category values are assigned as source IDs to the SYSMODs that resolve the APARs. During APPLY and ACCEPT command processing, you can use the assigned source IDs to select the SYSMODs that are associated with a particular fix category.

In addition, a new REPORT MISSINGFIX command identifies fixes associated with particular fix categories that have not yet been installed and identifies whether any SYSMODs are available to satisfy those missing fixes.

SMP/E V3R4 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R4. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

Enhancement to the RECEIVE command

To support Internet Service Retrieval, the RECEIVE command was enhanced to enable you to request HOLDDATA or PTF orders directly from the IBM Automated Delivery Request server, automatically download the resulting service package and receive the HOLDDATA and PTFs contained in the service package.

Using the RECEIVE command with the new ORDER operand, you can request a new HOLDDATA or PTF order based on the content criteria you specify. SMP/E waits for the server to fulfill the order request. The

order request is fulfilled by building a package of PTFs and HOLDDATA that meets your content criteria. When the resultant package is ready, SMP/E downloads the package to the local z/OS host and performs traditional RECEIVE command operations on the contents of the package.

In addition to submitting new orders, the RECEIVE ORDER command can also be used to download orders that are in a pending state. If a requested PTF service order cannot be fulfilled within the allowed time, the RECEIVE command processing stops and SMP/E considers the order to be in the pending state. The server continues fulfilling the order, however. Use the RECEIVE ORDER command to check on the status of the service package, and retrieve the package when it is ready to be downloaded.

ORDERSERVER data set

The new ORDERSERVER data set provides the necessary information about the server that fulfills an SMP/E HOLDDATA or PTF order request.

CLIENT data set

The CLIENT data set was changed to include information about the local z/OS client system used for RECEIVE FROMNETWORK and RECEIVE ORDER command processing, such as navigating an FTP firewall and an HTTP proxy server.

Coexistence considerations

RECEIVE ORDER processing requires a new ORDERSERVER data set and new operands on the RECEIVE command. SMP/E releases before SMP/E V3R4 cannot process the new ORDERSERVER data set and do not recognize the new RECEIVE command operands.

Impacts to SMP/E zone entries

This release introduces a new entry type, ORDER and a new subentry type for OPTIONS entries, ORDERRET.

ORDER entry in the global zone

The ORDER entry is a new entry in the global zone to describe a HOLDDATA or PTF order initiated with the RECEIVE ORDER command. An ORDER entry is created in the global zone when SMP/E sends an order request to the IBM Automated Delivery Request server and the order is accepted by the server. This ORDER entry is used to record information about the order so SMP/E can query the server for status of orders that have not yet been completed.

ORDER RETENTION subentry on the OPTIONS entry

Indicates the retention period, in days, ORDER entries are kept in the global zone before being deleted. During RECEIVE ORDER command processing, the ORDER entry is deleted from the global zone if the ORDER RETENTION subentry value was exceeded.

When an ORDER entry is deleted from the global zone, SMP/E also deletes the order package stored in the SMPNTS.

Coexistence considerations

SMP/E releases before SMP/E V3R4 cannot process the new ORDER entry or the new ORDERRET subentry of the OPTIONS entry. A toleration PTF provides:

- LIST, GZONEMERGE, GIMAPI and Query Dialog support for the ORDERRET subentry and causes SMP/E to ignore the subentry in all other instances.
- GZONEMERGE, GIMAPI and Query Dialog support for the ORDER entry and causes SMP/E to issue a warning message if ORDER entries exist in the zone when the LIST command is requested. The ORDER entry is ignored in all other instances.

ICSF not required for GIMZIP and RECEIVE FROMNETWORK

The z/OS Integrated Cryptographic Services Facility (ICSF) is used by SMP/E to calculate SHA-1 hash values. These hash values are calculated for files within a GIMZIP package to verify the integrity of the data within the package. SMP/E was enhanced to use an alternate method to calculate SHA-1 hash values if ICSF is not available for use.

Although ICSF is the preferred method, SMP/E no longer requires it for use by the GIMZIP and GIMUNZIP service routines, nor for the RECEIVE FROMNETWORK or RECEIVE FROMNTS command. If SMP/E detects that ICSF is not available, SMP/E automatically uses an SMP/E Java application class to calculate SHA-1 hash values as an alternative. See "Options that affect Java" on page 100for the details of the required setup to allow SMP/E to use the Java application class.

Improved load module build processing

The load module build phase of the APPLY, RESTORE and LINK LMODS commands has been enhanced to be more tolerant of allocation errors for the distribution libraries. This accommodates distribution libraries which may be offline. In this case, SMP/E continues its search for a usable copy of the module instead of immediately failing because of the error allocating the module's distribution library.

SMP/E order management dialog

Use the new ORDER Management dialog to manage ORDER entries in the global zone. ORDER entries are created by using the RECEIVE ORDER command to request orders of HOLDDATA and PTFs from an IBM server. With the ORDER Management dialog, you can

- See the status of all orders at a glance.
- View the details of individual orders.
- Delete the ORDER entry for selected orders.

SMP/E query dialog

Existing panel GIMQU1PO now allows you to specify the ORDER entry.

SMP/E V3R3 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R3. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

GIMGTPKG service routine

The new GIMGTPKG service routine can be used to get GIMZIP packages from a remote FTP server in a TCP/IP network and store the package on a local z/OS host. GIMGTPKG performs the functions of the SMP/E RECEIVE FROMNETWORK TRANSFERONLY command, but does so independently of SMP/E. It uses FTP to transport the files of a GIMZIP package from a remote FTP server to a local host, thus providing:

- Industry standard FTP protocol.
- Secure transmission using the capabilities of the z/OS FTP client.
- Ensured integrity of the transported files.

Enhancements to GIMZIP and GIMUNZIP service routines

Formerly GIMZIP could create and GIMUNZIP could process packages that contained only sequential and partitioned data sets. Also, GIMUNZIP would only extract data from an archive file into a new data set allocated directly by GIMUNZIP. These service routines were enhanced for SMP/E V3R3, as follows:

- Packages can now contain VSAM data sets and UNIX files and directories.
- You can assign a unique ID to an archive during GIMZIP processing and then use that ID to identify the archive that is to be extracted during GIMUNZIP processing.
- GIMUNZIP now allows GIMUNZIP operations into existing data sets. GIMUNZIP determines if the data set specified on the <ARCHDEF> tag exists. If the data set exists, GIMUNZIP copies the archive file into the existing data set. If the data set does not exist, GIMUNZIP allocates a new data set and then copies the archive file into that new data set.

RECEIVE FROMNETWORK FTP interface enhancements

RECEIVE FROMNETWORK has been enhanced to:

- Allow user credentials and file data transferred between an FTP client and server to be secured with respect to encryption, authentication, and data integrity using the Transport Layer Security (TLS) enablement for FTP.
- Allow the z/OS FTP client to connect to FTP servers that reside beyond a firewall that runs a SOCKS server.
- Make IPv6 connectivity possible for both the FTP client and server.
- Use the FTP.DATA configuration data set to allow the client to specify local site parameters. The FTP.DATA configuration data set is optional, but must be used by the client to specify the parameters for TLS security and SOCKS firewall support.

Migration tasks

1. SMP/E will now use the FTP.DATA configuration data set to allow the client to specify local site parameters. Two of the values specified in the FTP.DATA data set are *FWFriendly* and *FTPKEEPALIVE*. These values correspond to the *pasv* and *keepalive* attributes in the CLIENT data set. Therefore, these attributes should no longer be specified in the CLIENT data set. If the *pasv* and *keepalive* attributes are specified in the CLIENT data set, they will be ignored. If required, these values must be specified in the FTP.DATA data set. Refer to *z/OS Communications Server: IP User's Guide and Commands* for more information about the statements that can be coded in the FTP.DATA data set.
2. To enable TLS security, SOCKS firewall support, and IPv6 addressing, ensure that z/OS Communications Server V1R2 (or higher) is installed.
3. If you previously specified FTP commands to navigate your local firewall using the <FIRECMD> tag of the CLIENT data set, then you may need to update them. Specifically, subcommands such as USER, PASS, and ACCT should no longer be specified in <FIRECMD>. The commands you specify in the <FIRECMD> tags should be the same as those you use with the z/OS Communications Server FTP client, and should contain only the actual values (or the appropriate substitution variables) for user ID, password, and account.

REJECT CHECK command

A CHECK function has been added to the REJECT command. CHECK indicates whether SMP/E should do a trial run of a command without actually updating any libraries. This is a way to test for errors that might occur during actual processing and to receive reports on the changes that would be made.

Extended RECEIVE SOURCEID processing

The RECEIVE command will now assign the source ID specified on the SOURCEID operand of the command to SYSMODs found in the SMPPTFIN input stream, even if the SYSMOD is already received.

SPCLCMOD and CMWA

SMP/E passes new default parameters (SPCLCMOD and CMWA=256K) to the copy utility when copying modules, load modules, or programs.

SMP/E V3R2 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R2. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

LINK LMODS command

The new LINK LMODS command can be used to refresh the callable services for all load modules within a particular target zone. The LINK LMODS command can also be used to refresh the callable services only for those load modules that have a dependency on a particular set of CALLLIBS. The LINK LMODS command replaces the REPORT CALLLIBS command, which has been removed from SMP/E V3R2. For more information about the LINK LMODS command, see *z/OS SMP/E Commands*.

REPORT CALLLIBS command removal

The REPORT CALLLIBS command has been removed from SMP/E V3R2. It has been replaced by the LINK LMODS command.

UPGRADE command

New releases of SMP/E must sometimes make changes to SMP/E data sets that cannot be properly processed by prior SMP/E releases. SMP/E usually makes incompatible changes only when necessary to provide new and improved capabilities. For example, a new type of element requires a new entry type in SMPCSI data sets and these new entry types are typically not understood or processed correctly by SMP/E levels that have not been specifically updated to do so.

The UPGRADE command allows you to specify when SMP/E is permitted to make incompatible changes to SMP/E data sets. This, in turn, allows you to make the trade-off between exploiting new SMP/E functions and preserving compatibility with prior SMP/E releases. For more information about the UPGRADE command, see *z/OS SMP/E Commands*.

Coexistence considerations

A toleration PTF will enable OS/390® V2R7 SMP/E, z/OS V1R2 SMP/E, and SMP/E V3R1 to automatically check for incompatible changes made by a higher level of SMP/E. A toleration PTF will also provide a warning message should a user try to issue an UPGRADE command on a release of SMP/E prior to V3R2.

GIMXSID service routine

The GIMXSID service routine is used as part of the ShopzSeries offering. GIMXSID creates a single data source required by ShopzSeries to place customized software product and service orders. The data source created by GIMXSID, the software inventory data, is a composite of three kinds of information as follows:

1. A list of FEATUREs found in the SMPCSI data sets. The Feature List is used by ShopzSeries to perform product requisite checking and also to prime the order checklist when ordering a ServerPac.

2. A list of the FMIDs found in the SMPCSI data sets. The FMID List is used by ShopzSeries to scope service orders to the PTFs applicable solely to the user's desired configuration of target and global zones.
3. A bitmap representation of the PTFs found in the specified target zones and global zones. The PTF Bitmap is used by ShopzSeries (CCSS) to produce service packages that do not contain PTFs that are already present in the user's configuration.

GIMZIP: Archive segmentation

The GIMZIP service routine has been enhanced with a new SEGMENT option to allow you to specify the maximum size for the network transportable objects produced by GIMZIP. Very large objects will be divided into archive segments that are within the specified size. The GIMUNZIP service routine and the RECEIVE FROMNETWORK and RECEIVE FROMNTS commands will now accept network packages that contain archive segments as input. For more information about the GIMZIP and GIMUNZIP service routines, see *z/OS SMP/E Reference*. For more information about the RECEIVE command, see *z/OS SMP/E Commands*.

Also, you can use the SMPWKDIR DD statement to specify a location for temporary files produced during GIMZIP processing.

Coexistence considerations

SMP/E releases prior to SMP/E V3R2 cannot process GIMZIP output that contains segmented archive files. A toleration PTF will issue an error message should a user try to process GIMZIP output that contains segmented archive files on a release of SMP/E prior to V3R2.

GIMZIP: User defined subdirectories

Users of GIMZIP may now specify subdirectories in which to store documentation, samples, readme files, and other files. This is done with the new `subdir` attribute of the <FILEDEF> tag of the GIMZIP package control statement. The subdirectory is created in a UNIX file system, within the parent directory pointed to by the SMPDIR DD statement in the JCL used to invoke GIMZIP.

Coexistence considerations

Unless the required PTF is installed, SMP/E releases prior to SMP/E V3R2 cannot process GIMZIP packages that exploit user-defined subdirectories.

Java archive files

Many z/OS software products developed with Java use Java Archive (JAR) files as the packaging format for Java application files. To better accommodate such products, SMP/E V3R2 is introducing JAR file update support. For SMP/E, there will be two forms of JAR files; JAR replacement files and JAR update files. JAR replacement files are complete replacements for a JAR file and are treated simply as files in the UNIX file system. SMP/E will copy replacement JAR files into the UNIX file system, just as is done for hierarchical file system elements. JAR update files are archive files themselves, but do not contain all of the component files that make up the complete JAR file. A JAR update file contains only the new and changed component files. These new and changed component files are archived into a JAR file. SMP/E uses the JAR command and the contents of a JAR update file to update an existing JAR file.

Coexistence considerations

SMP/E releases prior to SMP/E V3R2 cannot process JAR replacement files or JAR update files, nor can they process the data entries for these elements. A toleration PTF will cause SMP/E to issue an error message if it encounters an unsupported JAR element or data entry.

Smaller SMPLTS data set

The SMPLTS data set is used by SMP/E to save the base version of a product's load modules that use callable services. To reduce SMPLTS space requirements, SMP/E now saves a base version of a load module in the SMPLTS data set only if it contains both CALLLIBS and XZMOD subentries. If a load module contains CALLLIBS subentries, but no XZMOD subentries, this load module is not saved in the SMPLTS.

For more detailed information about these command changes, see *z/OS SMP/E Commands*.

Coexistence considerations

The SMPLTS data set used by SMP/E V3R2 may not contain the base version of load modules with CALLLIBS subentries. Because of this, once the SMPLTS has been modified by SMP/E V3R2, you cannot use certain commands from older levels of SMP/E that depend on the base version of a load module being in the SMPLTS data set. Specifically, if any of the following updates were made to a zone with load modules containing CALLLIBS but no XZMOD subentries, the target zone is marked:

- CLEANUP command is run against the zone
- GENERATE command is run against the zone
- APPLY or RESTORE command is run and the base version of the load module is deleted from the SMPLTS

A toleration PTF against older releases of SMP/E allows certain commands to be processed against a target zone that has been marked to indicate that the SMPLTS data set cannot be used. These commands do not depend on the base version of load modules existing in the SMPLTS:

- APPLY
- BUILDMCS
- CLEANUP
- DEBUG
- GENERATE
- JCLIN
- LIST
- LOG
- RESETRC
- SET
- UCLIN
- UNLOAD
- ZONECOPY
- ZONEDELETE
- ZONEEDIT
- ZONEEXPORT
- ZONEIMPORT
- ZONEMERGE

The toleration PTF also prohibits the RESTORE or LINK MODULE command from being run against a target zone that has been marked.

DUMMY data set for SYSDEFSD

The SYSDEFSD DD statement defines the location for the binder to write IMPORT statements for all exported entries of a DLL load module. Whenever entries are to be exported, the binder expects the SYSDEFSD data set and, if it is not present, will issue a warning message to indicate the missing data set. Product developers who supply DLLs do not always want or need the IMPORT statements associated

with a DLL retained. In these instances, the SYSDEFSD would be better defined as either a temporary or DUMMY data set.

SMP/E is defining a new ddname called SMPDUMMY, which will always be allocated as 'DD DUMMY'. Product packagers may now specify the SYSDEFSD DD statement in the JCLIN input stream as any of the following:

- //SYSDEFSD DD DSN=SMPDUMMY,DISP=xxx
- //SYSDEFSD DD DSN=NULLFILE
- //SYSDEFSD DD DUMMY

In each case, the SIDE DECK LIBRARY subentry of the LMOD entry will be set to SMPDUMMY. When needed for processing, SMPDUMMY will be dynamically allocated by SMP/E as a DUMMY data set.

Coexistence considerations

Unless the required PTF is installed, SMP/E releases prior to SMP/E V3R2 cannot process SYSMOD input that uses the SYSDEFSD DUMMY enhancement, nor can they process the data entries for these elements.

For more information about SYSDEFSD DUMMY, see *z/OS SMP/E Reference*.

SMP/E dialog customization

A new option, Option 0, has been added to the SMP/E Primary Option Menu GIM@PRIM to implement the current SMP/E customization options. This new option allows you to enter or change the values for the customization options that were previously found in panel GIM@UPRM. When you select option 0 from the GIM@PRIM panel, the panel GIM@PARM will appear. The options you then specify are saved permanently in the ISPF profile pool for later use by other SMP/E dialog processes.

Migration tasks

All dialog customization formerly specified on panel GIM@UPRM must now be specified using Option 0 on the SMP/E Primary Option Menu. When you move to a new release of SMP/E and continue to use the same ISPF profile data set, no migration actions are required to use the options previously entered and saved.

For more information about SMP/E dialog customization, refer to the tutorial panels that accompany the SMP/E dialogs.

GIMUTTBL removal

Module GIMUTTBL and load module GIMUTTBL are no longer supplied as part of SMP/E. Macro GIMDFUT, which was used to replace the IBM-supplied copy of GIMUTTBL, is also no longer supplied. GIMUTTBL was formerly used to specify which utility programs SMP/E can call.

Migration tasks

On a system with RACF as the security manager, you can control which utility programs SMP/E can call by defining the appropriate profiles in the PROGRAM class. Refer to *z/OS Security Server RACF Security Administrator's Guide* for more information about how to use this function.

SMP/E V3R1 overview

The following sections describe the new and changed SMP/E functions that are introduced for SMP/E V3R1. The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item

- References to other publications that contain additional detailed information

Defining exit routines using SMPPARM member GIMEXITS

SMP/E V3R1 allows you to define exit routines that are to be given control during SMP/E processing by specifying new GIMEXITS control statements in SMPPARM member GIMEXITS. This replaces the previous method of updating module GIMMPUXD. Putting the exit routine information in an SMPPARM member means that the information is persistent and that you do not need to update module GIMMPUXD every time a new release of SMP/E is installed.

Migration tasks

If you have existing exit routines defined in GIMMPUXD or wish to create new exit routines, you must define them in a GIMEXITS member. Any exit routines defined in GIMMPUXD will be ignored by SMP/E

For more detailed information about this change, see *z/OS SMP/E Reference*.

Dynamic allocation using SMPPARM member GIMDDALC

SMP/E V3R1 allows you to define data sets to be dynamically allocated by SMP/E by specifying new GIMDDALC control statements in SMPPARM member GIMDDALC, as well as by using DDDEF entries. This replaces the previous method of updating module GIMMPDFT. Putting the dynamic allocation information in an SMPPARM member means that the information is persistent and that you do not need to update module GIMMPDFT every time a new release of SMP/E is installed.

Migration tasks

If you have used GIMMPDFT to define data sets for dynamic allocation, you must create new definitions in GIMDDALC. SMP/E will ignore any definitions in GIMMPDFT.

For more information about this change, see *z/OS SMP/E Reference*.

Enhanced link name values

SMP/E V3R1 has increased the maximum length allowed for a hierarchical file system element LINK value from 64 characters to 1023 characters. SMP/E has also increased the maximum length allowed for the alias values on ALIAS link-edit control statements from 64 characters to 1023 characters.

Coexistence considerations

Unless the required PTF is installed, SMP/E releases prior to SMP/E V3R1 cannot process a SYSMOD containing a hierarchical file system element MCS that specifies a LINK value longer than 64 characters, nor can they process a hierarchical file system element entry containing a LINK subentry, or an LMOD entry containing an LMODALIAS subentry, created by SMP/E V3R1, or higher. If installed, the PTF will update the SMP/E Query dialogs and the GIMAPI to retrieve and display information about LINK or LMODALIAS subentries created by SMP/E V3R1, or higher. For all other SMP/E processing, the PTF will cause SMP/E to issue an error message if it encounters an unsupported LINK value or LINK or LMODALIAS subentry.

Migration tasks

An application program that uses the SMP/E Alias Record Type 0 (A0) library change record may need to be updated to handle alias and link values of up to 1023 characters to avoid truncating data. For more information about this change, see the chapter on Library Change File Records in *z/OS SMP/E Reference*.

An application program that uses the GIMAPI to query the LINK subentry of a hierarchical file system element entry or the LMODALIAS subentry of an LMOD entry may need to be updated to allow for the longer length of these subentries. For more information about this change, refer to the description of these subentries in the GIMAPI chapter in *z/OS SMP/E Reference*.

Removal of function to create backup IEANUC01 load modules

The ability to save the target system's nucleus load module (IEANUC01) during APPLY, LINK, and RESTORE command processing has been removed from z/OS V1R2 SMP/E. The NUCID operand of the APPLY command and the NUCID subentry are no longer supported and will be ignored by SMP/E if specified.

Migration tasks

An application program that uses the GIMAPI to query the NUCID subentry of an OPTIONS entry must be updated because this subentry no longer exists. For more information about the GIMAPI, refer to the GIMAPI chapter in *z/OS SMP/E Reference*.

Conditional JCLIN processing

SMP/E V3R1 allows a packager to use special JCL comments in the JCLIN input to cause SMP/E to skip over parts of the JCLIN input based on the installation environment. The parts of the JCLIN input that are skipped are not processed by the JCLIN command and do not contribute to the structure information derived from JCLIN processing.

For more information about this change, refer to the section on "Conditional JCLIN Comment Statements" in *z/OS SMP/E Commands*.

Coexistence considerations

Unless the required PTF is installed, SMP/E releases prior to SMP/E V3R1 cannot process JCLIN input that contains the special JCL comments used to skip over parts of the JCLIN input. If installed, the PTF will cause SMP/E to issue an error message if the unsupported JCL comments are encountered.

Network delivery of SMP/E input

SMP/E V3R1 can receive input from a network server, in addition to tape and DASD. This enables the delivery of SMP/E-installable products and service over the internet or an intranet. By installing software directly from a network source, SMP/E enables a more seamless integration of electronic software delivery and installation. This reduces the tasks and time required to install software delivered electronically.

SMP/E also provides the GIMZIP and GIMUNZIP service routines to construct, and then later unwrap, network transportable packages of software. This allows you to create your own packages of SMP/E installable software, and then distribute them within your own enterprise, or to other enterprises. Specifically, the GIMZIP service routine will accept partitioned or sequential data sets as input and will create a network transportable package as output. For more information about the GIMZIP and GIMUNZIP service routines, see *z/OS SMP/E Reference*.

Once a package is made accessible on an FTP server, you can use the SMP/E RECEIVE command to transfer the package through a TCP/IP network directly into an SMP/E environment. The RECEIVE command has been extended with new DELETEPKG, FROMNETWORK, and FROMNTS operands to process these network transportable packages. For more information about the RECEIVE command changes, see *z/OS SMP/E Commands*.

New CLIENT and SERVER data sets and SMPDIR and SMPNTS directories have been created to support this new processing. For more information about CLIENT, SERVER, SMPDIR, and SMPNTS, see *z/OS SMP/E Reference*.

Coexistence considerations

Network delivery of SMP/E input requires a new packaging format for that input and new operands on the RECEIVE command. SMP/E releases prior to z/OS SMP/E cannot process this new packaging format and do not recognize the new RECEIVE command operands.

AMODE=64 and COMPAT=PM4 link edit parameters

SMP/E V3R1 recognizes and saves the AMODE=64 and COMPAT=PM4 link edit parameters.

The AMODE option assigns the addressing mode for all of the entry points into a program module (the main entry point, its true aliases, and all of the alternate entry points). AMODE=64 instructs the binder to create AMODE 31/64 executables with 8-byte adcons.

The COMPAT option identifies the compatibility level of the binder.

Selected SMP/E data sets may now reside in a UNIX file system

SMP/E V3R1 allows the following data sets to reside in a UNIX file system:

- SMPCNTL
- SMPDEBUG
- SMPHOLD
- SMPJCLIN
- SMPLIST
- SMPOUT
- SMPPTFIN
- SMPPUNCH
- SMPRPT

For more information about this change, refer to the description of each of these data sets in *z/OS SMP/E Reference*.

HFS data set identification

SMP/E V3R1 has enhanced the SMP/E File Allocation Report and SMP/E library change file records to identify the physical HFS data sets where files in a UNIX file system reside.

SMPPTS spill data sets

SMP/E RECEIVE processing can use SMPPTS spill data sets, if defined, to store SYSMODs when the primary SMPPTS data set is full. Up to 99 spill data sets, named SMPPTS1 through SMPPTS99, can be defined with DD statements or DDDEFs. By eliminating the tasks involved when recovering from an overflowing SMPPTS data set, the use of SMPPTS spill data sets can reduce the amount of manual intervention and data set management required to install software service.

HOLDDATA summary reports

SMP/E V3R1 now provides additional HOLDDATA reports for APPLY and ACCEPT processing. The new reports provide you with ++HOLD information in the context of the APPLY or ACCEPT processing output. This frees you from having to manually collect this information, thus saving you significant research time.

SMP/E load modules and service routines moved to SYS1.MIGLIB

The SMP/E load modules, service routines, and other SMP/E components that formerly resided in the SYS1.LINKLIB library have been moved to SYS1.MIGLIB. Putting SMP/E into SYS1.MIGLIB enables a driving system to STEPLIB to SMP/E while ensuring that the STEPLIB does not expose the driving system to other executables that are not at the correct level for the driving system.

GIMXTRX service routine

GIMXTRX is intended for use as part of an offering called ShopzSeries. It provides two basic functions:

1. Generate a list of target zone names associated with a given GLOBAL zone SMPCSI data set name.

2. Generate a bitmap representation of FUNCTION and PTF SYSMODs found in a given list of target zone names associated with a given GLOBAL zone SMPCSI data set name.

OS/390 version 2 release 7 SMP/E overview

The following sections describe the new and changed SMP/E functions that are introduced for OS/390 Version 2 Release 7 (V2R7). (This level of SMP/E was also supplied with OS/390 Version 2, Releases 8, 9, and 10.) The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

SMP/E planning and migration assistant

OS/390 V2R7 (or later) SMP/E provided a Planning and Migration Assistant to assist users in managing their existing system and planning to migrate to a new OS/390 system.

Data element reformatting

SMP/E can install data elements during APPLY, ACCEPT, RESTORE, and GENERATE into the output data sets based on their physical attributes.

Coexistence considerations

Releases of SMP/E prior to OS/390 Version 2 Release 7 cannot process the DEIINST and HFSINST jobs created by the GENERATE command of SMP/E V3R1 or z/OS SMP/E. Also, an HFSINST job created by the GENERATE command from a release of SMP/E prior to OS/390 Release 7 cannot be processed by z/OS SMP/E or by OS/390 V2R7 (or later) SMP/E.

Rerun the GENERATE job using the SMP/E release that will be used to process the resulting JCL.

Description for a SYSMOD

OS/390 V2R7 (or later) SMP/E enabled product developers and packagers to include additional descriptive information in a SYSMOD header MCS (that is, in a ++APAR, ++FUNCTION, ++PTF, or ++USERMOD statement).

Improved protection for UNIX file system files

Before manipulating files in a UNIX file system, SMP/E temporarily switches the SMP/E user to superuser authority (UID=0) when manipulating files in the UNIX file system and restores the user to the previous level of authority when the SMP/E updates are done. This means that SMP/E users do not need to have UID=0 (superuser) authority all the time, which reduces the chance of such users accidentally erasing or damaging files in a UNIX file system while performing non-SMP/E work.

Coexistence considerations

The SMP/E user must be defined to the security class BPX.SUPERUSER for this process to work properly.

Pre-built load module support

The ++PROGRAM MCS can be used to add, replace, or delete pre-built load modules and program objects in PDS and PDSE data sets as complete entities in functions and PTFs.

Product data

The ++PRODUCT and ++FEATURE MCS can be used to add, replace, or delete additional product and feature data.

Sequential data set support

SMP/E can now install a data element into a sequential data set.

Coexistence considerations

Releases of SMP/E prior to OS/390 Version 2 Release 7 cannot process the DEIINST and HFSINST jobs created by the GENERATE command of SMP/E V3R1 or z/OS SMP/E. Also, an HFSINST job created by the GENERATE command from a release of SMP/E prior to OS/390 Version 2 Release 7 cannot be processed by z/OS SMP/E or by OS/390 V2R7 (or later) SMP/E.

Rerun the GENERATE job using the SMP/E release that will be used to process the resulting JCL.

Shell script support

OS/390 V2R7 (or later) SMP/E enabled the execution of UNIX shell scripts during SMP/E installation of code into the OS/390 UNIX Services environment. SMP/E provides a generic interface to enable a packager to deliver a shell script that can be executed during SMP/E installation, thus reducing the pre-install and post-install requirements of OS/390 UNIX Services application programs.

Symbolic link support

Symbolic links can be specified on hierarchical file system MCS.

OS/390 version 2 release 5 SMP/E overview

The following sections describe the new and changed SMP/E functions that are introduced for OS/390 Version 2 Release 5 (V2R5). (This level of SMP/E was also supplied with OS/390 Version 2 Release 6.) The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

CBIPO dialogs

The CBIPO installation dialogs formerly included with SMP/E were removed from SMP/E in OS/390 V2R5 SMP/E. Customers who want to install a CBIPO on an OS/390 system can still do so using bootstrap dialogs provided with the CBIPO.

Client code installation

OS/390 V2R5 (or later) SMP/E provides facilities to simplify the installation of cooperative or client/server products (such as OS/2). This is done with a common SMP/E packaging structure, a common S/390® server repository for client components, and a server repository accessible from any client platform. These facilities allow, for example, storing the client parts in a UNIX file system, which allows them to be packaged and installed along with the host parts, rather than separately.

Coexistence considerations

Enhanced hierarchical file system element MCS cannot be used by SMP/E releases before OS/390 V2R5 SMP/E, unless the required PTF is installed.

Global zone merge

OS/390 V2R5 (or later) SMP/E provides a method to merge information from one global zone into another global zone, which customers can use to reduce the number of global zones that they must manage. The merged information includes:

- SYSMOD and HOLDDATA entries,
- SYSMOD members in the SMPPTS data set,
- OPTIONS, UTILITY, DDDEF, ZONESET, and FMIDSET entries
- Global zone entry information, such as zone indexes, FMID list, and SRELs.

This facility is useful to customers who use ServerPac.

Library change interface

OS/390 V2R5 (or later) SMP/E provides a programming interface that can be used to obtain a synopsis of SMP/E APPLY and RESTORE processing at the library or member level. Customers can use this interface to propagate the libraries and members modified by SMP/E APPLY and RESTORE processing to other systems requiring the changes, thereby facilitating the integration of SMP/E-managed information in multisystem environments.

Coexistence considerations

OPTIONS entries that contain the CHANGEFILE subentry cannot be used by SMP/E releases prior to OS/390 V2R5 SMP/E, unless the required PTF is installed.

Improved load module build processing

OS/390 V2R5 (or later) SMP/E will not build a load module if SMP/E cannot include all of the load module's component modules that have been installed or are being installed. If such a load module cannot be completely built, SMP/E terminates APPLY processing for all affected SYSMODs. In addition, OS/390 V2R5 (or later) SMP/E reduces the likelihood of termination owing to incomplete load modules by expanding its search for the component modules to include copies of modules from within previously installed SYSMODs in the SMPPTS data set.

Load module return code

OS/390 V2R5 (or later) SMP/E allows product packagers to provide information in the JCLIN to identify the highest return code allowable for each load module. IBM will provide toleration PTFs for this function for prior releases of OS/390 and currently supported releases of SMP/E.

Coexistence considerations

LMOD entries that contain the RETURN CODE subentry cannot be used by SMP/E releases before OS/390 V2R5 SMP/E, unless the required PTF is installed.

Performance improvements

OS/390 V2R5 (or later) SMP/E provides for multitasking of link-edit operations during APPLY, ACCEPT, and RESTORE processing.

PTF compaction in SMPPTS data set

OS/390 V2R5 (or later) SMP/E compacts the SYSMOD (PTF) data within the SMPPTS data set to reduce its size. This ability to compact PTF data prior to release by IBM also allows IBM to shrink the size of the OS/390 service stream, thus reducing the amount of physical media (tapes) required to distribute service, as well as shrink the size of service distributed through various electronic mediums.

Coexistence considerations

- Compacted SMPPTS data created by OS/390 V2R5 (or later) SMP/E cannot be used by releases before OS/390 V2R5 SMP/E, unless the required PTF is installed.
- OPTIONS entries that contain the COMPACT subentry cannot be used by SMP/E releases before OS/390 V2R5 SMP/E, unless the required PTF is installed.

Enhanced RECEIVE command processing

OS/390 V2R5 (or later) SMP/E enables users to prevent the RECEIVE command from processing SYSMODs that are already applied or accepted. Users can specify this with the OPTIONS entry, on the RECEIVE command, or both. This enhancement reduces the need for the user to manually manage the SMPPTS with REJECT commands.

Coexistence considerations

OPTIONS entries that contain the RECZGRP and RECEXZGRP subentries cannot be used by SMP/E releases before OS/390 V2R5 SMP/E.

Reduced SMP/E message output

OS/390 V2R5 (or later) SMP/E reduced the number of messages issued during APPLY, ACCEPT, and RESTORE processing for easier identification of potential problems. Also, users may specify that messages issued to SMPOUT be formatted to an 80 character width, instead of the previous 120 character width, to make the messages easier to view when displayed on a terminal screen.

Coexistence considerations

OPTIONS entries that contain the MSGWIDTH and MSGFILTER subentries cannot be used by SMP/E releases before OS/390 V2R5 SMP/E.

GIMAPI: All entries and subentries support

For OS/390 V2R5 (or later) SMP/E, an application program using the GIMAPI QUERY command may specify an asterisk (*) on entry and subentry parameters to retrieve the Consolidated Software Inventory (CSI) data for all entry types, all subentries, or both.

GIMAPI: Version support

OS/390 V2R5 (or later) SMP/E can supply to an application program the version of GIMAPI being executed to retrieve information from the CSI. This allows the application program to determine whether information stored in the CSI is supported with the level of the QUERY program that is being executed.

Coexistence considerations

Application programs cannot use the VERSION command of the GIMAPI programming interface on releases of SMP/E prior to OS/390 V2R5 (or later) SMP/E, unless the required PTF is installed.

OS/390 version 1 release 3 SMP/E overview

The following sections describe the new and changed SMP/E functions that are introduced for OS/390 Version 1 Release 3 (V1R3). (This level of SMP/E was also supplied with OS/390 Version 2 Release 4.) The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

API for user access to the CSI

A programming interface is provided for read only access to SMP/E's consolidated software inventory (CSI) data. The data in CSI can be used to further automate systems management tasks.

A program called GIMAPI is used to invoke the API. The function can be called from different languages. Examples are provided for C/370 and PL/I.

The following commands are used with the GIMAPI call:

QUERY
Request data from the SMP/E CSI and return it to the calling program.

FREE
Free storage allocated by invocations of the QUERY command.

Enhanced cross-zone requisite checking

Cross-zone requisite checking is enhanced. Immediate feedback from the APPLY, ACCEPT, and RESTORE commands assists you in verifying that cross-zone requisites are installed and satisfied.

Optional parameters with these commands provide you the flexibility to:

- Override SMP/E's default method for determining which zones are checked for cross-zone requisites
- Install unsatisfied cross-zone requisites into the set-to zone
- Lessen the severity of a missing cross-zone requisite to a warning versus a terminating error

Coexistence considerations

Releases of SMP/E prior to OS/390 V1R3 SMP/E cannot perform the cross-zone requisite checking requested by the XZREQCHK(YES) subentry of a ZONESET entry and will ignore the request.

Enhanced exception SYSMOD report

The enhanced Exception SYSMOD Report, available as a small programming enhancement (SPE) for OS/390 V1R3 SMP/E, includes IBM OS/390 Enhanced HOLDDATA that is provided in ++HOLD statements. The report is reformatted so that it is ordered by FMID within each requested zone. (Previously, the report was ordered by SYSMOD within each zone.) A summary section is placed at the end of the report. The enhanced REPORT ERRSYSMODS command continues to work with legacy HOLDDATA.

The REPORT ERRSYSMODS command was enhanced by this SPE to handle held SYSMODs. Previously, a held, resolving SYSMOD was placed in the SMPPUNCH output, but was commented out. The customer had to rerun REPORT ERRSYSMODS command against the GLOBAL zone to determine if the held, resolving SYSMOD had an available resolving SYSMOD. REPORT ERRSYSMODS does this research for the customer and produces one SMPPUNCH output.

Coexistence considerations

- The REPORT ERRSYSMODS command in SMP/E releases before OS/390 V1R3 cannot display IBM z/OS Enhanced HOLDDATA as intended. OS/390 V1R3 and V2R4 require the installation of the appropriate SPE.
- The format of the HOLDDATA provided by the SMARTMVS service in Europe or the Electronic HOLDDATA service in the U.S. is not compatible with z/OS Enhanced HOLDDATA and does not take advantage of the enhanced REPORT ERRSYSMODS command. Customers who currently use these services, and who wish to make full use of the REPORT ERRSYSMODS command, must refresh their CSI's HOLDDATA with z/OS Enhanced HOLDDATA.

Enhanced ++IF FMID processing

z/OS SMP/E allows the ++IF MCS FMID operand to specify the same value as the value of the FMID operand of the previous ++VER MCS (if the SYSMOD is not a base function) or the name of the SYSMOD (if the SYSMOD is a base function).

Enhanced internal HOLD SYS processing

Analysis of internal HOLD information when one SYSMOD supersedes another is simplified. When a SYSMOD has ++HOLD information and it is superseded by another SYSMOD, the ++HOLD can be brought forward unchanged. The SYSMOD ID on the ++HOLD need not change to that of the superseding SYSMOD. Even if the SYSMOD ID on the ++HOLD is not the same as the containing SYSMOD, the ++HOLD is effective only against the SYSMOD that contains it. If the SYSMOD ID on the ++HOLD is not the same as the containing SYSMOD, SMP/E can determine if internal HOLDs are satisfied during APPLY and ACCEPT processing and thereby eliminate manual analysis.

Coexistence considerations

SYSMODs that contain a ++HOLD MCS that specifies a SYSMOD ID that is superseded on a preceding ++VER MCS cannot be processed by previous releases of SMP/E during RECEIVE processing, unless the appropriate PTF is installed for the prior release.

Enhanced ZONEEDIT command

The ZONEEDIT command is enhanced to provide a simplified method of changing path names. A PATH subentry is included on the unconditional CHANGE statement of the ZONEEDIT DDDEF command.

An example of when you might want to use the PATH subentry on the CHANGE statement is to modify path names of DDDEFs during the z/OS UNIX service process.

Enhancements to the binder utility in DFSMS/MVS

SMP/E can use enhancements to the binder utility in DFSMS/MVS. The enhancements to the binder include elimination of the LE/370 prelinker utility, and building dynamic load library (DLL) program objects. SMP/E's support includes:

- New link-edit parameters are recognized on the LEPARM operand of the ++MOD MCS and in JCLIN used to define a load module. The new parameters are ALIASES, DYNAM, FILL, HOBSET, REUS(NONE|REFR|RENT|SERIAL), RMODE=SPLIT, and UPCASE(YES|NO). All of these new parameters can be specified in JCLIN and all except ALIASES and DYNAM can be specified on the LEPARM operand.
- SMP/E supports the binder in dynamically building a definition side deck file for DLL program objects when those program objects are installed. The library to contain the definition side deck file is identified by a new side deck library (SIDEDECKLIB) subentry in the LMOD entry.
- Load modules that use DLLs can reference the definition side deck files associated with the DLLs. This is done by including the definition side deck files during a link-edit operation. The LMOD entry will contain a new utility input (UTIN) subentry list to record definition side deck files to be included during a link-edit operation.

Coexistence considerations

Releases of SMP/E before OS/390 V1R3 SMP/E cannot:

- Correctly install products and service that were developed for installation using the INCLUDE statements in JCLIN that identify UTILITY INPUT for a load module, the SYSDEFSD DD statements in JCLIN that identify the definition side deck library for a load module, and the FILL, HOBSET, RMODE=SPLIT, and EXITS link-edit attributes on the LEPARM operand on the ++MOD MCS and in JCLIN.
- Use target and distribution zones containing LMOD or MOD entries updated by OS/390 V1R3 (or later) with FILL, HOBSET, RMODE=SPLIT, EXITS, ALIASES, or DYNAM link-edit parameters in the MOD and LMOD entries, the UTILITY INPUT subentry list, or the SIDE DECK LIBRARY subentry in the LMOD entries.

System/390 service update facility

The System/390® Service Update Facility (SUF) available as a small programming enhancement (SPE) for OS/390 V1R3 SMP/E, provides, along with other System/390 products, provides a common tool across multiple platforms to help customers to maintain their systems with System/390 service facilities.

OS/390 version 1 release 2 SMP/E overview

The following sections describe the new and changed SMP/E functions that are introduced for OS/390 Version 1 Release 2 (V1R2). The information about each item includes:

- Description
- Summary of the SMP/E tasks or interfaces that might be affected
- Coexistence considerations, if any, that are associated with the item
- Migration procedures, if any, that are associated with the item
- References to other publications that contain additional detailed information

BLOCKSIZE=8800 for SMP/E data sets

For the RELFILE data sets, target libraries, and distribution libraries containing z/OS SMP/E, data sets that are allocated with RECFM=FB and LRECL=80 are allocated with BLKSIZE=8800.

BUILDMCS command

The BUILDMCS command provides a process for copying products from one pair of target and distribution zones and libraries, to another pair of target and distribution zones and libraries. This command generates the MCS and JCLIN required to reinstall the specified FMIDs.

Coexistence considerations

SYSMOD input (modification control statements) created by the BUILDMCS command cannot be processed by SMP/E releases prior to OS/390 V1R2 SMP/E, unless the required PTF is installed.

Bypassing system holds for specific SYSMODs

For APPLY and ACCEPT processing, you can bypass a particular system hold for specific SYSMODs, instead of for all SYSMODs held for that reason ID. For example, a number of SYSMODs might be held because they require you to take some required action before installing them. If you completed the required action for some (but not all) of the held SYSMODs, you can request SMP/E to bypass that hold reason ID only for the SYSMODs that you specify. All other SYSMODs affected by that reason ID remain held.

FMIDSET selection

SMP/E provides additional granularity of FMIDSET specification on the SELECT operand of the APPLY, ACCEPT, RESTORE, and RECEIVE commands to allow you to install sets of FMIDs.

Receiving relative file data sets created from PDSEs

When allocating a new SMPTLIB data set during RECEIVE processing, SMP/E checks the format of the associated relative file (RELFILE) data set, then uses the appropriate data set type (LIBRARY or PDS) for the SMPTLIB data set. Here are some benefits of this change:

- When packaging SYSMODs, you can ship program objects in RELFILEs, because SMP/E can load RELFILEs that were created from PDSEs into SMPTLIB data sets that are PDSEs.
- When receiving SYSMODs, you do not have to preallocate SMPTLIB data sets with the appropriate data set type, because SMP/E can allocate the SMPTLIB data set as PDS or LIBRARY, based on the format of the corresponding RELFILE data set.

SMP/E dialogs: FIND command

You can use the FIND primary command in the SMP/E dialogs. The FIND command makes it easier for you to quickly locate a specified character string in the table display section of panels in the following dialogs:

- SYSMOD Management
- Query
- Receive

Panels that allow the FIND command state that you can use the command. The help panels for these dialog panels explain how to use the FIND command.

SMP/E GIMOPCDE member moved from PARMLIB

The GIMOPCDE member, which SMP/E optionally uses to determine valid OPCODES during the scanning of JCLIN, has been removed from PARMLIB. Instead, a ready-to-use default set of OPCODE definitions is contained within SMP/E.

You may optionally provide an SMPPARM data set, which may contain your own OPCODE member to override the defaults supplied with SMP/E. The user-provided OPCODE member is a text member that you store in a user-allocated PDS named SMPPARM. You are not required to allocate the SMPPARM data set, unless you want to supply your own OPCODE member. If you provide an OPCODE member, it is used instead of SMP/E's default set.

SMP/E also provides a sample text member, named GIMOPCDE, that you can use as a starting point for creating your own OPCODE member.

Appendix B. Recommended service upgrade (RSU)

Recommended Service Upgrade (RSU) is a preventive service philosophy that applies to z/OS. RSU is intended to reduce the volume of PTFs customers must apply for preventive maintenance and to reduce the chance of encountering a PTF in error (PE), resulting in a more stable system.

IBM recommends that customers APPLY all RSU PTFs as preventive maintenance on their z/OS systems. However, customers must make the final decision as to what maintenance they will install.

The recommended service for the following products is tested in the Consolidated Service Test (CST) cycle:

- z/OS
- CICS Transaction Server for z/OS
- Db2® for z/OS
- IMS
- MQ for z/OS

Recommended Service Upgrades, with an SMP/E SOURCEID of RSU*yymm*, are available:

- Quarterly, with all PTFs that completed Consolidated Service Test (CST) testing during the prior quarter, including severity 1 through severity 4 APARs.
- Monthly, with high impact or pervasive (HIPER) PTFs, PTF-in-error (PE) PTFs, and other recommended PTFs (security or integrity fixes) that have been CST tested.

For information about the latest recommended level of service, see Consolidated Service Test and the RSU (www.ibm.com/support/pages/ibm-zos-consolidated-service-test-and-rsu).

Note: Although all CST-tested PTFs become RSU PTFs, not all RSU PTFs are tested in CST. Only the following systems or applications are included in the CST testing: z/OS, CICS, Db2, IMS, and MQSeries®.

The RSUyymm SOURCEIDs are provided in all z/OS product and PTF offerings, like Shopz and SMP/E RECEIVE ORDER.

© Copyright IBM Corp. 1986, 2026

Appendix C. Accessibility

Accessible publications for this product are offered through IBM Documentation for z/OS (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see How to Send Feedback to IBM to leave documentation feedback.

© Copyright IBM Corp. 1986, 2026

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

© Copyright IBM Corp. 1986, 2026

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Glossary

This glossary defines technical terms and abbreviations that are used in SMP/E documentation. If you do not find the term that you are looking for, refer to the index of the appropriate SMP/E manual.

Sequence of entries: For clarity and consistency of style, this glossary arranges the entries alphabetically on a letter-by-letter basis. In other words, only the letters of the alphabet are used to determine sequence; special characters and spaces between words are ignored.

Organization of entries: Each entry consists of a single-word or multiple-word term or the abbreviation or acronym for a term, followed by a commentary. A commentary includes one or more items (definitions or references) and is organized as follows:

1. An item number, if the commentary contains two or more items.
2. A usage label, indicating the area of application of the term, for example, "In programming," or "In SMP/E." Absence of a usage label implies that the term is generally applicable to SMP/E, to IBM, or to data processing.
3. A descriptive phrase, stating the basic meaning of the term. The descriptive phrase is assumed to be preceded by "the term is defined as ..." The part of speech being defined is indicated by the opening words of the descriptive phrase: "To ..." indicates a verb, and "Pertaining to ..." indicates a modifier. Any other wording indicates a noun or noun phrase.
4. Annotative sentences, providing additional or explanatory information.
5. References, directing the reader to other entries or items in the dictionary.

References: The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

See. This refers you to multiple-word terms that have the same last word.

See also. This refers the reader to related terms that have a related, but not synonymous, meaning.

Deprecated term for or **deprecated abbreviation for**. This indicates that the term or abbreviation should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

Selection of Terms: A term is a word or group of words to be defined. In this glossary, the singular form of the noun and the infinitive form of the verb are the terms most often selected to be defined. If the term may be abbreviated, the abbreviation is given in parentheses immediately following the term. The abbreviation is also defined in its proper place in the glossary.

A

ACCEPT
The SMP/E command used to install SYSMODs in the distribution libraries.

accept
In SMP/E, to install SYSMODs in the distribution libraries. This is done with the ACCEPT command.

accepted SYSMOD
A SYSMOD that has been successfully installed by the SMP/E ACCEPT command. Accepted SYSMODs do not have the ERROR flag set and are found as SYSMOD entries in the distribution zone.

Access method services (AMS)
The system utility program used to support VSAM data sets.

AMS
Access method services.

© Copyright IBM Corp. 1986, 2026

APAR
Authorized program analysis report.

APAR fix
A temporary correction of a defect in an IBM system control program or licensed program that affects a specific user. An APAR fix is usually replaced later by a permanent correction called a PTF. APAR fixes are identified to SMP/E by the ++APAR statement.

applied SYSMOD
A SYSMOD that has been successfully processed by the SMP/E APPLY command. Applied SYSMODs do not have the ERROR flag set and are found as SYSMOD entries in the target zone.

APPLY
The SMP/E command used to install SYSMODs in the target libraries.

apply
In SMP/E, to install SYSMODs in the target libraries. This is done with the APPLY command.

archive
An archive is a network transportable file containing two files; a file attribute file (FAF) and the data file that the FAF describes.

ASSEM entry
An SMP/E entry containing assembler statements that can be assembled to create an object module.

authorized program analysis report (APAR)
A report of a problem caused by a suspected defect in a current unaltered release of a program. The correction is called an APAR fix.

B

BACKUP entries
A collection of SMP/E target zone entries that are copied into the SMPSCDS data set during APPLY processing before they are updated by inline JCLIN, a ++MOVE MCS, or a ++RENAME MCS, or before they are deleted by an element MCS with the DELETE operand.

BACKUP entries consist of:

- A SYSMOD entry indicating what entries were, deleted, or updated
- ASSEM entries for updated target zone ASSEM entries
- LMOD entries for updated target zone LMOD entries
- MAC entries for updated or deleted target zone MAC entries
- MOD entries for updated or deleted target zone MOD entries
- SRC entries for updated or deleted target zone SRC entries
- Data element entries for deleted target zone data element entries
- DLIB entries for updated target zone DLIB entries

BALR
Branch and link register commands.

base function
A SYSMOD defining elements of the base system or other products that were not previously present in the target libraries. Base functions are identified to SMP/E by the ++FUNCTION statement. SMP/E is an example of a base function.

base level system
The level of the target system modules, macros, source, and DLIBs created by system generation, to which function and service modifications are applicable.

base version of a load module
Some load modules include modules both explicitly (through INCLUDE statements) and implicitly (through a SYSLIB allocation). The base version of such a load module includes only the explicitly defined modules for the load module. It is maintained by SMP/E in the SMPLTS data set. The base version of a load module is used with the SYSLIB allocation as input to the link-edit utility in order to build the load module in its target libraries.

binder
A program that processes the output of language translators and compilers into an executable program (load module). It is part of the DFSMS element of z/OS.

bypass
In SMP/E, to circumvent errors that would otherwise cause SYSMOD processing to fail. This is done by using the BYPASS operand on an SMP/E command.

C

causer SYSMOD
A SYSMOD identified by root cause analysis to be at the base of errors that caused other SYSMODs to fail. See *root cause analysis*.

CBPDO
Custom-Built Product Delivery Offering.

CICS
Customer Information Control System.

CLEANUP
The SMP/E command used to delete entries from the SMPMTS, SMPSTS, and SMPSCDS data sets after a SYSMOD has been accepted into the related distribution zone.

CNTL
See *SMPCNTL*.

coexisting functions
Functions that meet these requirements: (1) they are for the same system or subsystem and have the same SREL value, (2) they do not delete or supersede each other and are not negative prerequisites, and (3) if they are base functions, they are for different products. See also *conditionally coexisting functions* and *unconditionally coexisting functions*.

comment statements
Special control statements that are coded as JCL comments and which are used to convey information to SMP/E during JCLIN processing.

conditional requisites
Requisites defined by an ++IF statement. These are requisites that must be installed if the functions specified on the ++IF statements are installed.

conditionally coexisting functions
Functions that coexist but do not have to be in the same zone.

consolidated software inventory
See *SMPCSI*.

corequisite SYSMODs
SYSMODs each of which can be installed properly only if the other is present. Corequisites are defined by the REQ operand on the ++VER statement.

corrective service
Any SYSMOD used to selectively fix a system problem. Generally, corrective service refers to APAR fixes.

cross-zone
A target zone other than the set-to zone that defines a load module containing modules from set-to zone. Also called a TIEDTO zone. The modules were added to the load module through the SMP/E LINK command. The relationship between the cross-zone and the set-to zone is established through the TIEDTO subentry in their zone definition entries. See also *set-to zone* and *TIEDTO relationship*.

Pertaining to relationships between zones, especially as a result of conditional requisites (++IF statements) or LINK processing. See also *cross-zone requisite, cross-zone load module,* and *cross-zone module*.

cross-zone load module
A load module containing modules from a different zone as a result of LINK processing.

cross-zone module
A module included in a load module from a different zone as a result of LINK processing.

cross-zone requisite
A conditional requisite that must be installed in one zone because of another SYSMOD that is installed in a different zone. The REPORT command can be used to check information saved from ++IF statements and determine where any cross-zone requisites should be installed.

CSI
Consolidated software inventory data set. See *SMPCSI*.

D

data element
An element that is not a macro, module, or source—for example, a dialog panel or sample code.

DDDEF entry
An SMP/E entry containing the information SMP/E needs in order to dynamically allocate a particular data set.

DEBUG
The SMP/E command used to obtain additional information for problem determination—for example, to trace messages or take dumps.

debug
In SMP/E, to obtain additional information for problem determination—for example, to trace messages or take dumps. This is done with the DEBUG command.

definition sidedeck
A file in a UNIX file system, a member of a partitioned data set, or a sequential data set that contains binder IMPORT control statements.

deleted function
In SMP/E, a function that was removed from the system when another function was installed. This is indicated by the DELBY subentry in the SYSMOD entry for the deleted function. See also *explicitly deleted function* and *implicitly deleted function*.

deleting function
A function that removes other functions from the system. This is done by specifying them on the DELETE operand of the ++VER statement.

dependent function
A function that introduces new elements or redefines elements of the base level system or other products. A dependent function cannot exist without a base function. Dependent functions are identified to SMP/E by the ++FUNCTION statement.

dialog
The interactive support provided by SMP/E through ISPF. Instead of entering specific commands and operands, you can use panels to specify the requested processing.

distribution library (DLIB)
A library that contains the master copy of all the elements in a system. A distribution library can be used to create or back up a target library.

distribution zone
In SMP/E, a group of records in a CSI data set that describes the SYSMODs and elements in a distribution library.

DLIB
Distribution library.

DLIB entry
An SMP/E entry describing a distribution library that has been totally copied into a target library.

DLIBZONE entry
An SMP/E entry containing information used by SMP/E to process a specific distribution zone and the associated distribution libraries.

DLL
Dynamic link library

E

EC
Engineering change.

element
In SMP/E, part of a product, such as a macro, module, dialog panel, or sample code.

element MCS
An MCS that is used to replace or update an element.

element selection
The process by which SMP/E chooses the appropriate changes for an element that is affected by several SYSMODs being installed at the same time.

entry
In SMP/E, a collection of records in a CSI data set. An entry can be created, updated, or deleted by use of UCL statements.

environment
The functions (FMIDs) installed on a particular system or subsystem (SREL).

ERROR indicator
In SMP/E, an indicator in a target or distribution zone SYSMOD entry that shows that SYSMOD processing failed. The ERROR indicator is set before SMP/E updates any libraries and is reset if processing is successful. If processing fails, it remains set to show that an error occurred.

ESO
Expanded service options.

exception SYSMOD
A SYSMOD that is in error or that requires special processing before it can be installed. ++HOLD and ++RELEASE statements identify exception SYSMODs.

EXCP
Execute channel programs.

expanded service options (ESO)
A tape that includes preventive service PTFs. Where available, it replaces PUTs as the vehicle for delivering preventive service. An ESO contains PTFs and ++ASSIGN statements assigning source IDs for the PTFs. In the United States, this tape is available from the IBM Support Center and can be ordered either by subscription or as needed.

explicitly deleted function
A function that was deleted because it was specified on the DELETE operand of a ++VER statement in another SYSMOD.

exported zone
A zone that was copied into a sequential data set by use of the SMP/E ZONEEXPORT command.

external HOLDDATA
++HOLD statements that are contained in SMPHOLD. Contrast with *internal HOLDDATA*.

F

FAF
file attribute file.

FE
Field engineering.

feature
See *dependent function*.

file attribute file
A file attribute file (FAF) is a file that contains control statements that describe the attributes of the file contained in an SMP/E network transportable archive. The FAF is contained within the archive information about how the file was created.

File Transfer Protocol
File Transfer Protocol (FTP) is a protocol that defines the interactions necessary between a client and server to facilitate the exchange of binary and textual data files.

firewall
A firewall is an intermediate server that functions to isolate a secure network from an insecure network.

FTP
File Transfer Protocol.

FTP.DATA
Configuration data set used to change local site default values for the z/OS FTP Client.

FMID
Function modification identifier.

FMIDSET
A group of FMIDs to be used in processing an SMP/E command—for example, to indicate that SYSMODs applicable to certain functions should be installed.

FMIDSET entry
An SMP/E entry defining an FMIDSET.

function
In SMP/E, a product (such as a system component or licensed program) that can be installed in a user's system if desired. Functions are identified to SMP/E by the ++FUNCTION statement. Each function must have a unique FMID.

function modification identifier (FMID)
The SYSMOD ID of a function SYSMOD. It identifies the function that currently owns a given element.

functionally higher SYSMOD
A SYSMOD that uses the function that is contained in an earlier SYSMOD and contains additional functions as well. The earlier SYSMOD is called the *functionally lower SYSMOD*.

functionally lower SYSMOD
A SYSMOD whose function is also contained in a later SYSMOD. The later SYSMOD is called the *functionally higher SYSMOD*.

G

GENASM
A subentry in the MAC entry that lists the ASSEM or SRC entries that must be assembled if the macro is replaced or updated.

GENERATE
The SMP/E command used to create a job stream that builds a set of target libraries from a set of distribution libraries.

generate
In SMP/E, to create a job stream that builds a set of target libraries from a set of distribution libraries. This is done with the GENERATE command.

GIMUNZIP
An SMP/E service routine used to extract files contained in network transportable packages that were built using GIMZIP.

GIMZIP
An SMP/E service routine used to produce network transportable packages.

global zone
A group of records in a CSI data set used to record information about SYSMODs received for a particular system. The global zone also contains information that (1) enables SMP/E to access target and distribution zones in that system, and (2) enables you to tailor aspects of SMP/E processing.

GLOBALZONE entry
An SMP/E entry containing information that SMP/E uses to process the global zone, the associated target and distribution zones, and SMPPTS.

GTF
Generalized trace facility.

H

hashing
An operation that uses a one-way (irreversible) function on data, usually to reduce the length of the data and to provide a verifiable authentication value (checksum) for the hashed data.

header MCS
An ++APAR, ++FUNCTION, ++PTF, or ++USERMOD statement. The header MCS indicates the type of SYSMOD.

HFS
Hierarchical file system.

hierarchical file system element
An element that has a UNIX file system as its "target library".

hierarchy
In SMP/E, the top-down structure of function and service SYSMODs, in which each SYSMOD is dependent on the one above it.

higher functional level
An element version that contains all the functions of all other relevant versions of that element.

HOLDDATA
In SMP/E, MCSs used to indicate that certain SYSMODs contain errors or require special processing before they can be installed. ++HOLD and ++RELEASE statements are used to define HOLDDATA. SYSMODs affected by HOLDDATA are called exception SYSMODs.

HOLDDATA entry
An SMP/E entry containing ++HOLD statements that either were received from SMPHOLD (external HOLDDATA) or were within a SYSMOD that was received (internal HOLDDATA).

I

ICSF
Integrated Cryptographic Service Facility.

IFREQ
A conditional requisite. Conditional requisites are specified on the REQ operand of the ++IF statement.

IMASPZAP
The system utility program used to install superzaps, which are changes for modules, load modules, or CSECTs within modules.

implicitly deleted function
A function deleted because of its dependency on an explicitly deleted function that is specified on the DELETE operand of the ++VER statement.

imported zone
A zone copied from a sequential data set into another zone by use of the SMP/E ZONEIMPORT command.

IMS
Information Management System.

IMSGEN
IMS generation.

indicator
See *subentry indicator*.

in effect
Having control over SMP/E processing. For example, an OPTIONS entry is in effect if (1) it is specified on the SET command or (2) it is defined as the default OPTIONS entry for the set-to zone.

inline data
Information (such as utility control statements or code for an element) that is packaged directly after the associated MCS, rather than in a separate file or data set.

inline JCLIN
The JCL statements associated with a ++JCLIN statement. Inline JCLIN may immediately follow the ++JCLIN statement, or it may be in the RELFILE or TXLIB data set pointed to by the ++JCLIN statement. Inline JCLIN is used to update the target zone when a SYSMOD is applied, or the distribution zone when a SYSMOD is accepted. Contrast with *JCLIN input*.

inner macro
A macro invoked by another macro. In particular, inner macros are those that SMP/E does not detect during JCLIN processing of assembler job steps.

install
In SMP/E, to apply a SYSMOD to the target libraries or to accept a SYSMOD into the distribution libraries.

internal HOLDDATA
++HOLD statements contained within a SYSMOD. Contrast with *external HOLDDATA*.

I/O
Input or output.

IOGEN
Input/output device generation.

IPL
Initial program load.

IPv6
Internet Protocol Version 6.

ISMD
IBM Software Manufacturing and Delivery, which was formerly called the PID.

ISPF
Interactive System Productivity Facility.

ISPF/PDF
Interactive System Productivity Facility/Program Development Facility.

IVP
Installation verification procedure.

J

JAR
The SMP/E entry or MCS that describes a Java ARchive (JAR) file. It is the abbreviation for Java Archive. See *Java ARchive(JAR)*.

JARUPD
The SMP/E MCS that is used to describe an update to a JAR element.

JCL
Job control language.

JCLIN
The SMP/E command used to process data from SMPJCLIN.

The ++JCLIN statement, which is associated with JCLIN data that is included in a SYSMOD.

SMPJCLIN. See *SMPJCLIN*.

See also *inline JCLIN* and *JCLIN data*.

JCLIN data
The JCL statements that are associated with the ++JCLIN statement or saved in the SMPJCLIN data set. They are used by SMP/E to update the target zone when the SYSMOD is applied. Optionally, SMP/E can use JCLIN data to update the distribution zone when the SYSMOD is accepted.

JCLIN input
The JCL statements contained in SMPJCLIN and used as input for the JCLIN command. Contrast with *inline JCLIN*.

jar
The Java command used to invoke the Java Archive Tool. The Java Archive Tool is used to perform operations on Java ARchive (JAR) files.

Java ARchive (JAR)
An archive file format based on the ZIP file format. Used for aggregating many Java applet component files into one.

job control language (JCL)
A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

L

licensed program
A program that performs a function for the user, and usually interacts with and relies upon the system control program or some other IBM-provided control program. Generally, a licensed program is a software package that can be ordered from the program libraries, such as IBM Software Manufacturing and Delivery (ISMD). IMS and CICS are examples of licensed programs.

LINK LMODS
The SMP/E command used to relink load modules that use CALLLIBS.

LINK MODULE
The SMP/E command used to link modules in one zone with load modules in another zone.

link library (LKLIB)
A data set containing link-edited object modules.

LIST
The SMP/E command used to display entries in SMP/E data sets.

list
In SMP/E, to display entries in SMP/E data sets. This is done with the LIST command.

LKLIB
Link library.

LMOD
In SMP/E, an abbreviation for load module.

LMOD entry
An SMP/E entry containing all the information needed to replace or update a given load module.

load module
A computer program in a form suitable for loading into main storage for execution. It is usually the output of a link-edit utility.

LOG
The SMP/E command used to write user-supplied information to the SMPLOG data set.

The SMPLOG data set. See *SMPLOG*.

lower functional level
An element version that is contained in a later element version.

M

MAC
The SMP/E entry or MCS that describes a macro.

macro
An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language.

MACUPD
The SMP/E MCS used to update a macro.

mass-mode processing
In SMP/E, processing that includes all eligible SYSMODs, regardless of whether they were individually selected.

master CSI
The CSI data set that contains the global zone.

MCS
Modification control statement.

MCS entry
An SMP/E entry containing a copy of a SYSMOD exactly as it was received from SMPPTFIN. MCS entries are in SMPPTS, which is used to store SYSMODs.

MOD
The SMP/E entry or MCS that describes an object module or a single-module load module.

MODID
Modification identifier.

modification
In SMP/E, an alteration or correction to a system control program, licensed program, or user program. Synonymous with *system modification* (SYSMOD).

modification control statement (MCS)
An SMP/E control statement used to package a SYSMOD. MCSs describe the elements of a program and the relationships that program has with other programs that may be installed on the same system.

modification identifier (MODID)
A list of SYSMOD IDs, including the last SYSMOD that totally replaced the element (RMID), any subsequent partial updates to the element (UMIDs), and the function that owns the element (FMID). MODIDs are contained in element entries.

modification level
A distribution of all temporary fixes that have been issued since the previous modification level. A change in modification level does not add new functions or change the programming support category of the release to which it applies. Contrast with *release* and *version*.

Note: Whenever a new release of a program is shipped, the modification level is set to 0. When the release is reshipped with the accumulated services changes incorporated, the modification level is incremented by 1.

module
Synonym for *object module* or *single-module load module*.

MTS
Macro temporary storage data set. See *SMPMTS*.

MTSMAC entry
An SMP/E entry that is a copy of a macro that resides only in a distribution library but is needed temporarily during APPLY processing. MTSMAC entries are in the SMPMTS data set.

MVS
Multiple Virtual Storage.

MVS Custom-Built Product Delivery Offering (CBPDO)
A software delivery offering used to add products or service to an existing MVS, NCP, CICS, or IMS system.

N

NCP
Network Control Program.

negative prerequisite (NPRE)
In SMP/E, a function that is mutually exclusive with another function. It is defined by the NPRE operand on the ++VER statement.

NPRE
Negative prerequisite.

O

object deck
Object module input to the link-edit utility that is placed in the input stream, in card format.

object module
A module that is the output from a language translator (such as a compiler or an assembler). An object module is in relocatable format with machine code that is not executable. Before an object module can be executed, it must be processed by the link-edit utility.

When an object module is link-edited, a load module is created. Several modules can be link-edited together to create one load module (for example, as part of SMP/E APPLY processing), or an object module can be link-edited by itself to create a single-module load module (for example, to prepare the module for shipment in RELFILE format or in an LKLIB data set or as part of SMP/E ACCEPT processing).

operating system
In SMP/E, the system updated by APPLY and RESTORE processing. It consists of the target libraries. Also called the target system.

OPTIONS entry
An SMP/E entry defining processing options that are to be used by SMP/E.

P

package attribute file
A package attribute file (PAF) is a file that contains control statements describing the contents of a network transportable package.

packaging
Adding the appropriate MCS statements to elements to create a SYSMOD, then putting the SYSMOD in the proper format on the distribution medium, such as a tape or direct access data sets.

PAF
package attribute file.

partitioned data set extended (PDSE)
A system-managed data set containing an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PE
See *program error PTF*.

PE-PTF
See *program error PTF*.

PID
The former name for ISD.

PRE
Prerequisite.

prerequisite (PRE)
In SMP/E, a SYSMOD that must be installed before or along with another SYSMOD in order for that other SYSMOD to be successfully installed. It is defined by the PRE operand on the ++VER statement.

preventive service
The mass installation of PTFs to avoid rediscoveries of the APARs fixed by those PTFs.

The SYSMODs delivered on the program update tape.

product
Generally, a software package, such as a licensed program or a user application. A product can contain one or more functions and can consist of one or more versions and releases.

product version
All the releases for a given version of a product.

program error PTF (PE-PTF)
A PTF that has been found to contain an error. A PE-PTF is identified on a ++HOLD ERROR statement, along with the APAR that first reported the error.

program object
An executable program stored in a PDSE program library. It is similar to a load module, but has fewer restrictions. For SMP/E purposes, program objects are referred to as load modules.

program packaging
See *packaging*.

program product
The former term for licensed program.

program temporary fix (PTF)
A temporary solution or bypass of a problem that may affect all users and that was diagnosed as the result of a defect in a current unaltered release of the program. In the absence of a new release of a system or component that incorporates the correction, the fix is not temporary but is the permanent and official correction mechanism. New elements can also be defined in a PTF. PTFs are identified to SMP/E by the ++PTF statement.

program update tape (PUT)
The former vehicle for preventive service. See *expanded service options*.

PSW
Program status word.

PTF
Program temporary fix.

PTS
PTF temporary store data set. See *SMPPTS*.

PTFIN
PTF input. See *SMPPTFIN*.

PUT
See *expanded service options*.

R

RACF
Resource Access Control Facility.

RECEIVE
The SMP/E command used to read in SYSMODs and other data from SMPPTFIN and SMPHOLD.

receive
In SMP/E, to read SYSMODs and other data from SMPPTFIN and SMPHOLD and store them on the global zone for subsequent SMP/E processing. This is done with the RECEIVE command.

regressed SYSMOD
A SYSMOD one or more of whose elements are modified by subsequent SYSMODs that are not related to it.

regressing SYSMOD
A SYSMOD that causes regression of previous modifications when it is installed.

regression
In SMP/E, the condition that occurs when an element is changed by a SYSMOD that is not related to SYSMODs that previously modified the element.

REJECT
The SMP/E command used to remove SYSMODs from the global zone and SMPPTS.

reject
In SMP/E, to remove SYSMODs from the global zone and SMPPTS and delete any related SMPTLIB data sets. This is done with the REJECT command.

related installation materials (RIMs)
In IBM custom-built offerings, task-oriented documentation, jobs, sample exit routines, procedures, parameters, and examples developed by IBM.

related SYSMOD
A SYSMOD associated with other SYSMODs by the FMID, PRE, REQ, or SUP operands.

related zone
The zone that is named in the RELATED subentry of a TARGETZONE or DLIBZONE entry. For a target zone, the related zone is generally the distribution zone for the libraries used to create the target libraries. For a distribution zone, the related zone is generally the target zone for the libraries that are built from the distribution libraries.

relative file (RELFILE) format
A SYSMOD packaging method where elements and JCLIN data are in separate relative files from the MCSs. When SYSMODs are packaged in relative file format there is a file of MCSs for one or more SYSMODs, and one or more relative files containing unloaded source-code data sets and unloaded link-edited data sets containing executable modules. The relative files can be either unloaded files in IEBCOPY format, or they can be partitioned data sets. Relative file format is the typical method used for packaging function SYSMODs.

relative files (RELFILEs)
Unloaded files containing modification text and JCL input data associated with a SYSMOD. These files are used to package a SYSMOD in relative file format.

release
A distribution of a new product or new function and APAR fixes for an existing product. Contrast with *modification level* and *version*.

replacement modification identifier (RMID)
The SYSMOD ID of the last SYSMOD that completely replaced a given element.

REPORT
The SMP/E command used to obtain information about SYSMODs that have been installed. These are the types of REPORT commands:

- REPORT CROSSZONE: Lists conditional requisites that must be installed in certain zones because of SYSMODs installed in other zones.
- REPORT ERRSYSMODS: Determines whether any SYSMODs already installed are now exception SYSMODs.
- REPORT SOURCEID: Lists the source IDs associated with SYSMODs in the specified zones.
- REPORT SYSMODS: Compares the SYSMODs installed in two target or distribution zones.

requisite
A SYSMOD that must be installed before or at the same time as the SYSMOD being processed. There are several types of requisites:

- Prerequisites, which are specified by the PRE operand on the SYSMOD's ++VER statement
- Corequisites, which are specified by the REQ operand on the ++VER statement for the SYSMOD
- Conditional requisites, which are specified by the REQ operand on the SYSMOD that is associated ++IF statement

requisite chain
A sequence of SYSMODs that are directly or indirectly identified as requisites for a given SYSMOD, (A SYSMOD may identify other SYSMODs as requisites, which in turn may have requisites of their own. The requisite chain extends from the initial SYSMOD, through the hierarchy of requisites, until no more SYSMODs are found that have requisites.) See *requisite*.

requisite set
The set of all SYSMODs on the requisite chain for a particular SYS

RESETRC
The SMP/E command used to set the return codes for the previous commands to zero, so that SMP/E can process the current command.

RESTORE
The SMP/E command used to remove applied SYSMODs from the target libraries.

restore
In SMP/E, to remove applied SYSMODs from the target libraries by use of the RESTORE command.

restore group
All the SYSMODs that have a direct or indirect relationship with a SYSMOD being restored by use of the GROUP operand.

RIM
Related installation material.

RMID
Replacement modification identifier.

RMF
Resource measurement facility.

root cause analysis
Processing done by SMP/E for the ACCEPT, APPLY, and RESTORE commands to identify causer SYSMODs (SYSMODs whose failure has led to the failure of other SYSMODs). The types of errors SMP/E analyzes to determine causer SYSMODs include the following:

- Held SYSMODs
- Missing requisite SYSMODs
- Utility program failures: copy, update, assembler, link, zap
- Out-of-space conditions: *x*37 abends
- Missing DD statements and other allocation errors
- ID errors (a SYSMOD does not supersede or specify as a prerequisite an RMID or a UMID)
- JCLIN failures (syntax errors)

RPL
Request parameter list.

RTM2WA
Recovery termination manager 2 work area.

S

SCDS
Save control data set. See *SMPSCDS*.

SCP
System control program.

select-mode processing
In SMP/E, processing that includes individually selected SYSMODs.

service
PTFs and APAR fixes.

service level
The FMID, RMID, and UMID values for an element. The service level identifies the owner of the element, the last SYSMOD to replace the element, and all the SYSMODs that have updated the element since it was last replaced.

service order relationship
A relationship among service SYSMODs that is determined by the PRE and SUP operands, and the type of SYSMOD.

service SYSMOD
Any SYSMOD identified by an ++APAR or ++PTF statement.

service update
The integration of available service into the current release of a function. Since this is not a new release of the function, it does not change the function's FMID.

SET
The SMP/E command used to indicate the zone to be processed.

set
In SMP/E, to indicate which zone should be processed by the subsequent commands. This is done with the SET command.

set-to zone
The zone that was specified on the previous SET command and that is currently being processed. Contrast with *cross-zone*.

SHA-1
Secure Hash Algorithm 1.

Sidedeck
See *definition sidedeck*.

single-module load module
A load module created by link-editing a single object module by itself—for example, to prepare the module for shipment in RELFILE format or in an LKLIB data set or as part of SMP/E ACCEPT processing.

SMPCNTL
The SMP/E data set or file in a UNIX file system that contains the SMP/E commands to be processed.

SMPCSI
The SMP/E data set that contains information about the structure of a user's system as well as information needed to install the operating system on a user's system. The SMPCSI DD statement refers specifically to the CSI that contains the global zone. This is also called the master CSI.

SMPDEBUG
The SMP/E data set or file in a UNIX file system that contains a dump requested by the DEBUG command. Depending on the operands specified, it may contain (1) a dump of SMP/E control blocks and storage areas associated with the specified dump points or (2) a dump of the VSAM RPL control block for the specified SMP/E function.

SMPDUMMY
The SMP/E data set used to define a load module's definition side deck library as a DUMMY data set. SMPDUMMY is always allocated by SMP/E as a DUMMY data set.

SMP/E
A program product, or an element of OS/390 or z/OS, used to install software and software changes on z/OS systems. SMP/E consolidates installation data, allows more flexibility in selecting changes to be installed, provides a dialog interface, and supports dynamic allocation of data sets.

SMP/E commands
Commands defining the processing to be done by SMP/E, such as RECEIVE.

SMP/E entry
An entry in an SMP/E data set—for example, a MOD entry in a CSI data set.

SMPHOLD
SMPHOLD is the source for HOLDDATA (++HOLD and ++RELEASE statements) to be processed by the RECEIVE command. SMPHOLD may be a tape file, a data set, or one or more files in a UNIX file system.

SMPHRPT
The alternate SMP/E report data set. If SMPHRPT is allocated, the HOLD reports generated during RECEIVE, APPLY, and ACCEPT processing are directed to the SMPHRPT data set while other reports are directed to SMPRPT.

SMPJCLIN
The SMP/E data set or file in a UNIX file system that contains a job stream of assembly, link-edit, and copy job steps. This data is typically the stage 1 output from the most recent full or partial system generation. However, it may also be other data in a similar format, such as the output of the GENERATE command. This job stream is used as input to the JCLIN command to update or create entries in a target zone.

SMPLIST
The SMP/E data set or file in a UNIX file system that contains the output of all LIST commands.

SMPLOG
The SMP/E data set that contains time-stamped records of SMP/E processing. The records in this data set can be written automatically by SMP/E or added by the user through the LOG command.

SMPLOGA
A secondary log data set for SMP/E processing. If SMPLOGA is defined, it is automatically used when the SMPLOG data set is full.

SMPLTS
The SMP/E data set used as a target load module library to maintain the base version of a load module that specifies a SYSLIB allocation in order to implicitly include modules.

SMPMTS
The SMP/E data set used as a target library for macros that exist only in a distribution library, such as macros in SYS1.AMODGEN. The SMPMTS enables the current version of these macros to be used for assemblies during APPLY processing.

SMPNTS
The SMPNTS is a directory structure and associated files contained in a UNIX file system used for temporary storage of network transported packages that were received during SMP/E RECEIVE processing.

SMPOBJ
The SMP/E data set used for source-maintained products. SMPOBJ contains preassembled modules that can be used to avoid reassembling those modules. These modules must be in load module format —that is, in the same format as modules residing in the distribution library.

SMPOUT
The SMP/E data set or file in a UNIX file system that contains messages issued during SMP/E processing. It might also contain a dump of the VSAM RPL, if a dump was taken. In addition, it might contain LIST output and reports if SMPHRPT, SMPLIST, and SMPRPT are not defined.

SMPPARM
The data set that contains members to define parameters, such as macros, assembler operation codes, GIMDDALC control statements, and exit routines.

SMPPTFIN
SMPPTFIN is the source of SYSMODs and ++ASSIGN statements to be processed by the RECEIVE command. SMPPTFIN may be a tape file, a data set, or one or more files in a UNIX file system.

SMPPTS
The SMP/E data set that contains SYSMODs received from SMPPTFIN. SMPPTS is the source of SYSMODs that are installed in the target and distribution libraries.

SMPPTS spill data sets
Optional SMP/E data sets that can be used to store SYSMODs when the SMPPTS data set becomes full.

SMPPUNCH
The SMP/E data set or file in a UNIX file system that contains output from various SMP/E commands. This output generally consists of commands or control statements.

- GENERATE: A job stream for building target libraries
- REPORT: Commands for installing or listing SYSMODs
- UNLOAD: UCLIN statements for re-creating the entries that were unloaded

SMPRPT
The SMP/E data set or file in a UNIX file system that contains the reports that are produced during SMP/E processing.

SMPSCDS
The SMP/E data set that contains backup copies of target zone entries that are created during APPLY processing. These backup copies are made before the entries are (1) changed by inline JCLIN, a ++MOVE MCS, or a ++RENAME MCS, or (2) deleted by an element MCS with the DELETE operand. The backup copies are used during RESTORE processing to return the entries to the way they were before APPLY processing.

SMPSNAP
The SMP/E data set that is used for snap dump output. When a severe error such as an abend or severe VSAM return code occurs, SMP/E requests a snap dump of its storage before doing any error recovery. In addition, the DEBUG command can request a snap dump of SMP/E storage when specified messages are issued, or can request a snap dump of control blocks and storage areas associated with a specified dump point.

SMPSTS
The SMP/E data set used as a target library for source that exists only in a distribution library. The SMPSTS enables the current version of this source to be used for assemblies during APPLY processing.

SMPTLIB
The SMP/E data sets used as temporary storage for relative files loaded from SMPPTFIN during RECEIVE processing. The SMPTLIB data sets are deleted when the associated SYSMOD is deleted by REJECT, RESTORE, or ACCEPT processing.

SMPWKDIR
An optional directory in a UNIX file system used for temporary work files.

SMPWRK1
The SMP/E data set used as temporary storage for macro updates and replacements that will be processed by an update or copy utility program. SMP/E places the input in SMPWRK1 during APPLY and ACCEPT processing before calling the utility.

SMPWRK2
The SMP/E data set used as temporary storage for source updates and source replacements that will be processed by an update or copy utility program. SMP/E places the input in SMPWRK2 during APPLY and ACCEPT processing before calling the utility.

SMPWRK3
The SMP/E data set used as temporary storage for object modules supplied by a SYSMOD, object modules created by assemblies, and zap utility input following ++ZAP statements.

SMPWRK4
The SMP/E data set used as temporary storage for zap utility or link-edit utility input that contains EXPAND control statements.

SMPWRK6
The SMP/E data set used during ACCEPT and APPLY processing as temporary storage for inline replacements for data elements. SMP/E places the input in this data set so that it can be directly accessed and installed by the copy utility or SMP/E.

source
The source statements that constitute the input to a language translator for a particular translation.

source ID
A 1- to 8-character identifier that indicates how a SYSMOD was obtained—for example, from a particular service level in an ESO. A source ID is associated with a specific SYSMOD by the RECEIVE command or the ++ASSIGN statement.

SOCKS
A networking proxy protocol that enables hosts on one side of a SOCKS server to gain full access to hosts on the other side of the SOCKS server without requiring direct IP-reachability.

SOURCEID
The operand used to refer to a source ID.

source module
The source statements that constitute the input to a language translator, such as a compiler or an assembler, for a particular translation.

SRC
The SMP/E entry or MCS statement that describes a source.

SRCUPD
The MCS used to update a source.

SREL
System release identifier.

Storage Management Subsystem (SMS)
A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

STS
Source temporary store data set. See *SMPSTS*.

STSSRC entry
An SMP/E entry that is a copy of source that resides only in a distribution library but is needed temporarily during APPLY processing. STSSRC entries are in the SMPSTS data set.

stub entry
An element entry or LMOD entry that does not contain the basic information SMP/E requires in order to process the element or load module (such as FMID, RMID, or library names), but does contain other information, such as subentries describing cross-zone relationships.

stub load module
A load module that does not contain the modules needed to perform its basic functions, but does contain other modules, such as cross-zone modules.

subentry
A field in an SMP/E entry. Each subentry has associated with it a type and a value. The same subentry type may occur several times in a single entry, each time with a different value. For example, the modules supplied by a PTF are saved as MOD subentries in the PTF's SYSMOD entry. Some subentries occur only once within an entry, such as the FMID subentry in a target zone MOD entry.

subentry indicator
A subentry that does not have a data value associated with it. An example of an indicator is the ERROR indicator in the SYSMOD entry. An indicator is either on or off.

subentry list
Multiple occurrences of the same subentry type in an entry, each with a different value. For example, the modules supplied by a PTF are saved as names in the MOD subentry list within the SYSMOD entry for that PTF.

SUP
Supersede.

superseded-only SYSMOD
A SYSMOD that has not been installed, but that has been superseded by another SYSMOD that has been installed.

superseded SYSMOD
In SMP/E, a SYSMOD that is contained in or replaced by the SYSMOD or requisite set of SYSMODs currently being processed. This is indicated by the SUPBY subentry in the SYSMOD entry for the superseded SYSMOD. A superseded SYSMOD is functionally lower than the SYSMOD that superseded it.

superseding SYSMOD
In SMP/E, a SYSMOD that contains all the functions in another SYSMOD and is recognized as the equivalent of that other SYSMOD. The superseding SYSMOD uses SUP operand on its ++VER statement to specify the superseded SYSMOD.

superzap
A generic term for the process performed by IMASPZAP. It can also refer to the module updates processed by IMASPZAP.

SVC
Supervised call.

SVRB
Supervisor request block.

SYSGEN
System generation.

SYSLIB
A subentry used to identify the target library in which an element is installed.

A concatenation of macro libraries to be used by the assembler.

A set of routines used by the link-edit utility to resolve unresolved external references.

SYSMOD
System modification.

SYSMOD entry
An SMP/E entry containing information about a SYSMOD that has been received into SMPPTS, accepted into the distribution libraries, or applied to the target libraries.

SYSMOD ID
System modification identifier.

SYSMOD packaging
See *packaging*.

SYSMOD selection
The process of determining which SYSMODs are eligible to be processed.

SYSPRINT
The data set that contains output from the utilities called by SMP/E.

SYSPUNCH
The temporary data set containing object modules assembled by running the job stream produced by system generation or the GENERATE command. These modules are not installed in the distribution libraries at ACCEPT time.

system control program (SCP)
IBM-supplied programming that is fundamental to the operation and maintenance of the system. It serves as an interface with licensed programs and user programs and is available without additional charge.

system generation (SYSGEN)
The process of selecting optional parts of an operating system and of creating a particular operating system tailored to the requirements of a data processing installation.

system modification (SYSMOD)
The input data to SMP/E that defines the introduction, replacement, or update of elements in the operating system and associated distribution libraries to be installed under the control of SMP/E. A system modification is defined by a set of MCS.

system modification identifier (SYSMOD ID)
The name that SMP/E associates with a system modification. It is specified on the ++APAR, ++FUNCTION, ++PTF, or ++USERMOD statement.

system release identifier (SREL)
A 4-byte value representing the system or subsystem, such as Z038 for MVS-based products.

SYSUT1, SYSUT2, SYSUT3
Scratch data sets for SMP/E and the utilities it calls.

SYSUT4
A data set that is used instead of the SYSIN data sets when certain utilities are called.

T

target library
A library containing the executable code that makes up a system.

target system
The system updated during APPLY and RESTORE processing, also referred to as the operating system. See also target libraries.

target zone
In SMP/E, a group of records in a CSI data set that describes the SYSMODs, elements, and load modules in a target library.

TARGETZONE entry
An SMP/E entry containing information used by SMP/E to process a specific target zone and the associated target libraries.

TCP/IP
Transmission Control Protocol/Internet Protocol (TCP/IP) is a hardware independent communication protocol used between physically separated computers. It was designed to facilitate communication between computers located on different physical networks.

temporary data set
A work data set (SMPWRK1–SMPWRK6) or utility data set (SYSUT1–SYSUT4). Temporary data sets are allocated when processing for an SMP/E command begins, and deleted when processing is finished.

text library (TXLIB)
A data set containing JCLIN input or replacements for macros, source, or object modules that have not been link-edited. It is used when the JCLIN or elements are provided in partitioned data sets rather than inline or in relative files.

TGTLIB
Target library.

TIEDTO relationship
A cross-zone relationship between two target zones created when the LINK command updates a load module in one of the zones to include modules from the other zone. This relationship is established through the TIEDTO subentry in the zone definition entries for each of the zones.

TIEDTO zone
See *cross-zone*.

TLIB
Temporary library. See *SMPTLIB*.

transformed data
Data processed by the GIMDTS service routine so that it can be packaged inline in fixed-block 80 records.

Transport Layer Security (TLS)
A protocol that provides communications privacy over the Internet.

TSO
Time-sharing option.

TXLIB
Text library.

U

UCL
Update control language.

UCL statement
An SMP/E control statement used to define or change information in an SMP/E data set entry. UCL statements are coded between the UCLIN and ENDUCL commands. The UCL statement specifies the action to be taken (ADD, REP, or DEL), the entry to be modified, and any indicators and subentries to be changed.

UCLIN
The SMP/E command used to mark the beginning of UCL statements, which are used to make changes to entries in SMP/E data sets.

UMID
Update modification identifier.

unconditionally coexisting functions
Functions that coexist and must be in the same zone.

UNLOAD
The SMP/E command used to copy data out of SMP/E data set entries in the form of UCL statements.

unload
In SMP/E, to copy data out of SMP/E data set entries in the form of UCL statements, by use of the UNLOAD command.

update
In SMP/E, to change an existing element without replacing it.

update modification identifier (UMID)
The SYSMOD ID of a SYSMOD that updated the last replacement of a given element.

user modification (USERMOD)
A change constructed by a user to modify an existing function, add to an existing function, or add a user-defined function. USERMODs are identified to SMP/E by the ++USERMOD statement.

USERMOD
User modification.

UTILITY entry
An SMP/E entry containing information used by SMP/E to invoke a particular system utility program.

V

VERSION
An operand on the ++VER or element statement. VERSION specifies one or more SYSMODs containing elements that are functionally lower than elements in the SYSMOD that specifies the operand. The VERSION operand is also used to change ownership of elements.

version
A separate licensed program that is based on an existing licensed program and that usually has significant new code or new functions. Contrast with *release* and *modification level*.

versioned element
An element that is part of more than one function—for example, one that is part of a base function and a dependent function.

VSAM
Virtual Storage Access Method.

VTOC
Volume table of contents.

Z

ZAP
The SMP/E MCS used to package an update for an object module.

The superzap control statement used to update an object module.

A shortened name for the superzap utility, which is used to install these updates. See *IMASPZAP*.

zone
A partition in a CSI data set.

ZONECOPY
The SMP/E command used to copy a zone from one CSI data set to another.

ZONEDELETE
The SMP/E command used to delete a zone from a CSI data set.

ZONEEDIT
The SMP/E command used to change the values for a subentry in all the DDDEF or UTILITY entries in a given zone.

ZONEEXPORT
The SMP/E command used to copy a zone into a sequential data set.

ZONEIMPORT
The SMP/E command used to load an exported zone from a sequential data set into another zone.

ZONEMERGE
The SMP/E command used to copy one zone into another, or to merge two zones into one.

ZONERENAME
The SMP/E command used to change the name of a zone.

ZONESET
A group of zones to be used when processing an SMP/E command. For example, it may define the zones that the REPORT command is to check for cross-zone requisites. A ZONESET may also define a group of zones to be checked or ignored by the REJECT command.

ZONESET entry
An SMP/E entry defining a ZONESET.

z/OS UNIX System Services (z/OS UNIX)
The set of functions provided by the Shell and Utilities, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the z/OS operating system that allow users to write and run application programs that conform to UNIX standards.

Index

Special Characters

++element MCS
    USERMODs 175
++HOLD MCS
    coexistence considerations 211
    operands 143
++IF MCS
    cross-zone requisite checking 161
++JAR MCS
    using 185, 186
++JARUPD MCS
    using 185
++JCLIN MCS
    USERMODs 173
++MAC MCS
    USERMODs 174
++MACUPD MCS
    USERMODs 174
++MOD MCS
    USERMODs 174
++PROGRAM MCS
    USERMODs 175
++RELEASE MCS
    operands 143
++SRC MCS
    USERMODs 175
++SRCUPD MCS
    USERMODs 175
++USERMOD MCS
    building USERMODs 172
    examples 176
++VER MCS
    coexistence considerations 211
    USERMODs 172
++ZAP MCS
    USERMODs 174

Numerics

8-character userids 191

A

ACCEPT CHECK command
    corrective service 137
    functions 122
    preventive service 131
ACCEPT command
    corrective service 138
    description 15
    examples 32
    exception SYSMOD processing 146
    functions 123
    preventive service 132
    processing 30
ACCEPT command *(continued)*
    reports produced 34
    summary 48
Access Method Services (AMS) 63
accessibility
    contact IBM 217
adding SAF checks to SMP/E processing 192
alias defined for user catalog 63, 65
allocating data sets
    DDDEF entry 70
    dynamic allocation 69
    PTS 69
    SCDS 69
    SMPCSI 63
ALLZONES 152
alternate software inventory format 190
AMODE=64 link edit parameter 205
AMS utility
    allocating the CSI 63
    default values 72
    reorganizing a CSI 68
APAR fixes
    defining 42
APAR SYSMODs 6
APPLY CHECK command
    corrective service 136
    functions 120
    preventive service 127
    USERMODs 140
APPLY command
    corrective service 137
    description 15
    examples 24
    exception SYSMOD processing 145
    functions 121
    preventive service 130
    processing 23
    reports produced 26
    summary 48
    USERMODs 140
ASMA90 utility 72
assembler utility
    default values 72
assistive technologies 217
authorizing
    unsigned GIMZIP packages 54
authorizing assembler
    SMP/E commands 53
Automated Service Delivery Package
    HOLDDATA
        source of 147
automatic call libraries 149
automatic cross-zone requisite
        checking
    specifying 83

B

BACKUP entry 23
base functions 41
binder 72
Binder LONGPARM options 191
Binder RMODE=64 options 191
BPX.SUPERUSER security class
    coexistence considerations 206

C

CALL
    effect of CALLLIBS subentry on 72
calling SMP/E 85
cataloged procedure for SMP/E 87
catalogs
    alias defined for user catalog 63, 65
    for CSI 63
    listing 68
causer SYSMODs 183
CBPDO
    Recommended Service Upgrade 215
CBPDO package
    functions, source of 117
    HOLDDATA
        processing 147
        source of 146
CHANGEFILE
    coexistence considerations 208
CHECK operand
    on REJECT command 198
CISIZE
    for CSI data set 64
CLEANUP command 50
CLIENT data set
    changes to 196
    migration tasks 198
CLIST data set for SMP/E dialogs, LIBDEF restrictions 78
CMWA
    copy utility parameter 199
commands
    ACCEPT 48
    APPLY 48
    CLEANUP 50
    DEBUG 51
    GENERATE 48
    JCLIN 50
    LINK LMODS 51
    LINK MODULE 51
    LIST 48
    LOG 50
    RECEIVE 47
    RECEIVE ORDER 47
    REJECT 49
    REPORT CROSSZONE 48
    REPORT ERRSYSMODS 48
    REPORT SOURCEID 49, 167
    REPORT SYSMODS 49, 169
    RESETRC 52
    RESTORE 49
    SET 46
    UCLIN 49
    ZONECOPY 50
commands *(continued)*
    ZONEDELETE 51
    ZONEEDIT 50
    ZONEEXPORT 50
    ZONEIMPORT 51
    ZONEMERGE 51
    ZONERENAME 51
COMPACT
    coexistence considerations 209
comparing two zones
    LIST command 154
    REPORT CROSSZONE command 161
COMPAT
    parameter
        EXEC statement for GIMSMP 86
COMPAT=PM4 link edit parameter 205
compress utility
    default values 72
compressing a CSI 68
concatenating dialog libraries 78
conditional JCLIN processing
    coexistence considerations 204
consolidated software inventory data set (SMPCSI) 13
contact
    z/OS 217
CONTROLINTERVALSIZE
    for CSI data set 64
copy utility
    default values 72
copy utility parameter
    CMWA 199
    SPCLCMOD 199
corrective service
    description of 42
    installation
        ACCEPT CHECK processing 137
        ACCEPT processing 138
        APPLY CHECK process 136
        APPLY processing 137
        construct the fix 133
        deciding whether to accept 137
        prepare 135
        RECEIVE ORDER processing 135
        RECEIVE processing 135
        research the ACCEPT CHECK reports 138
        research the APPLY CHECK reports 136
        summary 133
        test 137
corrective service (APAR SYSMODs) 6
cover letters, listing 154
Cross Global Zone Reporting 192
cross-product load modules
    example 149
cross-zone load modules
    example 149
cross-zone requisite checking 161
Cross-Zone Requisite SYSMOD report 162
cross-zone requisites
    bypassing unsatisfied 85
    checking for 84
    resolving 85
    unsatisfied 84
CSI
    API for 16

CSI *(continued)*
cataloging considerations 63
defining entries
sample UCL statements 66
defining zones 65
importing 68
master CSI
definition of 59
parameter
EXEC statement for GIMSMP 86
reorganizing 68
structures, examples of
multiple-CSI structure 60
separate CSI for each SREL, combining target and DLIB zones 62
separate CSI for each zone 61
separate global zones 59
single-CSI structure 60
summary 55
customizing
an element (USERMOD SYSMODs) 7
JOB statement 81
SMP/E dialogs 80
CYLINDERS
for CSI data set 64

D

data element MCS
USERMODs 175
data set
ORDERSERVER 196
data sets
CLIENT 196
dynamically allocating 69
DATE parameter, EXEC statement for GIMSMP 86
DDDEF entry
instead of DD statements in cataloged procedure 88
used for dynamic allocation 70
DEBUG command 51
default utilities used by SMP/E 72
default zone group
defining 83
defaults
for SMP/E dialogs 80
SMPOUT 71
SYSPRINT 71
defining data sets
DDDEF entry 70
dynamic allocation 69
PTS 69
SCDS 69
SMPCSI 55
DEIINST job
coexistence considerations 206, 207
deleting SYSMODs from your system (RESTORE command) 15, 26
dependent functions 42
deprecate SHA-1 hash validation 189
dialogs
administration 197
concatenating libraries 78
connecting to ISPF master application menu 80
customizing
dialogs *(continued)*
customizing *(continued)*
migration tasks 202
overview 202
default values
panel GIM@PARM 80
SMP/E dialogs 80
distribution libraries
in logon procedure for SMP/E 79
editing dialog JCL 79
LIBDEF restrictions 78
logon procedure for SMP/E 79
query 197
required programs 77
restrictions 78
saving JCL generated by SMP/E dialogs 79
specifying defaults for 80
table data sets 79
target libraries
in logon procedure for SMP/E 79
displaying SMP/E data
API for 38
LIST command 37
Query dialogs 35
REPORT commands 38
distribution libraries (DLIBs)
description of 43
zones for 44, 65
distribution zone
defining 65
description of 13, 44
SYSMOD entries 31
DLIBZONE entry 65
DYNAM
coexistence considerations 212
dynamic allocation
CSI parameter on EXEC statement for GIMSMP 86
DDDEF entries 71
DDDEF entry 70
GIMDDALC 70
migration tasks 203
SMPPARM 70
sources of information
DDDEF entries 70
standard defaults 71
summary 69

E

END
EXEC statement parameter for GIMSMP 87
enhanced link name values
coexistence considerations 203
migration tasks 203
enhanced utility input 193
entries in CSI data sets
DLIBZONE entry 65
GLOBALZONE entry 65
TARGETZONE entry 65
exception data (HOLDDATA) 14
exception SYSMOD data
Automated Service Delivery Package
source of 147
CBPDO package

exception SYSMOD data *(continued)*
CBPDO package *(continued)*
source of 146
exception SYSMOD management
++HOLD MCS
operands 143
++RELEASE MCS
operands 143
categories of HOLDDATA 143
examples of 143
HOLDDATA
CBPDO package, processing 147
obtaining 146
introduction 143
processing
ACCEPT 146
APPLY 145
RECEIVE 144
REJECT 146
RESTORE 146
exception SYSMOD report 165
EXEC statement
COMPAT=NOWARNBYPASS 86
COMPAT=WARNBYPASS 86
CSI=dsname 86
DATE=date 86
PARM 86, 87
PROCESS=END 87
PROCESS=WAIT 87
exit routines
migration tasks 203
Expanded Service Option (ESO)
Recommended Service Upgrade 215
exporting a CSI data set 68

F

FILL
coexistence considerations 212
fixing an element 6
function SYSMODs
installation
ACCEPT CHECK processing 122
ACCEPT processing 123
APPLY CHECK processing 120
APPLY processing 121
choosing the update mode 119
get additional SYSMODs 121
preparation 118
RECEIVE function 119
RECEIVE processing 119
research the ACCEPT CHECK reports 123
research the APPLY CHECK reports 120
summary 117
test the new function 122
summary 41
functions in a system 1

G

GENERATE command
summary 48
generated JCL, saving for SMP/E dialogs 79
GIM@PARM (SMP/E Dialog Settings) 80, 202
GIM@PRIM (SMP/E Primary Option Menu) 80
GIM@UPRM
removal of 202
GIMDFOG
ORDER RETENTION subentry 197
GIMDFUT
removal of 202
GIMGTPKG service routine 197
GIMMPDFT
migration tasks 203
GIMMPUXD
migration tasks 203
GIMSAMPU 66
GIMSMP 86, 87
GIMUNZIP 204
GIMUNZIP extensions 191
GIMUTTBL
migration tasks 202
removal of 202
GIMXSID 199
GIMXTRX 205
GIMZIP
archive segmentation
coexistence considerations 200
overview 200
network delivery of SMP/E input
coexistence considerations 204
overview 204
user defined subdirectories
coexistence considerations 200
overview 200
GIMZIP packages
authorizing unsigned 54
global zone
defining 65
description of 13, 44
HOLDDATA entries 19
ORDER entry 196
SYSMOD entries 19, 23, 27, 31
GLOBALZONE entry 65

H

HEWLH096 utility 72
HFSINST job
coexistence considerations 206, 207
hierarchical file system
copy utility
default values 72
hierarchical file system element MCS
coexistence considerations 208
USERMODs 175
HOBSET
coexistence considerations 212
HOLDDATA
CBPDO package
processing 147
checking effect on installed SYSMODs (REPORT ERRSYSMODS) 165
ESO
processing 136
from the IBM Support Center 136
provided for SYSMODs 14

HOLDDATA *(continued)*
summary reports 205
HOLDDATA entries
created during RECEIVE 144
in the global zone 19
HTTPS download support 190

I

IDCAMS utility 63, 72
IEANUC01 load module 204
IEBCOPY utility 72
IEBUPDTE utility 72
IEWBLINK utility 72
IEWL utility 72
IMASPZAP utility 72
implicitly including modules from another product 149
installation methods
RECEIVE-APPLY-ACCEPT
117
installation procedures
corrective service 133
functions 117
preventive service 125
USERMODs 139
installing SMP/E
connecting the dialogs 77
installing SYSMODs
distribution libraries (ACCEPT command) 15, 30
target libraries (APPLY command) 15, 23
Internet service retrieval
preparing to use 93
introducing an element 3
invoking SMP/E 85
ISPCTL1 79
ISPCTL2 79
ISPF (Interactive System Productivity Facility)
concatenating libraries 78
connecting dialogs 77, 80
ISPF/PDF (Interactive System Productivity Facility/Program Development Facility) 77

J

Java Archive (JAR) files
building 185
coexistence considerations 200
in FMIDs 185
in PTFs 185, 186
using 185
JCL generated by SMP/E dialogs, saving 79
JCLIN command
summary 50
JOB statement
customizing 81

K

keepalive attribute
migration tasks 198
keyboard
navigation 217
PF keys 217
keyboard *(continued)*
shortcut keys 217
KEYS
for CSI data set 64

L

LIBDEF restrictions 78
link edit utility output
recommended DDDEF entries 83
LINK LMODS command
overview of 199
summary 51
LINK MODULE command
description 149
example 149
summary 51
when to use 149
link-edit utility
default values 72
link-editing object modules into load modules 1
LIST command
comparing two zones 154
cover letters 154
description 16
examples 37
listing a specific entry type 152
listing an FMID or FMIDSET 154
listing specific entries 153
reports produced 37
summary 48, 151
LISTCAT job 68
listing SMP/E data
API for 38
LIST command 37
Query dialogs 35
REPORT commands 38
load module data set for SMP/E dialogs, LIBDEF restrictions 78
load module, created by link-editing object modules 1
LOG command 50
logon procedure (TSO) 79
logon procedure, sample 79

M

master application menu for ISPF 80
master catalog, alias for user catalog 63, 65
master CSI
definition of 59
specified on DD statement 87
specified on EXEC statement 86, 87
MCS entry, created during RECEIVE 144
methods of installation 117
migration
OS/390 V1R2 SMP/E summary 212
OS/390 V1R3 SMP/E summary 210
OS/390 V2R5 SMP/E summary 207
OS/390 V2R7 SMP/E summary 206
overview 187
recommended steps for 188
terminology 187
modification control statement (MCS) 3, 18

modification identifiers
function (FMID) 10
replacement (RMID) 10
update (UMID) 10
MSGFILTER
coexistence considerations 209
MSGWIDTH
coexistence considerations 209
multiple-CSI zone structure 56, 60
multitasking using GIMDDALC SYSPRINT allocation 192

N

navigation
keyboard 217
NCAL
effect of CALLLIBS subentry on 72
network delivery of SMP/E input
coexistence considerations 204
NUCID
migration tasks 204
operand of APPLY command 204
subentry 204

O

object module, link-editing into a load module 1
OPTIONS entry
ORDER RETENTION subentry 196
ORDER entry
global zone 196
ORDER RETENTION
OPTIONS entry subentry 196
ORDERSERVER
data set 196

P

pasv attribute
migration tasks 198
PCF (programming control facility) 78
prerequisites for SYSMODs 8
preventive service
description of 42
installation
ACCEPT CHECK processing 131
ACCEPT processing 132
APPLY CHECK processing 127
APPLY processing 130
get additional SYSMODs 129
preparation 126
RECEIVE processing 126
research the ACCEPT CHECK reports 132
research the APPLY CHECK reports 128
test 130
PTF SYSMODs 5
RECEIVE ORDER requests 125
PROCESS parameter, EXEC statement for GIMSMP 87
products (function SYSMODs) 3
program element MCS
USERMODs 175
program temporary fix (PTF) 5
programming control facility (PCF) 78
PTF
introduction 125
Recommended Service Upgrade 215
summary 42
PTF cover letters, listing 154
PTF SYSMODs 5
PTS, summary 69

Q

Query dialog
description 16
example 35

R

RACF (z/OS Security Server)
regulating SMP/E utility programs with 202
RC
coexistence considerations 208
reason IDs 143
RECEIVE command
assigning source IDs to SYSMODs 198
corrective service 135
description 15
entries created
HOLDDATA entry 144
MCS entry 144
SYSMOD entry 144, 145
examples 20
functions 119
preventive service 126
processing 18
reports produced 22
summary 47
USERMODs 139
RECEIVE ORDER command
corrective service 135
summary 47
RECEIVE ORDER request
preventive service, source of 125
RECEXZGRP
coexistence considerations 209
Recommended Service Upgrade (RSU) 215
RECORDSIZE
for CSI data set 64
recovering from utility errors 75
RECZGRP
coexistence considerations 209
REJECT command
CHECK operand 198
exception SYSMOD processing 146
summary 49
removing SYSMODs from your system (RESTORE command) 15, 26
reorganizing a CSI 68
REPORT CALLLIBS command
removal of 199
REPORT command
description 16
example 38
reports produced 38
REPORT CROSSZONE command

REPORT CROSSZONE command *(continued)*
introduction 161
summary 48
REPORT ERRSYSMODS command
HOLDDATA for installed SYSMODs 165
introduction 165
summary 48
REPORT SOURCEID command
introduction 167
summary 49
REPORT SYSMODS command
introduction 169
summary 49
reports
ACCEPT CHECK reports
corrective service 138
functions 123
preventive service 132
APPLY CHECK reports
corrective service 136
functions 120
preventive service 128
USERMODs 140
Causer SYSMOD Summary report 183
SYSMOD Status report 183, 184
RESETRC command 52
RESTORE command
description 15
examples 28
exception SYSMOD processing 146
processing 26
reports produced 29
summary 49
restrictions
CLIST data set for SMP/E dialogs 78
dialogs 78
LIBDEF 78
load module data set for SMP/E dialogs 78
retention of HOLDDATA (IO13643) 192
retry processing 75
retry utility
default values 72
RMODE=ALIASES
coexistence considerations 212
RMODE=SPLIT
coexistence considerations 212
root cause analysis 183
RSU (Recommended Service Upgrade) 215

S

sample logon procedure 79
saving JCL generated by SMP/E dialogs 79
SCDS, summary 69
separate CSI for each SREL, combining target and DLIB zones 62
separate CSI for each zone 61
ServerPac
Recommended Service Upgrade 215
servicing a product
corrective service (APAR SYSMODs) 6
preventive service (PTF SYSMODs) 5
SET command 15, 46
SHAREOPTIONS
SHAREOPTIONS *(continued)*
for CSI data set 64
shortcut keys 217
SIDEDECKLIB
coexistence considerations 212
single-CSI structure 55, 60
SMP/E cataloged procedure 87
SMP/E commands 15, 46
SMP/E data
changing 157
listing 151
SMP/E data sets
residing in UNIX file system 205
SMPCSI 13
SMPPTS 18
SMPSCDS 23
SMPTLIB 18
SMP/E dialog libraries, concatenating 78
SMP/E dialogs
administration 197
customizing
migration tasks 202
overview 202
query 197
SMP/E reports
ACCEPT CHECK reports
corrective service 138
functions 123
preventive service 132
APPLY CHECK reports
corrective service 136
functions 120
preventive service 128
USERMODs 140
SMP/E summary 41
SMP/E V3R1 overview 202
SMP/E V3R2 overview 199
SMP/E V3R3 overview 197
SMP/E V3R4 overview 195
SMP/E V3R5 overview 192
SMP/E V3R6 overview 192
SMP/E V3R7 overview 189
SMPCSI
allocating 63
master CSI 87
multiple-CSI structure 56
single-CSI structure 55
summary 55
zones, description of 44
SMPDUMMY ddname
allocation rules 71
coexistence considerations 202
overview 201
SMPLTS
coexistence considerations 201
use of 46, 201
SMPMTS
use of 46
SMPOUT
default 71
SMPPROC (cataloged procedure for SMP/E) 87
SMPPTS
coexistence considerations 209
spill data sets 205

SMPPTS *(continued)*
use of 45
SMPPUNCH output
REPORT CROSSZONE command 162
REPORT ERRSYSMODS command 165
SMPSCDS
use of 45
SMPSTS
use of 46
SMPTABL
description 78
space allocation 79
source code, assembling to create an object module 2
SPCLCMOD
copy utility parameter 199
specifying the zone to be updated (SET command) 15
standard method of installation 117
summary of changes
z/OS SMP/E User's Guide xx, xxi
superzap utility
default values 72
SYS1.LINKLIB 205
SYS1.LPALIB 3
SYS1.MACLIB 69
SYS1.MIGLIB 3, 205
SYS1.SVCLIB 3
SYSDEFSD
DUMMY data set for
coexistence considerations 202
overview 201
SYSLIB
concatenation 90
SYSMOD
assigning source IDs to 198
SYSMOD Comparison HOLDDATA Report 192
SYSMOD entries
created during RECEIVE 144, 145
distribution zone 31
global zone 19, 23, 27, 31
target zone 23, 27
SYSMODs
APAR 6
definition of 41
description 2
function
base 4
dependent 4
hierarchy 41
listing
REPORT SOURCEID 167
SMPPUNCH 167
prerequisites 8
PTF 5
summary 41
USERMOD 7
SYSPRINT
default 71
system modifications 2

T

table data sets for dialogs 79
target libraries, description of 43
target zone
defining 65
description of 13, 44
SYSMOD entries 23, 27
TARGETZONE entry 65
temporary fix (APAR SYSMODs) 6

U

UCLIN command
general syntax 158
introduction 157
samples in GIMSAMPU 66
summary 49
UNIX file system
identification of data sets 205
SMP/E data sets residing in 205
update utility
default values 72
UPGRADE command
coexistence considerations 199
overview of 199
user catalog
alias in master catalog 63, 65
for CSI 63
user interface
ISPF 217
TSO/E 217
user-defined subdirectories
coexistence considerations 200
USERMOD
creating 171
examples 176
installation
APPLY CHECK process 140
APPLY process 140
prepare 139
RECEIVE processing 139
research APPLY CHECK reports 140
summary 139
test 141
MCS statements
++element (for data elements) 175
++JCLIN 173
++MAC 174
++MACUPD 174
++MOD 174
++PROGRAM 175
++SRC 175
++SRCUPD 175
++USERMOD 172
++VER 172
++ZAP 174
hierarchical file system element 175
preventing ACCEPT processing 141
summary 42
USERMOD SYSMODs 7
utility errors, recovery from 75
utility programs
default values
access method services (AMS) 72
assembler 72
compress 72
copy 72

utility programs *(continued)*
default values *(continued)*
hierarchical file system copy 72
link-edit utility 72
retry 72
superzap 72
update 72
specifying which utility programs SMP/E can call 202
UTIN
coexistence considerations 212

V

VERSION command
coexistence considerations 209

W

WAIT
EXEC statement parameter for GIMSMP 87

X

XZREQCHK subentry
coexistence considerations 210
use of 83

Z

z/OS Security Server (RACF)
regulating SMP/E utility programs with 202
z/OS SMP/E User's Guide
content, changed xxi
content, deleted xxi, xxii
content, new xx, xxi
summary of changes xx, xxi
zone entries
impacts 196
zone group
default 83
defining 83
specifying on command 84
ZONEINDEX for 84
zone structures
examples 59
multiple CSIs 56
single CSI 55
ZONECOPY command
alternative to UCLIN 158
summary 50
ZONEDELETE command
alternative to UCLIN 158
summary 51
ZONEEDIT command
alternative to UCLIN 157
summary 50
ZONEEXPORT command
alternative to UCLIN 158
summary 50
ZONEIMPORT command
alternative to UCLIN 158
summary 51
ZONEINDEX
ZONEINDEX *(continued)*
for zone group 84
ZONEMERGE command
summary 51
ZONEMERGE command extensions 191
ZONERENAME command
alternative to UCLIN 158
summary 51
zones
comparing
LIST command 154
REPORT CROSSZONE command 161
REPORT SYSMODS command 169
description of 44
zones in the SMPCSI data set 13
ZONESET entry
cross-zone processing
161
defining 83

Product Number: 5655-ZOS

SA23-2277-70