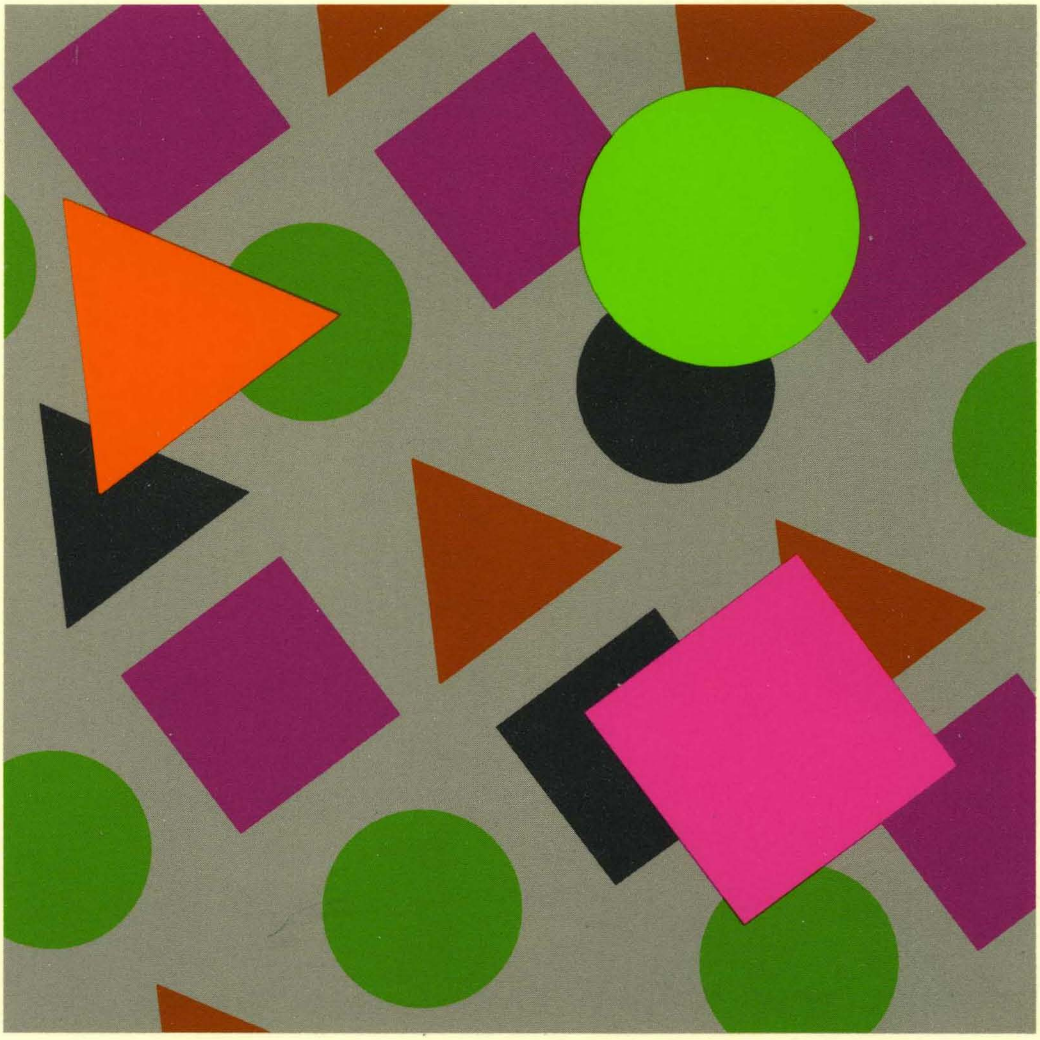
System Modification Program Extended

SC28-1302-11

User's Guide

Release 6



SC28-1302-11

System Modification Program Extended

User's Guide

Release 6

Note!

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

Twelfth Edition (April 1992)

This book replaces the previous edition, SC28-1302-10, which is now obsolete. Changes or additions to text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to System Modification Program Extended (SMP/E) Release 6 for OS/VS2 (MVS) and to all subsequent releases and modifications unless otherwise indicated in new editions.

Consult the latest edition of the *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for current information on this product.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation
Information Development
Department 52QA/MS 911
Neighborhood Road
Kingston, NY 12401-0000
USA

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

Contents

Figures

Tables

Notices

Notice to OS/VS1 Users

Do **not** use this publication for your OS/VS1 system. SMP/E Release 6 does not run under OS/VS1. Use the latest SMP/E Release 2 edition of this publication instead. The order number for the Release 2 edition is GT28-1108.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protected rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Trademarks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries:

BookManager
IBM
MVS/ESA
MVS/SP
MVS/XA

Preface

Who Should Read This Publication

Anyone who uses System Modification Program Extended (SMP/E) Release 6, 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 *SMP/E Reference* for details on commands.

This publication does not contain any programming interfaces for customers. For information on the programming interfaces provided by SMP/E, see the chapter on installation exit routines in *SMP/E Reference*.

How to Use This Publication

Use the chart below to guide you through this publication.

Chapter	Subject	Description
Part 1. Introduction to SMP/E		
1	A Summary of SMP/E	Discusses: • A summary of the system library structure • Some of the things you can do with SMP/E. ***Who should read:*** Everyone
2	Data Sets Used by SMP/E	Describes: • The data sets SMP/E uses • How to allocate these data sets • How to name data sets • How to set up the data SMP/E needs. ***Who should read:*** Everyone
3	Preparing to Use SMP/E	Describes: • How SMP/E is called • Using dynamic allocation of data sets • System utility programs used by SMP/E • Backing up the SMP/E data sets. ***Who should read:*** Everyone
Part 2. Guide to Using SMP/E		
4 (see Note)	Installing a New Function	Describes **recommended** methods for installing a new function using SMP/E. ***Who should read:*** Anyone installing a new function
5 (see Note)	Installing Preventive Service	Describes **recommended** methods for installing preventive service. ***Who should read:*** Anyone installing preventive service

Chapter	Subject	Description
6 (see Note)	Installing Corrective Service	Describes the **recommended** methods for installing corrective service. ***Who should read:*** Anyone installing corrective service
7 (see Note)	Installing a User Modification	Describes **recommended** methods for installing user modifications. ***Who should read:*** Anyone installing user modifications
Note: Chapters 4–7 do not assume that you have in-depth knowledge of SMP/E. Each chapter describes the **recommended** way of doing the task. The chapters do not tell you about all the options available. See *SMP/E Reference* for details on the available options.		
8	Managing Exception SYSMODs	Describes how SMP/E handles system modifications (SYSMODs) requiring special processing. ***Who should read:*** Anyone installing changes
9	Creating Target Zones After Using a Special Generation Procedure	Describes: • How to create new target zones • How to update existing target zones. ***Who should read:*** Anyone who wants to look at SMP/E data
10	Displaying the Data Managed by SMP/E: The LIST Command	Describes how to see the data SMP/E uses. ***Who should read:*** Anyone who wants to look at SMP/E data
11	Changing the Data SMP/E Manages: The UCLIN Command	Describes how to update the SMP/E database. ***Who should read:*** Anyone who wants to change any SMP/E data
12	Identifying Cross-Zone Requisites: The REPORT CROSSZONE Command	Describes how to check conditional requisites for products in separate zones. ***Who should read:*** Anyone installing changes
13	Identifying Installed SYSMODs Affected by Error Holds: The REPORT ERRSYSMODS Command	Describes how to check for installed SYSMODs that are affected by error holds. ***Who should read:*** Anyone installing changes
14	Listing the Source IDs in a Zone: The REPORT SOURCEID Command	Describes how to use the REPORT SOURCEID command to list the source IDs assigned to SYSMODs in a given zone or ZONESET. ***Who should read:*** Anyone who wants to look at SMP/E data
15	Comparing the SYSMODs Installed in Two Zones: The REPORT SYSMODS Command	Describes how to use the REPORT SYSMODS command to check if SYSMODs installed in one zone are applicable to and installed in a second zone. ***Who should read:*** Anyone who wants to look at SMP/E data
16	Building a User Modification	Describes how to build a user modification so that SMP/E can install it. ***Who should read:*** Anyone making a user modification to the system

Chapter	Subject	Description
17	Processing Inline JCLIN at ACCEPT Time	Describes how to use inline JCLIN to update the distribution zone during ACCEPT processing. ***Who should read:*** Anyone installing a mix of SYSGEN-supported and non-SYSGEN-supported products
18	Determining Which SYSMODs Led Others to Fail: The Causer SYSMOD Summary Report	Describes root cause analysis. Provides examples of how to use the causer SYSMOD information in the Causer SYSMOD Summary report and the SYSMOD Status report. ***Who should read:*** Anyone trying to determine the cause of SYSMOD failure
Part 3. Appendixes		
A	SYSMOD Installation Checklists	Contains checklists to help guide you through: • Installing new functions (see also Chapter 4) • Installing preventive service (see also Chapter 5) • Installing corrective service (see also Chapter 6) • Installing user modifications (see also Chapter 7) • Rebuilding target zones for a mix of products, of which some are included by a generation procedure and some are not. ***Who should read:*** Anyone using Chapters 4 through 7
B	Abbreviations	Explains the abbreviations used in this publication. ***Who should read:*** Everyone

The SMP/E Library

Table 1 lists the SMP/E Release 6 publications and briefly describes each one.

Table 1 (Page 1 of 2). Publications for SMP/E Release 6

Title	Description
MVS Software Manufacturing Offerings General Information, GC23-0351 Online book: GIM9MST	Provides a summary of SMP/E, CBIPO, and CBPDO.
SMP/E Program Directory for Installation Planning (English Feature), GC23-0130 Online book: GIM1MST *SMP/E Program Directory for Installation Planning (Japanese Feature)*, GC23-0469 Online book: GIMJMST	Explains how to plan for installing SMP/E Release 6 using SMP/E Release 5 or higher.

Table 1 (Page 2 of 2). Publications for SMP/E Release 6

Title	Description
SMP/E User's Guide, SC28-1302 Online book: GIM8MST	Describes how to use SMP/E to install programs and service.
SMP/E Messages and Codes, SC28-1108 Online book: GIM4MST	Explains SMP/E messages and return codes and the actions to take for each message and code.
SMP/E Reference, SC28-1107 Online book: GIM6MST	Explains SMP/E commands and processing in detail.
SMP/E Reference Summary, SX22-0006 Online book: GIM7MST	Reviews the SMP/E commands in a convenient form.
SMP/E Program Packaging Guide, SC23-0221 Online book: GIM5MST	Explains how to package programs for installation by SMP/E.
SMP/E CBIPO Dialogs User's Guide, SC23-0538 Online book: GIMIMST	Explains how to use the CBIPO dialogs to install, reinstall, and redistribute CBIPO orders.
SMP/E Diagnosis Guide, LY27-8047 (no online book)	Explains how to handle suspected SMP/E problems.

Notes:

1. You can order hardcopy versions of individual SMP/E Release 6 publications or use a Bill of Forms number to order the complete set of unlicensed publications. (Because *SMP/E Diagnosis Guide* is a licensed publication, it must be ordered separately and is only available if you are licensed for SMP/E Release 6.)
   - For the English feature of SMP/E, use **SBOF-1587**.
   - For the Japanese feature of SMP/E, use **SBOF-3161**.
2. Online versions of the SMP/E Release 6 books (except for *SMP/E Diagnosis Guide*) are provided with the product.
3. You can order binders and inserts for the SMP/E library. (Inserts are the slip-in covers and spine for a binder.) Here are the order numbers:
   - For *SMP/E Reference*:
     - Binder plus inserts: **SBOF-2136**
     - Inserts only: **SX23-0442**
   - For the rest of the SMP/E library:
     - Binder plus inserts: **SBOF-2137**
     - Inserts only: **SX23-0443**

   Note: You will probably need two binders for the rest of the library.

Related Publications

Product	Publications
CBIPO and CBPDO	- *MVS Software Manufacturing Offerings General Information*, GC23-0351 - *MVS Custom-Built Offerings Planning and Installation*, SC23-0352
ISPF Version 2 Release 3	- *ISPF and PDF General Information*, GC34-4116 - *ISPF Dialog Management Guide*, SC34-4112 - *ISPF Dialog Management Services Examples*, SC34-4113
MVS/ESA* (MVS/SP* Version 3)	- *MVS/ESA Library User's Guide*, GC28-1563 - *MVS/ESA Linkage Editor and Loader User's Guide*, SC26-4510 - *MVS/ESA JCL Reference*, GC28-1829 - *MVS/ESA JCL User's Guide*, GC28-1830 - *MVS/ESA Basics of Problem Determination*, GC28-1839 - *MVS/ESA Service Aids*, GC28-1844 - *MVS/ESA Message Library: System Codes*, GC28-1815 - *MVS/ESA Diagnosis: System Reference*, LY28-1011 - *MVS/ESA Integrated Catalog Administration: Access Method Services Reference*, SC26-4500 - *MVS/ESA VSAM Catalog Administration: Access Method Services Reference*, SC26-4501 - *MVS/ESA Catalog Administration Guide*, SC26-4502 - *MVS/ESA Data Administration Guide*, SC26-4505 - *MVS/ESA Data Administration: Utilities*, SC26-4516 - *MVS/ESA Data Administration: Macro Instruction Reference*, SC26-4517 - *MVS/ESA VSAM Administration Guide*, SC26-4518
MVS/XA*	- *MVS/XA System Macros and Facilities*, Volume 1, GC28-1150 - *MVS/XA Linkage Editor and Loader User's Guide*, GC26-4143 - *MVS/XA Data Administration: Utilities*, GC26-4150 - *MVS/XA System Data Administration*, GC26-4010 - *MVS/XA Data Administration Guide*, GC26-4013 - *MVS/XA Data Administration: Macro Instruction Reference*, GC26-4014 - *MVS/XA VSAM Administration Guide*, GC26-4015 - *MVS/XA VSAM Administration: Macro Instruction Reference*, GC26-4152 - *MVS/XA VSAM Catalog Administration: Access Method Services Reference*, GC26-4136 - *MVS/XA ICF Administration: Access Method Services Reference*, GC26-4135 - *MVS/XA Access Method Services Reference Summary for Integrated Catalog Facility*, GX26-3739 - *Assembler H Version 2 Application Programming: Guide*, SC26-4036 - *Assembler H Version 2 Application Programming: Language Reference*, GC26-4037 - *MVS/XA JCL User's Guide*, GC28-1351 - *MVS/XA JCL Reference*, GC28-1352 - *MVS/XA Service Aids*, GC28-1159

MVS/ESA, MVS/SP, and MVS/XA are trademarks of the IBM Corporation.

Product	Publications
MVS/370	• *MVS/370 Linkage Editor and Loader*, GC26-4061 • *MVS/370 Data Facilities Product Utilities Data Administration*, GC26-4065 • *MVS/370 Data Facilities Product System Data Administration*, GC26-4056 • *MVS/370 Data Facilities Product Data Administration Guide*, GC26-4058 • *MVS/370 Data Facilities Product Data Administration Macro Instruction Reference*, GC26-4057 • *MVS/370 VSAM User's Guide*, GC26-4066 • *MVS/370 VSAM Reference*, GC26-4074 • *MVS/370 Catalog Administration Guide*, GC26-4053 • *MVS/370 ICF Catalog Administration: Access Method Services Reference*, GC26-4153 • *MVS/370 Access Method Services Reference Summary for the Integrated Catalog Facility*, GX26-3745 • *MVS/370 VSAM Catalog Administration: Access Method Services Reference*, GC26-4154 • *OS/VSE, DOS/VS, and VM/370 Assembler Language*, GC33-4010 • *MVS/370 JCL Reference*, GC28-1350 • *MVS/370 JCL User's Guide*, GC28-1349 • *OS/VS2 MVS System Programming Library: Service Aids*, GC28-0674
MVS Release 3.8	• *OS/VS Linkage Editor and Loader*, GC26-3813 • *OS/VS, DOS/VSE, and VM/370 Assembler Language*, GC33-4010 • *OS/VS MVS Utilities*, GC26-3902 • *OS/VS2 MVS JCL*, GC28-0692 • *OS/VS2 System Programming Library: Service Aids*, GC28-0674 • *OS/VS2 System Programming Library: Data Management*, GC26-3830 • *OS/VS2 MVS Data Management Services Guide*, GC26-3875 • *OS/VS Virtual Storage Access Method Programmer's Guide*, GC26-3838 • *OS/VS2 Access Method Services*, GC26-3841

Summary of Changes for SMP/E Release 6

April 1992 (SC28-1302-11)

Change for Root Cause SPE

- **Root Cause Analysis**

  The changes for root cause analysis are now available in a small programming enhancement (SPE), PTF UR90259 for FMID HMP1600 and, if the Japanese feature is installed, PTF UR90260 for FMID JMP1611. Information about root cause analysis is clearly marked in all the SMP/E manuals to indicate that it applies only to the root cause SPE.

  To simplify the process of determining the errors that caused SYSMODs to fail, SMP/E provides a new report called the Causer SYSMOD Summary report. This report summarizes the messages describing the errors that actually caused the SYSMODs to fail and that must be fixed in order to resolve the problem. The report filters out subsequent messages that are generated because of these initial errors.

  The SYSMOD Status report has also been improved:

  - For each SYSMOD that failed, the report indicates the ID of the SYSMOD that caused this failure.
  - A new statistics section indicates the number of SYSMODs that have been applied, accepted, or restored.

  A new chapter has been added to the *SMP/E User's Guide* to explain how to use these reports.

  Also, messages have been added, reworded, and deleted to provide better information for determining the cause of SYSMOD failure.

End of Change for Root Cause SPE

- **Corrections**

  This book contains technical corrections and clarifications.

Vertical bars to the left of the text mark all these changes.

August 1991 (SC28-1302-10)

- **CBIPO Dialogs**

  SMP/E now includes dialogs that can be used to install CBIPO packages. These dialogs can be used instead of the existing batch installation process.

  With the CBIPO dialogs you can:

  - Install a system or subsystem from a CBIPO or redistribution tape.
  - Reinstall existing user application programs and user data while installing a new system or subsystem from a CBIPO or redistribution tape.

- Redistribute a system or subsystem by either:

  - Copying the complete system or subsystem to DASD at the same location. This is called *local redistribution*.

  - Copying the complete system or subsystem to tape, transporting it, and installing the copied system or subsystem from the tape onto DASD at either the same location or at a different location. This is called *remote redistribution*.

  For both types of redistribution, the data set, DASD, and catalog configuration of the copied system or subsystem need not be the same as the configuration of the original system or subsystem.

- Process I/O definitions for an MVS system that was installed using the CBIPO dialogs:

  - Process an MVSCP deck

  - Include existing IODFs (I/O definition files) for use during IPL and to be copied during reinstall or redistribution processing.

- Connect a previously installed subsystem to another MVS system.

- **Online Books**

  Online books will now be provided for SMP/E. These online books can be viewed using IBM* BookManager* READ Release 2.

- **New Packaging for SMP/E**

  SMP/E is now packaged using data elements. As a result, you can now install SMP/E only with SMP/E Release 5 or higher. (To install the online books you need SMP/E Release 5.1 or higher.) You cannot install this release of SMP/E using SMP4 or a release of SMP/E before Release 5.

- **SYSMOD Status Report Accuracy**

  The accuracy of the SYSMOD Status report has been improved when listing unsatisfied requisite system modification (SYSMOD) and hold conditions. SMP/E has been enhanced to search beyond the initial point of SYSMOD termination and report on additional requisite SYSMOD and hold conditions that are unresolved for the failing SYSMOD. These conditions, identified with a dash (-) in the SYSMOD Status report, are also identified with additional GIM359*xx* messages. This enhancement can improve system programmer productivity by reducing the number of times a system programmer has to run a mass APPLY CHECK or ACCEPT CHECK in order to resolve all conditions required to install the desired SYSMODs.

IBM and BookManager are trademarks of the IBM Corporation.

- **REPLACE Operand on GENERATE Command**

  The GENERATE command now provides a new operand, REPLACE, to create copy and link-edit jobs that replace existing members and load modules.

Change for Root Cause SPE

- **Root Cause Analysis**

  To simplify the process of determining the errors that caused SYSMODs to fail, SMP/E will provide a new report called the Causer SYSMOD Summary report. This report summarizes the messages describing the errors that actually caused the SYSMODs to fail and that must be fixed in order to resolve the problem. The report filters out subsequent messages that are generated because of these initial errors.

  The SYSMOD Status report has also been improved:

  - For each SYSMOD that failed, the report indicates the ID of the SYSMOD that caused this failure.
  - A new statistics section summarizes the number of SYSMODs that were successfully processed and the number of SYSMODs that were not successful.

  In addition, messages have been added, reworded, and deleted to provide better information for determining the cause of SYSMOD failure.

  The changes for root cause analysis will be available later in a small programming enhancement (SPE). (See the announcement letter for details.) Information about root cause analysis is, however, currently included in the SMP/E manuals. It is clearly marked to indicate that it only applies to the root cause SPE.

End of Change for Root Cause SPE

Part 1. Introduction to SMP/E

Chapter 1. A Summary of SMP/E

This chapter provides a summary of the System Modification Program Extended (SMP/E). It briefly describes:

- What SMP/E is
- What system modifications are
- The data sets used by SMP/E
- How to use SMP/E.

What Is SMP/E?

SMP/E is a tool that helps you install and maintain software products and service by selecting the correct level of change, calling the utility programs to install the changes, and recording the changes that have been made. SMP/E can be used to install and service any software packaged as a system modification, or *SYSMOD*.

SMP/E can be run using either batch jobs or using dialogs under Interactive System Productivity Facility/Program Development Facility (ISPF/PDF). Two types of dialogs are provided by SMP/E:

- **CBIPO dialogs.** These are dialogs for installing and redistributing Custom-Built Installation Process Offering (CBIPO) packages (instead of using the batch jobs provided in the CBIPO related installation materials [RIMs]).
- **SMP/E dialogs.** These are dialogs that help you interactively query the SMP/E database as well as create and submit jobs to process SMP/E commands.

What Are SYSMODs?

A program consists of *elements* such as macros, modules, or other types of data. For SMP/E to install and service a program, it needs SMP/E modification control statements, or MCSs, for the elements. MCSs describe the elements of the program and any relationships the program has with other programs that may also be installed on the same system. The combination of elements and MCSs is called a *system modification*, or *SYSMOD*.

There are four types of SYSMODs:

- **Function SYSMODs** (or functions). These 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:

  - Base functions are function SYSMODs that either add or replace a complete functional area in the system. Examples of base functions are SMP4, SMP/E, and MVS/XA.*

* MVS/XA is a trademark of the IBM Corporation.

- Dependent functions are function SYSMODs that provide an addition to an existing functional area in the system. This type of function is called a dependent function because its installation depends on a base function already being installed. Examples of dependent functions are the language features for SMP/E.

- **PTFs**. These are IBM-supplied, tested fixes for reported problems. These fixes are meant to be installed in all environments. PTFs may be used as preventive service to avert 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.

  PTFs are always dependent on a function SYSMOD being installed and often depend on the installation of other PTFs.

- **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.

  APARs are always dependent on a function SYSMOD being installed and sometimes depend on the installation of a particular PTF (that is, an APAR is designed to be installed on a particular preventive service level of an element).

- **User modifications** (or USERMODs). These are SYSMODs built by you, either to change IBM code or to add independent functions to the system.

  USERMODs are always dependent on a function SYSMOD being installed and sometimes depend on the installation of some PTFs, or APAR fixes, or other user modifications.

  Note: If you wish 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 user modification.

Figure 1 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.

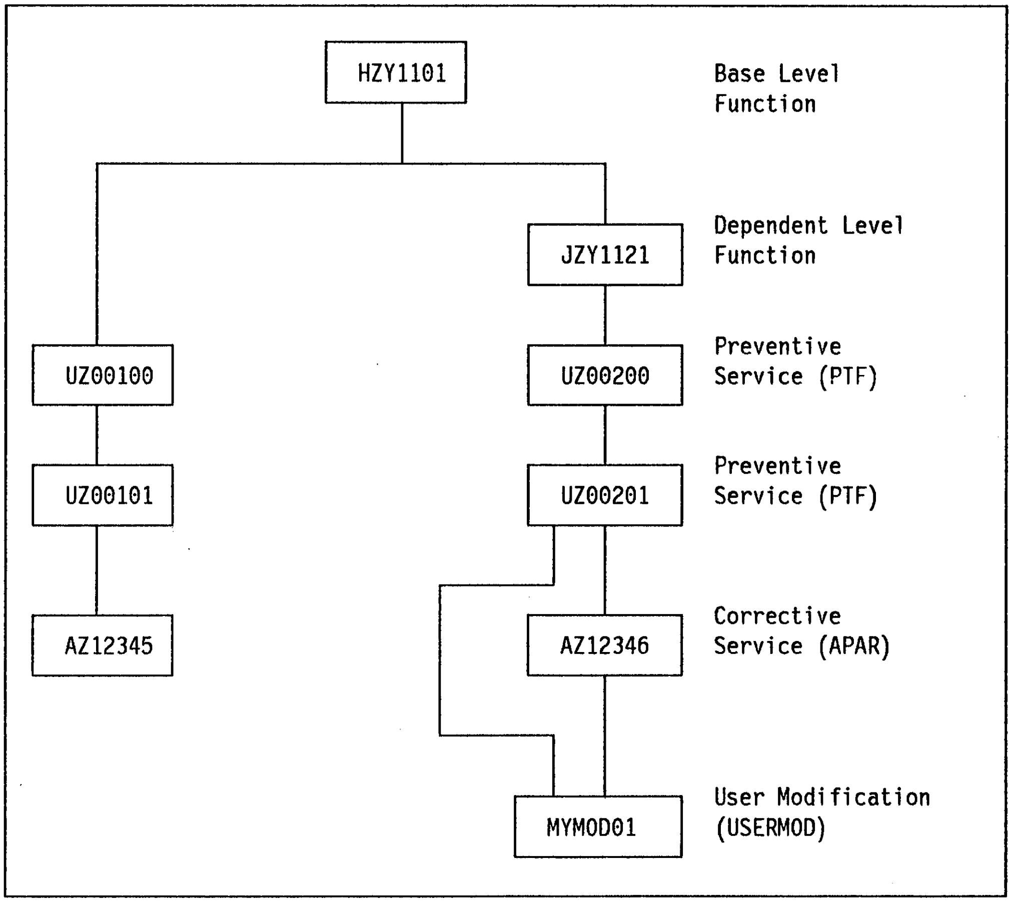


Figure 1. Example of a SYSMOD Hierarchy

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, see the APPLY and ACCEPT commands in *SMP/E Reference*.

Data Sets Used by SMP/E

When SMP/E processes SYSMODs, it installs the changes in the libraries that contain the affected elements and updates its own records of the processing that was done. SMP/E installs program elements into two types of libraries:

- **Target libraries**. These contain the executable code that constitutes the running system.
- **Distribution libraries**. These contain the master copy of all elements for a system. The distribution libraries (DLIBs) 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 it is necessary to replace or update part of a running system.

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

- **SMPCSI (CSI) data sets**. These are Virtual Sequential Access Method (VSAM) data sets used to control the installation process and record the results of processing.
- **SMPPTS (PTS) data set**. This is a data set for temporary storage of SYSMODs waiting to be installed.
- **SMPSCDS (SCDS) data set**. This is a data set that contains backup copies of target zone entries that are created during APPLY processing.
- **Various 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 available on the structure, content, and status of the system. SMP/E provides this information in reports, listings, and dialogs to help you:

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

Summary of Using SMP/E

You can run SMP/E either using batch jobs or using the SMP/E dialogs, which run under ISPF. For either method, these are the basic steps you must follow to install and maintain your system with SMP/E:

1. Obtain or build the SYSMODs to be installed.

   For an example of building a USERMOD, see Chapter 16. For more information about building and packaging SYSMODs, see *SMP/E: Program Packaging Guide*.

2. Allocate and initialize the CSI and other data sets.

   For more information, see Chapter 2.

3. Make sure you have any necessary cataloged procedures and utilities to call SMP/E.

   For more information, see Chapter 3.

4. Run SMP/E commands to install SYSMODs and to monitor and maintain your system.

   For more information, see Part 2.

The following subsections summarize the SMP/E commands you can use to do this processing, the MCSs that define how the elements in SYSMODs should be processed, and the SMP/E reports that describe the results of this processing.

Summary of SMP/E Commands

SMP/E commands are used to establish the SMP/E database, to install product and service SYSMODs, to remove SYSMODs that contain errors, and to display and manage the information in the SMP/E database. Following are brief descriptions of the SMP/E commands used to do this processing. For more information about these commands, see *SMP/E Reference*.

Establishing the SMP/E Database: UCLIN and JCLIN

All SMP/E processing is controlled by information in the CSI data set. Before processing any SYSMODs, you must first provide the required information in the CSI.

The UCLIN Command: The UCLIN command allows you to create or change entries in CSI data sets. Initially, it is used to create all the entries required to control processing, such as entries for:

- Defining your system structure
- Identifying the utilities SMP/E is to use
- Providing information for dynamic allocation support.

Once you create your CSI, you can use UCLIN to change certain entries after errors or to change processing information.

The JCLIN Command: In order for SMP/E to successfully install a SYSMOD, it must have information about the structure of your target libraries, such as:

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

Although you can use UCLIN to define all this information, the task is time consuming and error-prone. Instead, you can use the JCLIN command. 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.

Installing SYSMODs: RECEIVE, APPLY, and ACCEPT

Once you have set up the SMP/E database, your next task will probably be the installation of SYSMODs, which is the primary purpose of SMP/E.

The RECEIVE Command: The RECEIVE command performs the following functions:

- Reads in SYSMODs from an input file, which is defined by the SMPPTFIN DD statement
- Reads in HOLDDATA for exception SYSMODs from another input file, which is 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.

The APPLY Command: Once the SYSMODs have been received, you can use the APPLY command to direct SMP/E to install them into the appropriate target libraries.

The ACCEPT Command: After installing a SYSMOD into the target library and testing it to your satisfaction, you can use the ACCEPT command to have SMP/E install those SYSMODs into the distribution libraries and to delete them from the PTS and the global zone.

Note: While the normal way of processing for a SYSMOD is RECEIVE, APPLY, and then ACCEPT, you may skip the APPLY step and go directly from RECEIVE to ACCEPT. See Chapter 4, "Installing a New Function" on page 49 for additional information.

Removing SYSMODs: REJECT and RESTORE

In addition to the commands that help you install SYSMODs, SMP/E also has two commands to help you remove SYSMODs. These are REJECT and RESTORE.

Note: There is no SMP/E command to remove a SYSMOD from the distribution libraries once you have accepted it.

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

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

The RESTORE Command: Once you have installed a SYSMOD into the target libraries, you should test it to make sure the fix 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.

Doing Other SMP/E Processing

In addition to the commands that directly deal with SYSMODs, SMP/E also has commands that give you further access to the data retained by SMP/E.

Identifying Where the Commands Are Applicable: The SET Command: The main SMP/E database is the CSI. 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*. Each zone represents a single set of target libraries (these zones are called *target zones*), or a set of distribution libraries (called *distribution zones*), or the data present in the PTS (called the *global zone*). Therefore, a single CSI can be used to control many sets of libraries.

Because the CSI can be partitioned into multiple zones, you must tell SMP/E which zone you are working with before it can process any other commands.

You direct SMP/E processing to a specific zone by coding the zone name on the SET command.

A SET command must be specified before SMP/E will process any other commands.

Changing Information in the SMP/E Database: The UCLIN Command: As mentioned previously, the UCLIN command lets you add, delete, or change any entry in the SMP/E database.

Note: This command should be used with **extreme** caution, as an incorrect change could result in the later incorrect installation of a SYSMOD.

Building a System from a Distribution Library: The GENERATE Command: Use the GENERATE command to create a job stream that builds a system (a set of target libraries) from a set of distribution libraries.

Displaying SMP/E Data: The LIST Command: Use the LIST command to display the data that SMP/E retains. There are several operands of the LIST command that allow you to list subsets of the data.

Copying Zones: The ZONECOPY Command: Use the ZONECOPY command to copy an entire target or distribution zone from one CSI data set to another CSI data set. When you use the ZONECOPY command, SMP/E checks to make sure that the zone you copy into is empty.

Deleting Zones: The ZONEDELETE Command: 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 describing the new set of target system libraries that were 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.

Exporting Zones: The ZONEEXPORT Command: 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:

- A backup copy

  You can use the copy to recreate a zone that was accidentally erased or destroyed.
- A “transportable” copy

  You can use the copy to create the same zone on another system or in another CSI data set on the same system.

Importing Zones: The ZONEIMPORT Command: Use the ZONEIMPORT command to reload an exported zone into a distribution, target, or global zone. ZONEIMPORT may only be used with output from ZONEEXPORT.

Merging Zones: The ZONEMERGE Command: 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 the two zones.

Modifying Zone Entries: The ZONEEDIT Command: Use the ZONEEDIT command to quickly change the values for a field in different DDDEF or UTILITY entries in the same zone.

Renaming Zones: The ZONERENAME Command: 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 will use when building a new target zone.

Cleaning Up the SMPSTS, SMPMTS, and SMPSCDS Data Sets: The CLEANUP Command: Use the CLEANUP command to delete entries from the SMPSTS, SMPMTS, and SMPSCDS data sets when a SYSMOD has been accepted into the related distribution zone.

Recording Information: The LOG Command: Use the LOG command to write to the SMPLOG (LOG) and SMPLOGA (LOGA) data sets. This is useful for maintaining documentation about your system, such as who installed a certain SYSMOD, and why.

Requesting Diagnostic Processing: The DEBUG Command: 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.

Unconditionally Processing a Command: The RESETRC Command: Normally, the execution of an SMP/E command depends on the successful execution of all previous 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 previous command failed.

Comparing Zones: The REPORT Command: The REPORT command helps you compare zones to find out if any additional SYSMODs need to be processed. For example, you can use the REPORT command to do the following:

- To list conditional requisites that must be installed in certain zones because of SYSMODs that are installed in other zones. This information can help you synchronize service for related products that are in different zones.
- To list SYSMODs that have already been installed but that are affected by error holds subsequently received.
- To list source IDs that are assigned to SYSMODs in a given zone or ZONESET.
- To list SYSMODs that are installed in one zone that are applicable to, but not yet installed in, a second zone.

Converting Data Sets to SMP/E Release 6 Format: The CONVERT Command: Because SMP/E Release 6 is packaged using data elements, users who are currently at SMP4 or SMP/E Release 4 or earlier cannot migrate directly to SMP/E Release 6. You must do the following:

However, you can wait until SMP/E Release 6 is available in an MVS CBIPO, then order a CBIPO MVS feature that includes SMP/E Release 6.

Install the CBIPO feature. Then convert any data sets from your previous release that will be used with SMP/E Release 6. (This is done with the SMP/E CONVERT command.)

Note: If you are migrating to SMP/E Release 6 from SMP/E Release 5.1 or SMP/E Release 5, you do not need to convert any data sets—their format is compatible with SMP/E Release 6.

For detailed information on CONVERT, see the SMP/E program directory and the chapter on the CONVERT command in *SMP/E Reference*.

Summary of SMP/E MCS Statements

As mentioned earlier, SYSMODs consist of elements and MCSs. MCSs describe the elements of a program and any relationships the program has with other programs that may also be installed on the same system. These are the kinds of information defined by MCSs:

- The type of SYSMOD
- The environment in which the SYSMOD can be installed
- Requirements for installing the elements in the SYSMOD
- The elements that are affected by the SYSMOD
- If the SYSMOD requires special processing before it can be installed.

Normally, you should not be concerned with MCSs. However, there may be times when you need to understand the MCSs and their operands:

- When building a user modification
- When packaging one of your applications in SMP/E format
- When an error is found in a SYSMOD.

Following are the MCSs and a brief description of what each does. For a complete description of each MCS and its operands, see the chapter on SMP/E MCSs in *SMP/E Reference*.

SYSMOD-Type MCSs

++APAR
identifies an APAR fix.

++FUNCTION
identifies a base or dependent function.

++PTF
identifies a PTF.

++USERMOD
identifies a USERMOD.

Environment Definition MCSs

++IF
identifies a conditional requisite for the SYSMOD.

++VER
describes the environment required to install the SYSMOD.

Installation Definition MCSs

++ASSIGN
assigns a source ID to specified SYSMODs.

++DELETE
deletes a load module and any known alias names from a target library and from the associated target zone.

++JCLIN
identifies job control language (JCL) input that must be processed as part of installing the SYSMOD.

++MOVE
moves an element or load module (and any known aliases) from one library to another and automatically updates the associated target or distribution zone.

++RENAME
renames a load module in a target library and automatically updates the target zone.

Element Definition MCSs

++element
identifies a replacement for a data element. Data elements are elements other than macros, modules, and source code; for example, TSO CLISTs, help panels, and PARMLIB members.

Note: For a list of the data elements supported, see the description of the data element MCS in *SMP/E Reference*.

++MAC
identifies a macro replacement within a SYSMOD.

++MACUPD
identifies a macro update within a SYSMOD.

++MOD
identifies a module replacement within a SYSMOD.

++SRC
identifies a source replacement within a SYSMOD.

++SRCUPD
identifies a source update within a SYSMOD.

++ZAP
identifies a module update within a SYSMOD.

Exception SYSMOD MCSs

++HOLD
identifies a SYSMOD to be placed into exception SYSMOD status.

++NULL
acts as a placeholder and allows IBM service tapes to be built with the same format.

++RELEASE
removes a previously held SYSMOD from exception SYSMOD status.

Summary of SMP/E Reports

After SMP/E processes a command, it writes one or more reports that summarize the results. Following is a brief description of the information provided by the SMP/E reports. For a complete description of each report, see the chapter on SMP/E reports in *SMP/E Reference*.

- Causer SYSMOD Summary report[1]

  This report is produced during APPLY, ACCEPT, and RESTORE command processing when errors occur. It contains a list of causer SYSMODs and helpful information as to the errors that caused the SYSMODs to fail.

- CLEANUP Summary report

  This report is produced at the completion of CLEANUP processing and lists the elements or aliases that were deleted and the ddnames of the data sets where they were deleted.

- Cross-Zone Requisite SYSMOD report

  This report is produced when you run the REPORT CROSSZONE command. This report lists the SYSMODs that must be installed in a zone because of SYSMODs that are installed in another zone.

- Deleted SYSMOD report

  This report is produced at the completion of APPLY and ACCEPT processing and shows the function SYSMODs that were deleted and the service SYSMODs that were applicable to those functions. This report is only produced when a SYSMOD with the DELETE operand on its ++VER statement is processed.

- Dynamic Allocation report

  This report is produced at the completion of each SMP/E command (except SET, RESETRC, and DEBUG) and identifies all the DD statements used during the command, how each DD statement was obtained, and significant information about each DD statement.

- Element Summary report

  This report is produced at the completion of APPLY, ACCEPT, and RESTORE processing and describes the status of the libraries that were updated for each element that was processed.

[1] The Causer SYSMOD Summary report is issued only when the root cause SPE (PTF UR90259 for FMID HMP1600 and, if the Japanese feature is installed, PTF UR90260 for FMID JMP1611) is installed.

- Exception SYSMOD report

  This report is produced when you run the REPORT ERRSYSMODS command. This report lists SYSMODs that have already been installed but that are affected by error holds subsequently received.

- GENERATE Summary report

  This report is produced during GENERATE processing to summarize the jobs that were created.

- JCLIN Cross-Reference report

  This report is produced at the completion of JCLIN processing and lists each ASSEM module, macro, LMOD, SRC, and DLIB and the pages in the JCLIN Summary report where each appears.

- JCLIN Summary report

  This report is produced during JCLIN processing and lists the changed, added, and deleted entries for the CSI, and the EXEC statements that were ignored because the program or procedure was not recognized.

- LIST Summary report

  This report is produced during LIST processing to summarize which entries were or were not found in the zone being checked.

- MOVE/RENAME/DELETE report

  This report is produced during the processing of a SYSMOD containing ++MOVE, ++DELETE, or ++RENAME modification control statements. The report lists the elements and load modules that were moved, deleted, or renamed.

- RECEIVE Exception SYSMOD report

  This report is produced at the completion of RECEIVE processing and lists the ++HOLD and ++RELEASE statements processed.

- RECEIVE Summary report

  This report is produced at the completion of RECEIVE processing and lists those SYSMODs processed from the SMPPTFIN data set.

- REJECT Summary report

  This report is produced at the completion of REJECT processing and summarizes:

  - SYSMODs and FMIDs that were successfully or unsuccessfully deleted
  - Zones that were used during REJECT processing

  Also, if SMP/E could not reject a SYSMOD, the report explains why the SYSMOD was not rejected.

- SOURCEID report

  This report is produced when you run the REPORT SOURCEID command. It shows which source IDs have been assigned to SYSMODs in a given zone or ZONESET.

- SYSMOD Comparison report

  This report is produced when you run the REPORT SYSMODS command. It shows which SYSMODs (that have been installed in one zone) are applicable to, but not yet installed in, a second zone.

- SYSMOD Regression report

  This report is produced at the completion of APPLY and ACCEPT processing and summarizes the elements that were regressed when the BYPASS(ID) operand was specified. This report is only generated when BYPASS(ID) is specified, and at least one element is regressed.

- SYSMOD Status report

  This report is produced at the completion of APPLY, ACCEPT, and RESTORE processing and summarizes the processing that occurred for every eligible SYSMOD.

- UNLOAD Summary report

  This report is produced during UNLOAD processing to summarize which entries were or were not found in the zone being checked.

- ZONEEDIT Summary report

  This report is produced at the completion of ZONEEDIT processing and lists all the DDDEF and UTILITY entries that were changed.

- ZONEMERGE report

  This report is produced as the ZONEMERGE command is processing and shows if each entry was successfully merged or copied.

Chapter 2. Data Sets Used by SMP/E

For SMP/E to install SYSMODs, it needs a great deal of information about the SYSMODs and the target and distribution libraries where they are to be installed. This information is kept in a database composed of two primary data sets, and of one data set to back up entries from one of the primary data sets. These are:

- The CSI
- The PTS
- The SCDS.

The CSI is a VSAM data set; the PTS and SCDS are partitioned data sets. Subsequent sections of this chapter describe these data sets.

For users familiar with SMP4, Table 2 shows the relationships between the SMP/E and SMP4 data sets.

Table 2. SMP/E and SMP4 Data-Set Relationships

SMP4 Data Set	**SMP4 Entry**	**SMP/E Data Set**	**SMP/E Zone**
PTS	MCS	PTS	–
PTS	SYSMOD	CSI	Global
PTS	SYSTEM	CSI	Global
SMPACDS	All	CSI	DLIB
SMPACRQ	All	CSI	DLIB
SMPCDS	All	CSI	Target
SMPCRQ	All	CSI	Target
SCDS	All	SCDS	–

Note: For users who are currently at SMP/E Release 5 or SMP/E Release 5.1, no conversion is needed. The only step required is to run the SMP/E RECEIVE, APPLY, and ACCEPT commands to install SMP/E Release 6, which deletes your previous release from the system.

Because SMP/E Release 6 is packaged using data elements, users who are currently at SMP4 or SMP/E Release 4 or earlier cannot migrate directly to SMP/E Release 6. However, you can order a CBIPO MVS feature that includes SMP/E Release 6. Install the CBIPO feature. Then convert any data sets from your previous release that will be used with SMP/E Release 6. (This is done with the SMP/E CONVERT command.) Once you have converted this information, the new format can be processed only by SMP/E Release 6. The original data sets, however, can still be processed by your current release of SMP/E or SMP4.

These are the data sets used by SMP/E Release 6:

- **CSI data sets**. If you are currently using SMP/E Release 5 or SMP/E Release 5.1, your CSI data set can be used in its present format. No conversion is necessary.

  If you are currently using SMP/E Release 4, or any previous release, you cannot use your current CSI data set with SMP/E Release 6. If you want to use the CSI data with your new version of SMP/E, you must convert it to

SMP/E Release 6 format. This is done with the SMP/E CONVERT command. For more information about using the CONVERT command, see the SMP/E program directory and *SMP/E Reference.*

If you are currently using SMP4, you cannot use your SMP4 data sets with SMP/E. (These include SMPACDS, SMPACRQ, SMPCDS, and SMPCRQ.) If you want to use the data in your SMP4 data sets with SMP/E, you must convert those data sets into CSI data sets. This is done with the SMP/E CONVERT command. For more information about using the CONVERT command, see the SMP/E program directory and *SMP/E Reference.*

- **PTS data sets**. If you are currently using any release of SMP/E, you can use your current PTS data set with SMP/E Release 6.

  If you are currently using SMP4, you cannot use your SMP4 PTS with SMP/E. If you want to use the data in your SMP4 PTS with SMP/E, you must convert it into SMP/E format. This is done with the the SMP/E CONVERT command. For more information about using the CONVERT command, see the SMP/E program directory and *SMP/E Reference.*

- **SCDS data sets**. If you are currently using SMP/E Release 5 or SMP/E Release 5.1, your SCDS data set can be used by SMP/E Release 6. No conversion is necessary.

  If you are currently using SMP/E Release 4, or any previous release, or SMP4, you cannot use your current SCDS data set with SMP/E Release 6. If you want to use the SCDS data with your new release of SMP/E, you must convert it into SMP/E Release 6 format. This is done with the SMP/E CONVERT command. For information about using this command, see the SMP/E program directory and *SMP/E Reference.*

CSI Data Sets

SMP/E uses CSIs to keep records of the system. These data sets are used in much the same way SMP4 data sets were used. CSI data sets are logically divided into zones. These zones describe the various subsystems and products in the system. There are three types of zones:

- **Global Zone**—This type of zone is similar to the information kept in the SMP4 PTS. It describes the associated target and distribution zones and the SYSMODs waiting to be installed, and gives information about system utilities SMP/E calls to install SYSMODs.
- **Target Zone**—This type of zone is similar to the SMP4 CDS and CRQ. It describes the structure and contents of a set of target libraries and names the related distribution zone. (The related distribution zone is used when SMP/E is restoring a SYSMOD and needs to check the level of the elements in the distribution libraries.)

- **Distribution Zone**—This type of zone is similar to the SMP4 ACDS and ACRQ. It describes the structure and contents of a set of distribution libraries and names the related target zone. (The related target zone is used when SMP/E is accepting a SYSMOD and needs to check if the SYSMOD has already been applied.)

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. 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 approaches to organizing CSI data sets:

- Single-CSI structure
- Multiple-CSI structure.

Single-CSI Structure

You may 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 optimizes direct access storage use.
- The single CSI 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.

However, there is a drawback to single-CSI systems. 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 that contains 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 2 shows a single-CSI data set structure.

```
//SMPCSI   DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
//TGT1     DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
//TGT2     DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
//DLB1     DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
//DLB2     DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
```

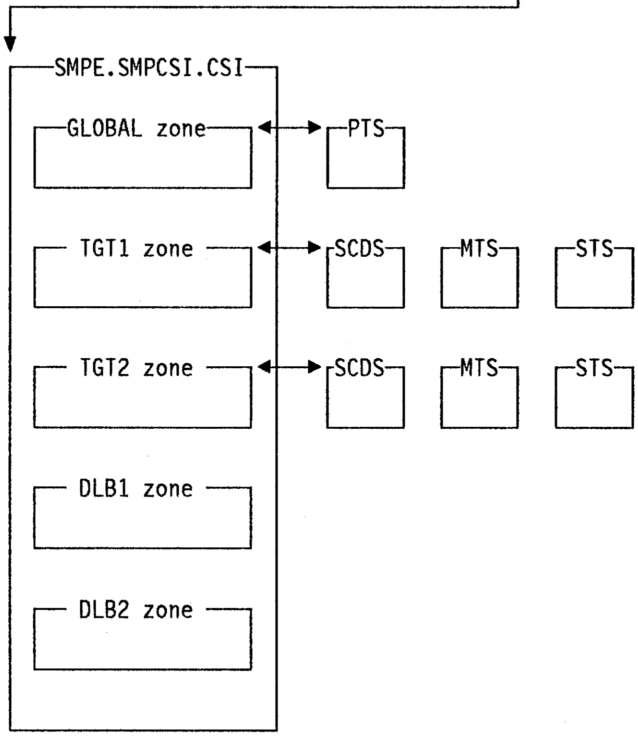


Figure 2. Sample Single-CSI Structure

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.)

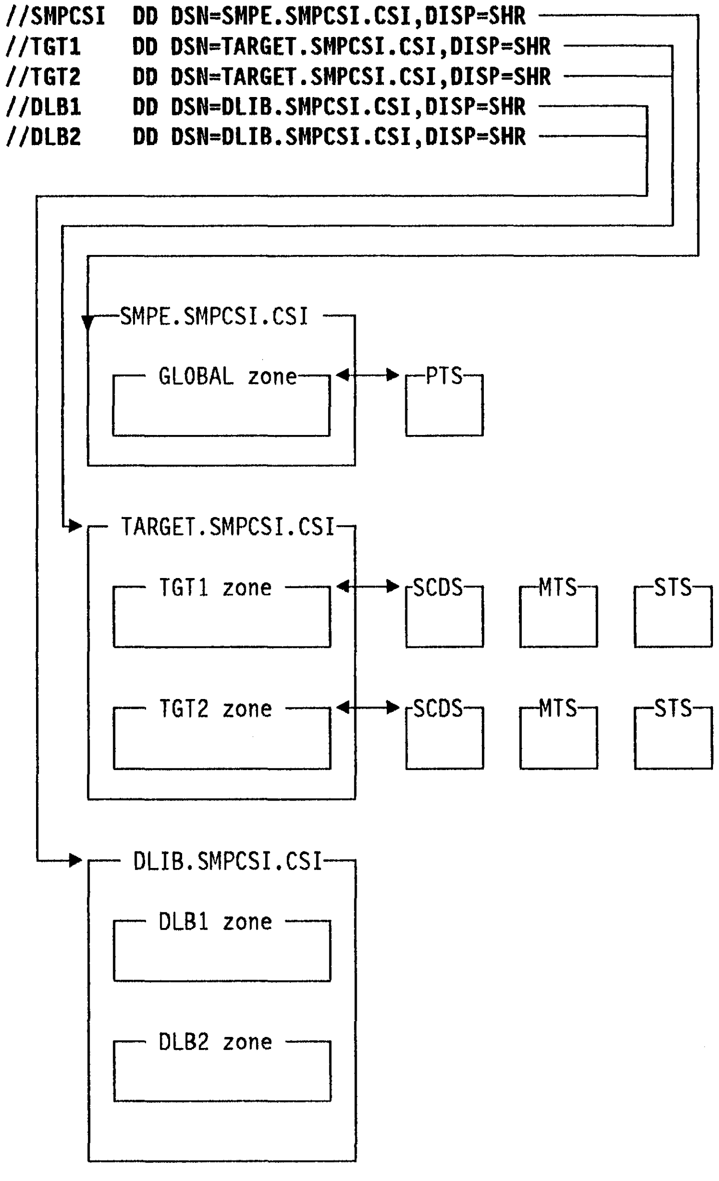
Multiple CSIs separate the global, target, and distribution zones to more than one VSAM data set. These are some reasons for having multiple CSI data sets:

- Your system may need multiple CSIs because of a particular installation's programming support, its backup and update needs, and the added security and data integrity of the installation.

  For example, it is easier to keep libraries and their associated zones synchronized when you dump them for backup if you keep them on the same physical DASD.
- Your system may need multiple CSIs because of the physical location of the support teams for different products, such as MVS, CICS/VS, IMS/VS, and NCP.

- 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 that contains that zone. If your zones are in separate CSI data sets, processing for one zone does not prevent access to another zone.

Figure 3 shows a multiple-CSI data set structure.



```
//SMPCSI  DD DSN=SMPE.SMPCSI.CSI,DISP=SHR
//TGT1    DD DSN=TARGET.SMPCSI.CSI,DISP=SHR
//TGT2    DD DSN=TARGET.SMPCSI.CSI,DISP=SHR
//DLB1    DD DSN=DLIB.SMPCSI.CSI,DISP=SHR
//DLB2    DD DSN=DLIB.SMPCSI.CSI,DISP=SHR
```

Figure 3. Sample Multiple-CSI Structure

Catalog Considerations

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

- **ICF catalogs:** If you use ICF 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.
- **VSAM catalogs:** If you use VSAM catalogs, the date and time stamp on each CSI must match the date and time stamp in the catalog entry for that CSI. Therefore, you should catalog the CSI in a user catalog that is on the same DASD volume as the CSI, and not in the master catalog. This ensures that if the volume is dumped, the catalog information is carried with the dump, and the date and time stamp in the catalog match the date and time stamp in the data set. If a CSI data set is cataloged in a user catalog, the high-level data set name qualifier must not be *SYS1*. For this reason the name of the CSI data set in the example starts with *SMPE* rather than *SYS1*. If a data set whose name starts with *SYS1* were cataloged in a user catalog, you could not use an alias in the master catalog to point to the data sets in the user catalog.
- **Alias entries for user catalogs:** Catalog information should be accessible through your master catalog. This is true for both VSAM and ICF catalogs. One way to do this is to define each user catalog as an alias in the master catalog. This allows you to access all the CSI data sets without having to use a STEPCAT DD statement. In addition, you would not have to restore both the DASD containing the master catalog and the DASD containing the CSI if an I/O error occurs. For an example of defining an alias for a user catalog, see "Defining an Alias Entry for a User Catalog" on page 24. If no alias is defined for the user catalog, you must include a STEPCAT DD statement each time SMP/E is called.

Allocating a CSI Data Set

CSI data sets are keyed VSAM clusters and are allocated using access method services. For additional information and a description of the access method services parameters, see the access method services manual for your operating system. Figure 4 on page 23 is a sample job for allocating a CSI data set. This job allocates the CSI data set with enough space to have multiple target or distribution zones.

Note: If you plan to convert SMP4 data sets into the CSI data set, allocate the CSI data set with 20% more free space than you would otherwise. SMP/E needs more space to receive the converted data and to allow for control interval splits that occur during conversion processing. For more information about converting from SMP4, see *SMP/E Reference*.

```
//DEFINE   JOB   'accounting info',MSGLEVEL=(1,1)
//STEP01   EXEC  PGM=IDCAMS
//CSIVOL   DD    UNIT=3380,VOL=SER=volid1,DISP=SHR
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
  DEFINE CLUSTER(
                 NAME(SMPE.SMPCSI.CSI)        -
                 FREESPACE(10 5)              -
                 KEYS(24 0)                   -
                 RECORDSIZE(24 143)           -
                 SHAREOPTIONS(2 3)            -
                 UNIQUE                       -
                 VOLUMES(volid1)              -
                )                             -
            DATA(                             -
                 NAME(SMPE.SMPCSI.CSI.DATA)   -
                 CONTROLINTERVALSIZE(4096)    -
                 CYLINDERS(250 20)            -
                )                             -
           INDEX(                             -
                 NAME(SMPE.SMPCSI.CSI.INDEX)  -
                 CYLINDERS(5 3)               -
                 IMBED                        -
                )                             -
         CATALOG(user.catalog)
/*
```

Figure 4. Access Method Services Job for Allocating a 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 instead of 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.

  SMP/E does not support cross-system sharing of the CSI, so you cannot specify **4** as the cross-system value for SHAREOPTIONS. If you want to support cross-zone sharing, you must use either Global Resource Serialization (GRS), or a similar product, or control access yourself.

- UNIQUE

  If you are using VSAM, be sure to use the UNIQUE parameter. That way, the CSI cluster is assigned the name specified in the NAME parameter, rather than a VSAM-assigned name.

  Note: The UNIQUE parameter refers only to the CSI cluster. It is used to get unique data space.

  If your CSI is in an ICF catalog, you do not need to specify UNIQUE. If you do specify it, it is ignored.

- CONTROLINTERVALSIZE (CISIZE)

  If you allocate more than one CSI, give each CSI the same control interval (CI) size.

  Note: If you are operating in a multiple CSI environment, make sure of the following:

  - That the index CI sizes are equal.
  - That the data CI sizes are equal.

- CYLINDERS.

  The DATA and INDEX CYLINDERS values shown in Figure 4 on page 23 are only **estimated** starting values for 3380s. Your cylinder values may vary based on the device type, software arrangement, 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. This lets you access the new CSI data sets without having to use a STEPCAT DD statement.

Figure 5 is an example of a job for creating 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)
/*
```

Figure 5. Access Method Services Job for Creating an Alias Entry in the Master Catalog

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

Initializing a CSI Data Set

Before you can use a CSI, you must initialize it with the GIMZPOOL record, which is in SYS1.MACLIB. Use the access method services job shown in Figure 6 to initialize the newly allocated CSI data set.

```
//ALLOC    JOB  'accounting info',MSGLEVEL=(1,1)
//AMS      EXEC PGM=IDCAMS
//SMPCSI   DD   DSN=SMPE.SMPCSI.CSI,DISP=OLD
//ZPOOL    DD   DSN=SYS1.MACLIB(GIMZPOOL),DISP=SHR
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
           REPRO OUTFILE(SMPCSI) -
           INFILE(ZPOOL)
/*
```

Figure 6. Access Method Services Job for Initializing a CSI

Notes:

1. It is possible to use the access method services REPRO command to copy a complete CSI from one data set to another. If you have allocated a CSI in preparation for using the REPRO command to copy in an existing CSI, do **not** initialize your newly allocated CSI with the GIMZPOOL record. A GIMZPOOL record is copied in when the existing CSI is copied into your new CSI. If you initialized the new CSI with the GIMZPOOL record, it would contain two GIMZPOOL records after REPRO processing, instead of just the one required.
2. Make sure you use the SMP/E Release 6 GIMZPOOL record to initialize CSIs that you will use with SMP/E Release 6.

Reorganizing a CSI Data Set to Reclaim Space

During normal SMP/E processing, VSAM control interval splits and control area splits can occur. This uses up some of the free space in each control interval or control area, which can degrade SMP/E performance and DASD space utilization. You should monitor your VSAM data sets on a regular basis to determine how many splits have occurred and how much free space remains. Figure 7 is an example of a job for listing 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
/*
```

Figure 7. Access Method Services Job for Listing the Catalog Entry

After examining the LISTCAT output, you may determine that the CSI should be reorganized to eliminate the control interval splits or control area splits 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, based on your experience and knowledge of VSAM.

Figure 8 is an example of a job for exporting the CSI. After exporting the CSI, you can allocate a new CSI as shown in Figure 4 on page 23. Figure 9 is an example of a job for importing the CSI.

```
//EXPORT   JOB    'accounting info',MSGLEVEL=(1,1)
//STEP01   EXEC   PGM=IDCAMS
//SMPCSI   DD     DSN=SMPE.SMPCSI.CSI,DISP=OLD
//TAPECSI  DD     DSN=SMPCSI.DATA,UNIT=TAPE,DISP=NEW,
//                VOL=SER=xxxxxx,LABEL=(1,SL)
//SYSPRINT DD     SYSOUT=A
//SYSIN    DD     *
  EXPORT SMPE.SMPCSI.CSI  -
         INFILE(SMPCSI)   -
         OUTFILE(TAPECSI)
/*
```

Figure 8. Access Method Services Job for Exporting a CSI

```
//IMPORT   JOB    'accounting info',MSGLEVEL=(1,1)
//STEP01   EXEC   PGM=IDCAMS
//SMPCSI   DD     DSN=SMPE.SMPCSI.CSI,DISP=OLD
//TAPECSI  DD     DSN=SMPCSI.DATA,UNIT=TAPE,DISP=OLD,
//                VOL=SER=xxxxxx,LABEL=(1,SL)
//SYSPRINT DD     SYSOUT=A
//SYSIN    DD     *
  IMPORT  INFILE(TAPECSI)  -
          OUTFILE(SMPCSI)  -
          INTOEMPTY
/*
```

Figure 9. Access Method Services Job for Importing a CSI

Notes:

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

   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. A sequential data set must be used to receive the exported CSI.

3. After allocating a new CSI to be imported into, you should **not** prime it with the GIMZPOOL record provided in SYS1.MACLIB; otherwise, the import operation will fail.

Entries in CSI Data Sets

Once you have allocated and initialized the CSI data sets, you need to create entries in those data sets that SMP/E will use to maintain your system. Some of these entries must be created (such as by UCLIN or the SMP/E dialogs) before you can do any other processing for the zone. Other entries are automatically created when you install SYSMODs, when you run the JCLIN command, or when you copy entries from one zone to another.

To help you define entries, SMP/E provides a member in SYS1.SAMPLIB (GIMSAMPU) that contains sample UCL statements to define entries for a basic MVS system. You can access this member using 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 *SMP/E Reference*. The chapter on the UCLIN command shows the UCLIN syntax for each entry type, and the chapter on SMP/E data set entries contains a description of the syntax plus examples and special considerations.

Following are descriptions of the entries that may exist in the global zone and in the target and distribution zones.

Global Zone Entries

The following entries may be defined in the global zone:

- The GLOBALZONE entry
- DDDEF entries
- FMIDSET entries
- HOLDDATA entries
- OPTIONS entries
- SYSMOD entries
- UTILITY entries
- ZONESET entries.

The 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 is created by UCLIN or the SMP/E dialogs and must be defined before you can do any other processing for that global zone.

DDDEF Entries: The DDDEF entry contains the information SMP/E needs to dynamically allocate a specific data set. It is created by UCLIN or the SMP/E dialogs. With DDDEF entries, you do not have to provide a DD statement for every data set SMP/E may need to process a particular command. When SMP/E determines that it needs a specific data set, it looks for a DD statement that it can use to allocate that data set. If there is no DD statement, SMP/E checks if the current zone contains a DDDEF entry for that data set. If so, it uses the information in the DDDEF entry to dynamically allocate the data set. For more information about dynamically allocating data sets, see "Dynamically Allocating Data Sets" on page 40.

FMIDSET Entries: The FMIDSET entry defines a group of FMIDs that are to be used to limit the SYSMODs processed by an SMP/E command. For example, you could specify an FMIDSET on the APPLY command to process only SYSMODs applicable to one of the FMIDs in the FMIDSET. The FMIDSET entry is created by UCLIN or the SMP/E dialogs.

HOLDDATA Entries: The HOLDDATA entry contains ++HOLD statements that either were received from SMPHOLD (external HOLDDATA) or were within a SYSMOD that was received (internal HOLDDATA). It may be created during RECEIVE processing or by UCLIN or the SMP/E dialogs. The HOLDDATA entry is a record of the ++HOLD statements and is not used by SMP/E to control exception SYSMOD processing, but it is used by other commands, such as LIST. However, when SMP/E saves these ++HOLD statements, it also creates HOLDDATA subentries in the associated global zone SYSMOD entry. SMP/E uses these HOLDDATA subentries to control exception SYSMOD processing. For information about managing exception SYSMODs, see Chapter 8.

OPTIONS Entries: The OPTIONS entry defines processing options that are to be used for an SMP/E command or set of commands. For example, if you want to define the processing options to be used by specific utility programs during APPLY processing, you define an OPTIONS entry that points to the UTILITY entries that contain those options. The OPTIONS entry is created by UCLIN or the SMP/E dialogs.

Although OPTIONS entries exist only in the global zone, they are also used to process commands for target and distribution zones. There are two ways you can specify which OPTIONS entry should be in effect when you are processing a zone:

- In the GLOBALZONE, TARGETZONE, and DLIBZONE entries. The OPTIONS entry specified here is the default OPTIONS entry for that zone.
- On the SET command. The name specified on the SET command overrides the default OPTIONS name.

When SMP/E processes a command, it checks these sources to determine which OPTIONS entry should be in effect for that command. If SMP/E cannot find a reference to an OPTIONS entry, it uses defaults for some of the information.

SYSMOD Entries: The global zone SYSMOD entry describes a SYSMOD that exists as an MCS entry in the PTS. It may be created during RECEIVE processing or by UCLIN or the SMP/E dialogs, and contains information SMP/E obtained when it received that SYSMOD.

When SMP/E processes SYSMODs, it uses the global zone SYSMOD entry to determine which SYSMODs (of those you selected) are applicable. In addition, after a SYSMOD has been successfully applied or accepted, SMP/E records in the SYSMOD entry the names of the zones to which the SYSMOD was applied or accepted.

UTILITY Entries: The UTILITY entry contains information SMP/E uses when invoking a particular system utility program, such as the load module name of the program and the parameters that should be passed. It is created by UCLIN or the SMP/E dialogs.

Information in a UTILITY entry is used only if the UTILITY entry is named in the OPTIONS entry that is in effect. For example, to have SMP/E use certain parameters when it calls a specific linkage editor, you must do the following:

1. Define a UTILITY entry that names the program and specifies the parameters.
2. Define an OPTIONS entry that specifies that UTILITY entry as the UTILITY entry for the linkage editor.
3. Put that OPTIONS entry into effect, either by specifying it on the SET command or by defining it as the default OPTIONS entry for the zone being processed.

If the OPTIONS entry does not point to a UTILITY entry for a particular system utility, SMP/E uses default values for that utility.

ZONESET Entries: The ZONESET entry defines a group of zones that are to be used to limit the SYSMODs processed by an SMP/E command. For example, you could specify a ZONESET on the REPORT CROSSZONE command to define which zones SMP/E should check. The ZONESET entry is created by UCLIN or the SMP/E dialogs.

Target Zone and Distribution Zone Entries

The following entries may be defined in a target or distribution zone:

- The TARGETZONE entry (target zones only)
- The DLIBZONE entry (distribution zones only)
- ASSEM entries
- Data element entries
- DDDEF entries
- DLIB entries
- LMOD entries
- MAC entries
- MOD entries
- SRC entries
- SYSMOD entries.

The 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 is created by UCLIN or the SMP/E dialogs and must be defined before you can do any other processing for that target zone.

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

ASSEM Entries: The ASSEM entry contains assembler statements that can be assembled to create an object module. Generally, it is created during JCLIN processing when SMP/E encounters an assembler step with inline assembler input. When the module is reassembled by use of the statements in the ASSEM entry, SMP/E copies those statements into a work data set and then assembles the module.

If a macro is called in the assembly, the ASSEM entry is pointed to by the MAC entry created for that macro. As a result, when that macro is updated, SMP/E can reassemble the affected module, using the statements in the ASSEM entry.

Data Element Entries: The data element entry describes an element that is not a macro, module, or source code. Data elements may exist in distribution libraries, target libraries, or both. Generally, a data element entry is created by installing a SYSMOD containing an MCS for a data element that does not yet have an entry.

SMP/E records the function and service level of the data element in the entry. Once a data element entry exists, it is updated as subsequent SYSMODs that affect the data element are installed.

DDDEF Entries: The DDDEF entry contains the information SMP/E needs to dynamically allocate a specific data set. It is created by UCLIN or the SMP/E dialogs. With DDDEF entries, you do not have to provide a DD statement for every data set SMP/E may need to process a particular command. When SMP/E determines that it needs a specific data set, it looks for a DD statement that it can use to allocate that data set. If there is no DD statement, SMP/E checks if the current zone contains a DDDEF entry for that data set. If so, it uses the information in the DDDEF entry to dynamically allocate the data set. For more information about dynamically allocating data sets, see "Dynamically Allocating Data Sets" on page 40.

DLIB Entries: The DLIB entry describes a distribution library that was totally copied to a target library. It is created during JCLIN processing when SMP/E encounters a COPY statement without the SELECT option. The INDD value on the COPY statement becomes the name of the DLIB entry, and the OUTDD value on the COPY statement becomes the target library specified in the DLIB entry.

SMP/E uses the DLIB entry to determine where elements should be installed when the element entry does not specify a target library. SMP/E looks for a DLIB entry whose name matches the distribution library of the element. It then installs the element in the indicated target library.

LMOD Entries: The LMOD entry contains all the information needed to replace or update a given load module. This includes such information as whether the load module is link-edited or copied during system generation, any link-edit statements required to relink the load module, the link-edit attributes of the load module, and the libraries in which it resides. Generally, an LMOD entry is created by one of the following methods:

- Installing a SYSMOD that adds the load module.

  LMOD entries may be created when a SYSMOD is installed. SMP/E builds an LMOD entry if it encounters a ++MOD statement for a load module that does not yet have an LMOD entry and if the distribution library for the module was totally copied.

- Processing JCLIN.

  LMOD entries may be created during JCLIN processing when SMP/E scans the copy and link-edit steps. At the same time, SMP/E builds MOD entries for modules that are linked or copied to the load module.

MAC Entries: The MAC entry describes a macro that exists in the distribution or target libraries. A MAC entry is generally created by one of the following methods:

- Installing a SYSMOD that contains the macro.

  MAC entries are created the first time you install a SYSMOD that contains a ++MAC statement for a macro that does not yet have a MAC entry.

- Processing JCLIN.

  MAC entries may be built during JCLIN processing when SMP/E scans the assembler statements for inline assemblies to determine which macros are used in that assembly. The name of the assembly is kept in the MAC entry so that when the macro is updated, SMP/E can reassemble the required modules.

  MAC entries may also be built when SMP/E scans copy steps and finds a SELECT statement that specifies TYPE=MAC.

SMP/E records the function and service level of the macro in the MAC entry, as well as information about how that macro affects the structure of the distribution or target libraries and modules. Once a MAC entry exists for a macro, it is updated as subsequent SYSMODs that affect the macro are installed.

MOD Entries: The MOD entry describes a particular module, its function and service level, and how it relates to a load module in the target library. A MOD entry is generally created by one of the following methods:

- Installing a SYSMOD that contains the module.

  MOD entries are created the first time you install a SYSMOD that contains a ++MOD statement for a module that does not yet have a MOD entry. If the module comes from a copied distribution library, SMP/E also builds an LMOD entry with the same name.

  A MOD entry may also be created when you install a SYSMOD that causes the assembled source code to be linked to the distribution library. This can happen when a SYSMOD contains both a ++MOD and a ++SRC statement for the same module, or when the DISTMOD operand is specified on the ++SRC statement.

- Processing JCLIN.

  MOD entries may be built during JCLIN processing when SMP/E scans the link-edit and copy steps. Whenever a MOD entry is built during JCLIN processing, it is always associated with an LMOD entry. Thus, when the module is changed, SMP/E can determine which load modules are affected.

SMP/E records the function and service levels of each module in the MOD entry, as well as information about how that module affects the structure of the distribution or target library. Once a MOD entry exists for a module, it is updated as subsequent SYSMODs that affect the module are installed.

SRC Entries: The SRC entry describes source code that exists in the distribution or target libraries. (SMP/E assumes that for each SRC entry in a particular zone there exists a MOD entry with the same name.) Generally, an SRC entry is created by one of the following methods:

- Installing a SYSMOD that contains the source.

  SRC entries are created the first time you install a SYSMOD that contains a ++SRC statement for source code that does not yet have an SRC entry.

- Processing JCLIN.

  SRC entries may be built during JCLIN processing when SMP/E scans the assembler step and determines that the assembler input is a member of a partitioned data set.

  SRC entries may also be built when SMP/E scans copy steps and finds a SELECT statement that specifies TYPE=SRC.

SMP/E records the function and service levels of the source code in the SRC entry, as well as information about how that source code affects the structure of the distribution or target libraries and modules. Once an SRC entry exists for source code, it is updated as subsequent SYSMODs that affect the source code are installed.

SYSMOD Entries: The SYSMOD entry describes a SYSMOD that has been installed in a distribution or target library. It may be created during APPLY or ACCEPT processing, or by UCLIN or the SMP/E dialogs. This SYSMOD entry contains the same information as the global zone SYSMOD entry, except that it has information from only one ++VER statement, the one used to install the SYSMOD.

When SMP/E installs a SYSMOD, it uses SYSMOD entries in the distribution or target zone to:

- Determine the functional level of the system. SMP/E checks which function SYSMODs have been installed and then uses that information to determine which service SYSMODs may be applicable.
- Determine the service level of the system. SMP/E checks which service SYSMODs have been installed.
- Make sure the requisites are satisfied for each SYSMOD to be installed. SMP/E checks if a SYSMOD entry exists in the distribution or target zone for each requisite.

PTS Data Sets

The PTS data set is used as temporary storage for SYSMODs. It contains one member for each SYSMOD that was received. Each member is called an *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 SYSMOD ID of the SYSMOD it contains. Generally, the MCS entries are kept on the PTS until the SYSMOD is accepted, after which, under normal processing, they are deleted.

The PTS is associated with the global zone. Therefore, you should allocate one PTS data set for each global zone you define. The space and directory block allocation for the SMP/E PTS depends on your new function installation plans and maintenance activity. In general, it should be allocated with half the number of directory blocks and three-quarters as much space as the SMP4 PTS. For more information about allocating the PTS data set, see *SMP/E Reference*.

SCDS Data Sets

The SCDS data set 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.

The SCDS contains SYSMOD, ASSEM, LMOD, DLIB, and element entries. Each entry is associated with only one SYSMOD. 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.

An SCDS must exist for each target zone within a CSI. 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 that any normal partitioned data set is. The space and directory-block allocation for this data set depends on your new-function installation plans and maintenance activity. For more information about allocating the SCDS data set, see *SMP/E Reference*.

Chapter 3. Preparing to Use SMP/E

This chapter describes how to prepare for using SMP/E. It discusses the following:

- Calling SMP/E
- Dynamically allocating data sets
- System utility programs used by SMP/E
- Considerations for backing up data sets.

Calling SMP/E

SMP/E can be called by:

- Using the SMP/E dialogs.
- Submitting a background job that calls GIMSMP, the program name for SMP/E. This job may call SMP/E directly or through a cataloged procedure.

This section describes the types of information you need to provide if you use a cataloged procedure to call SMP/E. It discusses the following:

- Required JCL statements
- 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 call SMP/E:

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

The **JOB statement** describes your installation-dependent parameters. It may also include the REGION parameter to set the size of the region in which SMP/E runs. Formulas for estimating the region size for SMP/E are provided in *SMP/E Reference.*

The **EXEC statement** must specify PGM=GIMSMP or the name of your cataloged procedure for calling SMP/E. The following may be specified in the EXEC statement PARM parameter:

CSI=*dsname*

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

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

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

- U or IPL—To use the IPL date of the system.
- REPLY—To request the date from the operator. SMP/E issues message GIM399 as a result.
- *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.

LANGUAGE=*xxx*
where *xxx* can be one of the following:

- ENU—U.S. English
- JPN—Japanese.

The LANGUAGE option defines which language to use for SMP/E messages.

LANGUAGE can also be specified as **L**. If **LANGUAGE** is not specified, the default is LANGUAGE=ENU.

If you have installed the Japanese feature of SMP/E, you may specify **LANGUAGE=JPN** or **LANGUAGE=ENU**, depending on whether SMP/E messages are required in Japanese or in English. However, Japanese users are not required to install both features.

The output devices used must support the selected language and English single-byte characters. SMP/E does not check to verify that the output devices provide this support.

PROCESS=WAIT or
PROCESS=END
The PROCESS parameter is used to control the action SMP/E should take if a CSI or PTS data set is not immediately available (because it is currently in use with another job or is in use by a dialog).

- WAIT causes the job to wait for the CSI until it 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 CSI data set is still not available after the 10-minute wait, the command that needs the data set is stopped.

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

See *SMP/E Reference* for more information on obtaining and sharing CSI data sets.

DD statements define the data sets that may be used in SMP/E processing. See the chapters on individual SMP/E commands in *SMP/E Reference* for information about which data sets are required for each command.

Note: DDDEF entries may be used instead of DD statements to allocate many of the necessary data sets. See "Dynamically Allocating Data Sets" on page 40 for more information.

Sample Cataloged Procedure for SMP/E

Figure 10 on page 38 is a sample cataloged procedure, SMPPROC, that may be used to run SMP/E. The numbers to the left of the statements correspond to the notes that follow the example. Remember these considerations when you write a cataloged procedure for SMP/E:

- Your own cataloged procedure should be tailored 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 may have the SMPPTFIN DD statement defined and have no DD statements for the target and distribution libraries.

  Note: The SYSLIB concatenation for APPLY and ACCEPT should point to different libraries.
- Most of the data sets in the cataloged procedure can be allocated without using DD statements. If you use the methods described for the data sets listed below, you may not need a cataloged procedure.
  - **Master CSI.** The master CSI data set can be specified on the CSI= parameter of the EXEC statement for GIMSMP, instead of on the SMPCSI DD statement. For more information about parameters you may specify on the EXEC statement, see "Required JCL Statements" on page 35.
  - **Target and Distribution Zones.** CSI data sets for target and distribution zones can be dynamically allocated with zone indexes in the global zone, instead of with zone DD statements. For an example of defining zone indexes, see *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 shorter cataloged procedure than the one in Figure 10. For more information about DDDEF entries, see *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 is required to point to the commands that SMP/E processes.
  - An SMPPTFIN DD statement is required by the RECEIVE command to point to the source of the SYSMODs that are processed.
  - An SMPHOLD DD statement is required by the RECEIVE command to point to the source of the HOLDDATA that is processed.
  - An SMPTLIB DD statement is required if the SYSMODs being processed were packaged in relative files.

- An LKLIB DD statement is required by the APPLY and ACCEPT commands if any of the SYSMODs being installed contain elements packaged in an LKLIB data set.
- A TXLIB DD statement is required by the APPLY and ACCEPT commands if any of the SYSMODs being installed contain elements or JCLIN packaged in a TXLIB data set.

- 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 *SMP/E Reference*.

```
   //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 ------------------------------------
2  //SMPPTS   DD DSN=SYS1.SMPPTS,DISP=SHR
3  //SMPMTS   DD DSN=SYS1.SMPMTS,DISP=OLD
4  //SMPSTS   DD DSN=SYS1.SMPSTS,DISP=OLD
   //SMPSCDS  DD DSN=SYS1.SMPSCDS,DISP=OLD
   //SMPLOG   DD DSN=SYS1.SMPLOG,DISP=MOD
   //SMPLOGA  DD DSN=SYS1.SMPLOGA,DISP=MOD
5  //* -----  Master CSI -----------------------------------------
   //SMPCSI   DD DSN=SMPE.SMPCSI.CSI,DISP=SHR

6  //* -----  SMP/E temporary data sets----------------------------
   //SMPWRK1  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
   //            DCB=BLKSIZE=3360
   //SMPWRK2  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
   //            DCB=BLKSIZE=3360
7  //SMPWRK3  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,CATALOG),
   //            DCB=BLKSIZE=3200
   //SMPWRK4  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
   //            DCB=BLKSIZE=3200
   //SMPWRK6  DD UNIT=SYSDA,SPACE=(CYL,(2,1,5)),DISP=(,DELETE),
   //            DCB=BLKSIZE=3200

   //* -----  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 --------------------------
```

Figure 10 (Part 1 of 2). Sample SMP/E Cataloged Procedure

```
8  //SYSLIB   DD DSN=data set name,DISP=OLD
                    •  •  •
   //* -----  PARMLIB data set for JCLIN --------------------------
   //PARMLIB  DD DSN=SYS1.PARMLIB,DISP=OLD
   //* -----  Target libraries ------------------------------------
   //LINKLIB  DD DSN=SYS1.LINKLIB,DISP=OLD
                    •  •  •
   //* -----  Distribution libraries ------------------------------
   //AOSC5    DD DSN=SYS1.AOSC5,DISP=OLD
                    •  •  •
   //         PEND
```

Notes:

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

2 **PTS** should be coded with **DISP=SHR**. This allows concurrent jobs to share the PTS as much as possible.

See *SMP/E Reference* for more information on how SMP/E shares data sets.

3 **SMPMTS** is required for changes to macros that do not reside in a target library.

4 **SMPSTS** is required for changes to source code that does not reside in a target library.

5 **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. See *SMP/E Reference* for more information on how SMP/E shares data sets.

6 **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.

7 **SMPWRK3** may be permanently allocated in order to reuse assemblies. For more information, see the description of the REUSE operand under the APPLY command in *SMP/E Reference.*

8 **SYSLIB** concatenation depends on how you intend to use the distribution libraries. See *SMP/E Reference* for details on which data sets to include and in what order.

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.

Figure 10 (Part 2 of 2). Sample SMP/E Cataloged Procedure

Figure 11 is an example of how to call SMP/E using the cataloged procedure in Figure 10.

```
//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 */
/*
```

Figure 11. Sample Job Using SMP/E Catalog Procedure

Dynamically Allocating Data Sets

A variety of data sets are required to process SMP/E commands. You can either provide the DD statements for these data sets (such as in a cataloged procedure) or have SMP/E dynamically allocate the data sets. 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.

This section describes where SMP/E can get the information it needs to dynamically allocate data sets and how it chooses which of these sources to use.

Sources of Information for Dynamic Allocation

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

- DDDEF entries
- Module GIMMPDFT
- Standard defaults.

DDDEF Entries

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

- Permanent data sets (such as target libraries, distribution libraries, and SMP/E data sets)
- Temporary data sets

- SYSOUT data sets
- Work data sets.

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 will use the data set. DDDEF entries provide more flexibility than DD statements; they allow 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. For more information about DDDEF entries, see *SMP/E Reference*.

Module GIMMPDFT

Another way to provide SMP/E with information about data sets is through module GIMMPDFT, which is part of SMP/E. Unlike DDDEF entries, however, this information applies to all zones, not just to the set-to zone. (Although you can use GIMMPDFT to define temporary data sets, it is easier to tailor the definitions to your system if you use DDDEF entries instead.) GIMMPDFT contains two tables:

- Table 1 defines SYSOUT data sets.
- Table 2 defines temporary data sets (such as the SMPWRKx data sets and the SYSUTx data sets).

GIMMPDFT also contains information on space allocation values for SMPTLIB data sets. For more information about module GIMMPDFT, see the appendix on dynamic allocation in *SMP/E Reference*.

Standard Defaults

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

- SMPOUT is allocated as either 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, it checks to see if DD statements were provided for any of those data sets. If so, SMP/E allocates the data sets using information from the DD statements. Otherwise, it must dynamically allocate the data sets. To get the information it needs for dynamic allocation, SMP/E checks the following sources, in the order shown:

1. **DDDEF entries.** If the global zone, target zone, or distribution zone specified in the SET command contains a DDDEF entry for the required data set, SMP/E allocates the data set using that entry. Otherwise, it checks the next source.
2. **Table 1 in GIMMPDFT.** If table 1 defines the data set, SMP/E allocates the data set using that information. Otherwise, it checks the next source.
3. **Table 2 in GIMMPDFT.** If table 2 defines the data set, SMP/E allocates the data set using that information. Otherwise, it checks the next source.

4. **Standard defaults.** If the data set is for SMPOUT or SYSPRINT, SMP/E allocates the data set using the standard default. Otherwise, SMP/E reports that the required DD statement is missing.

System Utility Programs Used by SMP/E

The following list shows the default load module names defined for the utility programs used by, but not necessarily invoked by, SMP/E. These utilities must all reside in data sets in the LINKLST concatenation (as defined in SYS1.PARMLIB).

- If you do not define UTILITY entries and OPTIONS entries to specify which utility programs to use, SMP/E uses these default utility programs and its own default values for the parameters to be passed.
- If you want to use utility programs other than these, or if you want to specify different parameters for the default utility programs, you need to define UTILITY entries and OPTIONS entries to identify the programs to SMP/E. For more information, see the descriptions of UTILITY entries and OPTIONS entries in *SMP/E Reference*.
- If you want to restrict which utility programs SMP/E can use, you can specify this information in module GIMUTTBL, which is provided by SMP/E. For more information, see the appendix on restricting utility programs in *SMP/E Reference*.

ASMBLR
is the default used for assemblies.

Note: In an MVS/XA or MVS/ESA* environment, the assembler used **must** be IEV90.

IDCAMS
is the default used for access method services.

IEBCOPY
is the default used for copy operations, library compresses, and error recoveries after *x*37 abends.

IEBUPDTE
is the default used for updating.

IEHIOSUP
is used to install maintenance on a VS1 system.

IEWL
is the default used for link-edits.

Note: In an MVS/XA or MVS/ESA environment, the linkage editor used **must** be HEWLH096.

IMASPZAP
is the default used for superzap operations.

* MVS/ESA is a trademark of the IBM Corporation.

Backing Up Data Sets Used by SMP/E

SMP/E data sets, target libraries, and distribution libraries should be backed up every time they are updated by SMP/E. This is done so that you can recover your system in the event of a system failure or a head crash on a disk pack, and so on. You can back up either the devices that contain the data sets, or the individual data sets.

Device Backup

Devices not containing SMP/E VSAM data sets (CSIs) can be backed up in their entirety by use of Data Facility Data Set Services (DFDSS) Release 2, or higher, if you have it installed on your system.

Devices containing both VSAM and non-VSAM (or just VSAM) data sets can be backed up in their entirety by use of DFDSS with a STEPCAT DD statement pointing to your VSAM user catalog.

Individual Data Set Backup

Individual non-VSAM data sets can be backed up by use of IEBCOPY if you have it installed on your system.

Individual VSAM data sets can be backed up by use of DFDSS with a STEPCAT DD statement pointing to your VSAM user catalog or by use of the IDCAMS EXPORT and IMPORT commands.

Part 2. Guide to Using SMP/E

Chapter 4. Installing a New Function

This chapter discusses methods you can use to install a new function. It describes:

- Sources of new functions and sources of installation information
- How to choose an installation method
- An example of installing a function using the RECEIVE-APPLY-ACCEPT method.

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. In addition, some products provide special generation support that allows you to build a set of target libraries with the output of macro assemblies.

This chapter gives you a summary of 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 3 lists sources of new functions and places where you can find information for installing the new functions.

Table 3. Sources for Functions and Their Installation Information

Product Delivery Vehicle	Products and Service You Get	Installation Materials You Can Use
CBIPO	Customized DLIBs with integrated service SERV tape	Related installation materials (RIMs) Program directories for the products you ordered Other CBIPO documentation (these are listed in the preface) Installation manuals for the products you ordered (if provided)
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 manuals for the products you ordered
Independent product	Product tape CUM tape	Program directory for the product Installation manual for the product (if one is provided)

Methods for Installing Function SYSMODs

There are various methods of using SMP/E commands and product-supplied generation procedures to install function SYSMODs. Here are the procedures you might need to use:

- The RECEIVE-APPLY-ACCEPT method

  This is the **standard** installation method.

  The standard method of installation uses SMP/E RECEIVE, APPLY, and ACCEPT commands to install functions onto a system or 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.

- RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-GENERATE

  RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-Stage 2 Generation

  RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-APPLY

  RECEIVE-ACCEPT-Stage 1 Generation-DELETE-JCLIN-APPLY.

  These methods are **special generation** installation methods.

  A special generation method uses generation procedures (such as MVS SYSGEN, IMSGEN, CICS subsystem generation, or SMP/E product generation) with SMP/E commands to install functions. Usually, you need to use a special generation method when your function requires some kind of tailoring through a macro assembly. For example, your new product may be one that cannot be customized independently after it is installed. The JCLIN needed for the function is contained in the generation macros and procedures.

Choosing an Installation Method

The installation method you choose depends on the function you are installing and the system you are installing the function on. The following are points to consider to help you decide on an installation method:

- You should follow the installation method described in the program directory or installation manual (if one exists) for the function you are installing.
- Table 4 on page 51 provides a summary of the installation methods.
- The descriptions of the various installation methods on pages 51 to 57 provide more detail on each method.

Table 4. Comparison of Methods for Installing Functions

Method	Functions Processed	When to Use
RECEIVE-APPLY-ACCEPT	Functions that do not use generation macros and procedures for tailoring.	To update an existing system or subsystem with these functions, or to create or replace a system or subsystem that consists solely of these functions. **Note:** This method is used for installing many of the functions obtained from a CBPDO or that were individually ordered.
RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-GENERATE	Functions that are included by generation macros and procedures, and functions that must be reinstalled because they are not included by these macros and procedures.	To create or replace a system or subsystem with these functions by running job steps created and tailored by the SMP/E GENERATE command. **Note:** This method is used for installing functions included with a CBIPO.
RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-Stage 2 Generation (No JCLIN data is provided with the product.)	Functions that are included by special generation macros and procedures.	To create or replace a system or subsystem with these functions by running job steps created directly by these macros and procedures. **Note:** Any needed functions not included by these macros and procedures must be separately installed or reinstalled.
RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-APPLY	A function that: • Uses a generation macro and procedure • Does not delete another function.	To update an existing system or subsystem with this function.
RECEIVE-ACCEPT-Stage 1 Generation-DELETE-JCLIN-APPLY	A function that: • Uses a generation macro and procedure • Deletes another function.	To update an existing system or subsystem with this function.

RECEIVE-APPLY-ACCEPT Method

Use this method for installing a function that does not require special processing, such as when you are creating or replacing a system that consists of functions that do not need special generation macros. For more details on this installation method, see "Example of Using the Standard RECEIVE-APPLY-ACCEPT Method" on page 57.

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.

RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-GENERATE Method

Use this method if you are creating or replacing a system by running job steps created and tailored by the SMP/E GENERATE command. If you are installing functions packaged as an MVS Custom-Built Installation Process Offering (CBIPO), you use this method.

GENERATE eliminates the need to reinstall products that are not included by a generation procedure. GENERATE is also used to install products obtained through a CBIPO, or to install a mix of products, of which some are included in a generation procedure and some are not.

Notes:

1. This method is not applicable to an MVS/ESA SP Version 4 environment.
2. If any elements (such as macros or modules) are owned both by products that are included and by products that are not included in a generation procedure, you must make sure that the proper version of the element is used. To do this, you should examine all the output of steps 4 and 10 below and edit the generation procedure output or the GENERATE output when necessary. You may also need to process the steps in a different order from the order described below.

For Building a System or Subsystem Using CBIPO

See the CBIPO RIMs and *MVS Custom-Built Offerings Planning and Installation* for the latest methods.

For Building Your Own System or Subsystem

In this method of installation, you:

1. Make the service level of the distribution libraries you will use match the old service-level target zone from which you will do the GENERATE to reinstall any products not included by the generation procedure. (For example, before you do a system generation, accept or restore any service or products that have been applied to avoid regression.)
2. Use the RECEIVE command to get the new SYSMODs.
3. Use the ACCEPT command with the BYPASS(APPLYCHECK) operand to install the SYSMODs directly into the distribution libraries. This tells SMP/E to accept a SYSMOD even though it has not been applied.

   Note: You can simplify installation of products not included in the generation procedure that contain inline JCLIN if you process their JCLIN data at ACCEPT time. To do this, set the ACCJCLIN indicator in the DLIBZONE entry before the products are initially installed. Then accept these products. See Chapter 17 for more information.

4. Use the Stage 1 generation procedure provided to create the JCL for a job (or set of jobs). This procedure should be described in one of the documents associated with either the operating system (for an MVS SYSGEN) or the product.
5. Allocate new target libraries.
6. Allocate the following new SMP/E data sets:

   SMPMTS
   SMPSTS
   SMPLOG/SMPLOGA
   SMPSCDS

   If you do not start your system with new, empty versions of these data sets, you may inadvertently regress your new system when service is applied.
7. Build a new target zone that reflects the structure of the system that you are building. This involves:

   - Allocating a new target zone
   - Copying the distribution zone into the new target zone

   For more information about these processes, see "When Defining a New Target Zone" on page 115.

   Note: If you saved inline JCLIN at ACCEPT time for any of the products not included by the generation procedure, the target zone now describes the structure of those products.
8. Add any necessary entries to the new target zone and to the global zone. (For example, define DDDEF entries in the new target zone and OPTIONS entries in the global zone.)
9. Run JCLIN against the new target zone, using the Stage 1 generation output JCL from step 4 as input. This updates the target zone with information on the structure of all the products included by the generation procedure.

   Note: Process step 9 after step 11 if there are intersections between SYSGEN-supported products and products without SYSGEN support.
10. Run GENERATE against the old target zone, using the FORFMID operand to specify all products not included by a generation procedure. This creates jobs to reinstall these products.

    Note: If inline JCLIN was saved for any of these products at ACCEPT time, you should not include them on this GENERATE command.
11. Run JCLIN against the new target zone, using the GENERATE output JCL from step 10 as input. This updates the target zone with information on the structure of the specified products not included by the generation procedure.
12. Process GENERATE without the FORFMID operand to create the tailored JCL for the jobs that will install all products as defined in the target zone.
13. Run the jobs created by GENERATE to build the system.

Usage Notes

- DDDEF entries for target and distribution libraries must exist in the target zone. You must also define temporary data sets. You can use either DDDEF entries or module GIMMPDFT.
- If you intend to build your own system or subsystem that has products not included by a generation procedure, you may want to save inline JCLIN for these products in the distribution zone during ACCEPT processing. Saving the inline JCLIN eliminates the need to run the GENERATE and JCLIN steps to update the target zone for these products. See Chapter 17 for more information on ACCJCLIN and on saving and using inline JCLIN.
- Because you are accepting a SYSMOD without having applied it first, SMP/E will not delete the global zone SYSMOD entry or the PTS MCS entry when the SYSMOD is accepted. If you want the SYSMODs to be purged, you should use the REJECT command to purge all the accepted SYSMODs. See the chapter on the REJECT command in *SMP/E Reference*.
- Use CLEANUP to delete entries from the MTS, STS, and SCDS data sets. Use REJECT to delete MCSs from the PTS data set and to delete SYSMOD entries from the global zone. See *SMP/E Reference* for more details on the CLEANUP and REJECT commands.

Note: A checklist to help guide you through the above procedure is included in Appendix A.

RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-Stage 2 Generation Method

Use this method if you are creating or replacing a system by running job steps created and tailored by generation macros and procedures.

Notes:

1. This method is not applicable to an MVS/ESA SP Version 4 environment.
2. Any needed product functions not included by these macros and procedures must be separately installed or reinstalled.

In this method of installation, you:

1. Use the RECEIVE command to read in the new SYSMODs.
2. Use the ACCEPT command with the BYPASS(APPLYCHECK) operand to install the SYSMODs directly into the distribution libraries. This tells SMP/E to accept a SYSMOD even though it has not been applied.

   Note: Accepting the SYSMOD puts the generation macros supplied with the product into a set of distribution libraries. This allows the assembler to access the macros during the generation procedure.
3. Use the Stage 1 generation procedure provided to create the JCL for a job (or set of jobs). This procedure should be described in one of the documents associated with either the operating system (for an MVS SYSGEN) or the product.
4. Allocate new target libraries. If you do not start your new system with an empty MTS and STS, you may inadvertently regress your new system when service is applied.

5. Build a new target zone that reflects the structure of the system you are building. This involves:
   - Allocating a new target zone
   - Copying the distribution zone into the new target zone

   For more information about these processes, see "When Defining a New Target Zone" on page 115.
6. Add any necessary entries to the new target zone and to the global zone. (For example, define DDDEF entries in the new target zone and OPTIONS entries in the global zone.)
7. Run JCLIN against the new target zone, using the Stage 1 generation output JCL from step 3 as input. This updates the target zone with information on the structure of all the products included by the generation procedures.
8. Run the Stage 1 generation output JCL to install the products in the target libraries. This is the Stage 2 generation step.
9. Reinstall any required products that were not included in the generation procedure. For more information, see "Installation of a Product after System or Subsystem Generation" on page 119.

Usage Notes

- Because you are accepting a SYSMOD without having applied it first, SMP/E does not delete the global zone SYSMOD entry or the PTS MCS entry when the SYSMOD is accepted. If you want the SYSMODs to be purged, you should use the REJECT command to purge all the accepted SYSMODs. See the chapter on the REJECT command in *SMP/E Reference*.
- Use CLEANUP to delete entries from the MTS, STS, and SCDS data sets. Use REJECT to delete MCSs from the PTS data set and to delete SYSMOD entries from the global zone. See *SMP/E Reference* for more details on the CLEANUP and REJECT commands.

RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-APPLY Method

Use this method to update an existing system or subsystem with a function that does not supply inline JCLIN and uses a special generation macro or procedure to define the product structure.

Notes:

1. Do **not** use this for a function that supplies inline JCLIN.
2. This method is not applicable to an MVS/ESA SP Version 4 environment.

In this method of installation, you:

1. Receive the function SYSMOD from the distribution tape.
2. Accept the SYSMOD, using the BYPASS(APPLYCHECK) operand. This tells SMP/E to accept a SYSMOD even though it has not been applied. In this case, when accepting a SYSMOD, SMP/E does not look at the target zone at all.

   Note: Accepting the SYSMOD puts the generation macros supplied with the product into a set of distribution libraries. This allows the assembler to access the macros during the generation assembly.

3. Do a Stage 1 generation.
4. Run JCLIN, using the Stage 1 generation output as input for JCLIN. This updates the target zone with information about the target libraries.
5. Apply the SYSMOD. This installs the function in the target libraries.

Usage Notes

- Make sure the macro libraries for the new products are in the SYSLIB concatenation when doing the Stage 1 generation.
- The generation procedure provided with some products lets you generate job steps that allocate the required target libraries besides generating the link-edit and copy steps necessary to create the members of those libraries. Other pregeneration or postgeneration job steps required by that specific product may also be provided.

  When the SMP/E JCLIN command is run after a system generation, SMP/E does not do anything to support these other steps. SMP/E only looks at the copy, assembly, and link-edit steps to create the correct structure entries in the target zone. Therefore, when using this method, you may have to edit the assembler output to create a subset of the original job stream that is really run. GENERATE, which works on products not included by a generation procedure, does not need this step.
- Use CLEANUP to delete entries from the MTS, STS, and SCDS data sets. Use REJECT to delete MCSs from the PTS data set and to delete SYSMOD entries from the global zone. See *SMP/E Reference* for more details on the CLEANUP and REJECT commands.

RECEIVE-ACCEPT-Stage 1 Generation-DELETE-JCLIN-APPLY Method

Use this method to update an existing system or subsystem with a function that uses a special generation macro or procedure and that deletes another function. In this case, two functions are required: the replacement function and a dummy function. The dummy function SYSMOD must contain a ++VER statement with a DELETE operand specifying the previous releases. The dummy function SYSMOD and the RECEIVE, APPLY, and ACCEPT job stream used to install it would be packaged as an element in the replacement function being installed. This dummy function SYSMOD is received, applied, and accepted to delete the previous releases of the product from the existing target and distribution libraries.

In this method of installation, you:

1. Receive the function SYSMOD from the distribution tape.
2. Accept the SYSMOD, using the BYPASS(APPLYCHECK) operand. This tells SMP/E to accept a SYSMOD even though it has not been applied. In this case, when accepting a SYSMOD, SMP/E does not look at the target zone at all.

   Note: Accepting the SYSMOD puts the system generation macros supplied with the product into a set of distribution libraries. This allows the assembler to access the macros during the system generation.
3. Do a Stage 1 generation.
4. Install the dummy SYSMOD, using RECEIVE, APPLY, and ACCEPT. This deletes the previous releases of the product from the system.

5. Run JCLIN, using the Stage 1 generation output as input for JCLIN. This updates the target zone with information about the target libraries.
6. Apply the SYSMOD. This installs the function in the target libraries.

Usage Notes

- Make sure the macro libraries for the new products are in the SYSLIB concatenation when doing the Stage 1 generation.
- The generation procedure provided with some products lets you generate job steps that allocate the required target libraries besides generating the link-edit and copy steps necessary to create the members of those libraries. Other pregeneration or postgeneration job steps required by that specific product may also be provided.

  When the JCLIN is processed after a system generation, SMP/E does not do anything to support these other steps. SMP/E only looks at the copy, assembly, and link-edit steps to create the correct structure entries in the target zone. Thus, when using this method, you may have to edit the assembler output to create a subset of the original job stream that will really be run. GENERATE, which works on products not included by a generation procedure, does not need this step.
- Use CLEANUP to delete entries from the MTS, STS, and SCDS data sets. Use REJECT to delete MCSs from the PTS data set and to delete SYSMOD entries for the previous release and the dummy function from the global zone. Use UCLIN to delete the SYSMOD entries for the previous release and the dummy function from the target and distribution zones. See *SMP/E Reference* for more details on the CLEANUP, REJECT, and UCLIN commands.
- When you apply a function that deletes a previous release, SMP/E deletes the related entries from the target zone. Therefore, you do not want to update those entries using the JCLIN command and then have the APPLY command delete them. This is why you save the JCL and process it at APPLY time. This way, SMP/E deletes the old entries and then builds new ones using the JCL information.

Example of Using the Standard RECEIVE-APPLY-ACCEPT Method

This section describes the steps necessary for a standard function installation using the RECEIVE, APPLY, and ACCEPT commands.

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.

Prepare 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 use Information/ACCESS or SoftwareXcel Extended, or ask your IBM Support Center to check the IBM Preventive Service Planning (PSP) files for the latest information about the product. This is important because there may be a PTF for the product that is not included on a CUM tape or PUT, or one of the PTFs may contain an error you should know about.

2. Update the FMID list.

   When you order a product, you should update the FMID list in the global zone with the FMIDs of those products you will be receiving. (Check the program directory for this information.) This lets you receive any preventive service shipped on the normal PUTs between the time you order the product and the time you install it.

3. Read the program directory.

   The program directory tells you which libraries are affected, if any existing libraries must be expanded and by how much, and if any new libraries are required.

4. Prepare the target and distribution libraries.

   If the target and distribution libraries are properly prepared before you apply or accept a SYSMOD, very 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 may be applied or accepted. You may want to check full data sets against the SYSMODs you will be processing if you are unsure how large a data set will grow.

   Partitioned data sets with a high percentage of their space used can be compressed using IEBCOPY. If it looks like more space may be needed even 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 was necessary to allocate it on a different pack, update any procedure necessary with the new VOLUME data.

   This preparation is somewhat time consuming but takes less time and work than recovering from an out-of-space abend (E37, B37, and so on).

   The RETRY option or the COMPRESS(ALL) on your APPLY or ACCEPT command compresses the data set for any x37 ABEND and tries again, but if you still need space, SMP/E stops processing.

5. Allocate any new libraries required.

   Determine where they are to be allocated and then allocate them. Remember that, normally, the program directory shows how much space will be used—not how much space to allocate for the libraries. Libraries should be allocated with more space than required to allow for subsequent modifications. Normally, 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 be installing this function.

6. Check the SMP/E data sets to make sure 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 STS, which fills very 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). See Figure 8 on page 26 and Figure 9 on page 26 for examples.

7. **Create a backup for the volumes affected.**

   This is a very 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 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 enough time is available to allow these jobs to run to completion.

   The program directory or installation guide may contain information to help you in estimating the time required.

Stage the SYSMODs: The RECEIVE Process

When you get a new function, you usually get the function itself, plus any related service that is available.

- If the function was part of a CBPDO, you get one logical tape that contains the function and the unintegrated service.
- If the function was individually ordered from the IBM software distribution center, you get one tape containing the function itself and another tape (called the *cumulative service tape*, or *CUM tape*) that contains PTF service through the latest PUT.

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 that they can later be installed.

- 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. This is true for functions obtained in a CBPDO, as well as for functions obtained from the IBM distribution centers.
- If the function was part of a CBPDO, you can use the RIMLIB job included on the CBPDO tape to receive the function, service, and HOLDDATA shipped on the CBPDO. For more information, see *MVS Custom-Built Offerings Planning and Installation* and the *MVS CBPDO Memo to Users Extension* that was included with your tape.
- If the function was individually ordered from the IBM distribution center, you can use the jobs later in this chapter to receive the function from the product tape and the service from the CUM tape.

Receive the Function SYSMOD

Function SYSMODs obtained from IBM CBPDO tapes are packaged in RELFILE format. During the RECEIVE process, the unloaded partitioned data sets from the function tape are placed into the SMPTLIBs, which are used as temporary storage. SMPTLIBs that are uncataloged are automatically cataloged by SMP/E.

Note: Do not deallocate the SMPTLIBs after the RECEIVE step; they must remain allocated until after the function is applied and accepted.

Use the SMP/E dialogs or the sample job shown in Figure 12 to receive the function.

```
//RECEIVE  JOB 'accounting info',MSGLEVEL=(1,1)
//RECFUNCT EXEC SMPPROC
//SMPTLIB  DD UNIT=3380,VOL=SER=TLIB01
//SMPPTFIN DD DSN=SMPMCS,
//            UNIT=(TAPE,,DEFER),
//            VOL=SER=xxxxxx,
//            LABEL=(,SL),
//            DISP=(OLD,PASS)
//SMPCNTL  DD  *
SET      BDY(GLOBAL).       /* Set to global zone.      */
RECEIVE  S(HXX1200)         /* Receive only this        */
         SYSMODS            /*  SYSMOD.                 */
         LIST               /* List its MCS             */
         .                  /*  statements.             */
/*
```

Figure 12. Sample Job for Receiving a Function Tape

Receive the Cumulative Service Tape

If you got a cumulative service tape with your function order, the next step is to receive the PTFs on that tape.

Table 5 defines the contents of each file on the CUM tape.

Table 5. Format of a CUM Tape

File Number	Used with SMP/E	Description of Contents
1	Yes	Preventive service PTFs
2	No	Reserved
3	No	Reserved
4	Yes	HOLDDATA for exception SYSMODs
5	No	JCLIN file
6	Optional	UCLIN file

Use the SMP/E dialogs, or the sample job shown in Figure 13 on page 61, to receive service and HOLDDATA from the CUM tape.

```
//RECEIVE  JOB 'accounting info',MSGLEVEL=(1,1)
//RECCUM   EXEC SMPPROC
//SMPPTFIN DD UNIT=(TAPE,,DEFER),
//            VOL=(PRIVATE,RETAIN,SER=xxxxxx),
//            LABEL=(1,NL),
//            DCB=(LRECL=80,BLKSIZE=7200,RECFM=FB),
//            DISP=(SHR,PASS)
//SMPHOLD  DD UNIT=AFF=SMPPTFIN,
//            VOL=(PRIVATE,RETAIN,SER=xxxxxx),
//            LABEL=(4,NL),
//            DCB=(LRECL=80,BLKSIZE=7200,RECFM=FB),
//            DISP=(SHR,PASS)
//SMPCNTL  DD *
SET      BDY(GLOBAL).       /* Set to global zone.     */
RECEIVE  SYSMODS            /* Receive all SYSMODs     */
         HOLDDATA           /*  and any exception data. */
         SOURCEID(xxxxxxxx) /* Assign them a common    */
         .                  /*  identifier.            */
LIST     SYSMODS            /* Now list the SYSMOD     */
         MCS                /*  and MCS (cover letter) */
         SOURCEID(xxxxxxxx) /*  for only those SYSMODs. */
         .
/*
```

Figure 13. Sample Job for Receiving a Cumulative Service Tape

The SOURCEID operand of the RECEIVE and LIST commands is used to list the PTF cover letters, rather than the LIST operand of RECEIVE, because the output is in a more usable format. You should substitute a meaningful value for *xxxxxxxx* in each command. The value should be unique to this product.

The DCB values for SMPHOLD and SMPPTFIN may have to be modified.

If any of the PTFs on the tape indicate (by the ++HOLD SYSTEM statement with a reason ID equal to "UCLIN") that the PTF requires UCLIN to be run, file 6 contains the UCLIN for that PTF. You must follow the instructions given in the PTF for the installation of the UCLIN. See "PTFs That Need UCLIN" on page 82 for further information.

Update 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. Some functions cannot be installed by the standard RECEIVE-APPLY-ACCEPT method; they require a generation procedure (such as MVS SYSGEN, IMSGEN, or CICS subsystem generation) for installation. Review the program directory for the products you are installing to make sure the function can be installed by the RECEIVE-APPLY-ACCEPT process.

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 may be able to apply all the required SYSMODs with one APPLY command. There are several advantages to this method:

- It eliminates the iterations through the APPLY process looking for the complete set of SYSMODs required.
- You replace elements in the target libraries less often, reducing the risk of running out of space.
- Overall processing time is decreased because of less SMP/E overhead and fewer invocations of the system utilities.

Therefore, although SMP/E provides support for 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 direct otherwise. This is the method illustrated in subsequent examples. See the chapter on the APPLY command in *SMP/E Reference* for more information about the APPLY command operands.

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 contact the IBM Support Center to get additional HOLDDATA from the PUT PSP file or the CORPE PSP file. Obtaining and receiving additional HOLDDATA is covered under "Processing HOLDDATA from PSP Files" on page 110. The other two steps are covered in more detail in the following sections.

- 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.

Check the Update: The APPLY CHECK Process

The purpose of this step is to determine:

- If any errors will occur while the new function is being applied (except for those errors that occur as a direct result of an update, such as a target library running out of space). This includes missing DDDE entries.
- If 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 sample job shown in Figure 14 to do an APPLY CHECK for the function and related SYSMODs.

```
//APPLY     JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPTLIB  DD UNIT=3380,VOL=SER=TLIB01
//SMPCNTL  DD *
SET      BDY(TGT1).        /* 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(ID         /* Bypass all options        */
                IFREQ
                PRE
                REQ
                HOLDSYSTEM
                HOLDERROR
                HOLDUSER
                HOLDCLASS(UCLREL,ERREL)
               )
         .
/*
```

Figure 14. Sample Job for Applying a Function and Cumulative Service (Check Mode)

Notes:

1. The SMPTLIB DD statement included here is the same one used when the function was being received. (This DD statement is not necessary if the SMPTLIB data sets are cataloged.)
2. This example uses the GROUPEXTEND operand because the function and all its service are being installed at the same time.
3. All the BYPASS options are specified to allow SMP/E to check for all possible errors.

Research 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 may have been detected:

- Some DD statements may be missing. Check the program directory or *SMP/E Reference* to determine why they are required and how the DD statement should be specified.
- Some APAR fixes or USERMODs may 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 be 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 may be missing. If so, you should determine how these PTFs can be obtained. Some may already be on PUTs you 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, unlike the previous condition, 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 may 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 is a result of 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, then further research is necessary. Contact the current owner of the element to determine that product's relationship with the one you are installing.

- Some of the PTFs may 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 if the condition should be bypassed (using the BYPASS(HOLDERR) operand), or if the PTF should not be installed, or if a fix for the APAR should be obtained.

Get Additional SYSMODs: After doing the research step, you may decide that additional SYSMODs are needed. If so:

1. Obtain the additional SYSMODs from CBPDO or the applicable PUTs, or from CSSF Information/Access, SoftwareXcel Extended, or the IBM Support Center.
2. Receive the additional SYSMODs, using the same source ID value as used when processing the CUM tape.
3. Rerun the APPLY CHECK job.

This process should continue until no new errors are reported.

Update the Target Library: The APPLY Process

Once the APPLY CHECK and associated research is done, and if enough preparation was done, 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 product.

Use the SMP/E dialogs or the sample job shown in Figure 15 on page 65 to apply the function and any related SYSMODs.

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLY    EXEC SMPPROC
//SMPTLIB  DD  UNIT=3380,VOL=SER=TLIB01
//SMPCNTL  DD  *
SET      BDY(TGT1).          /* 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,...) /*                */
               )             /*  reason IDs.             */
         .                   /*                          */
/*
```

Figure 15. Sample Job for Applying a Function and Cumulative Service (Update Mode)

If you have obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand or specify the APARS and USERMODS operand if **all** applicable APARs and USERMODs are to be applied.

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.

Test the New Function

After installing the new function, you should perform two operations:

1. Create a backup of the updated data sets, including those 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.

   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 the new function on a noncritical test system should you put that function into production. The test phase should not be considered completed until after you have run the new function in production mode for some period (as determined by your installation requirements). At the end of that period, if no errors are found, you are ready to go to the next step, updating your distribution libraries.

Update the Distribution Libraries: The ACCEPT Process

The last major phase of installing a new function is updating the distribution libraries using the SMP/E ACCEPT command. This is a very critical step because 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 "Update the Target Libraries: The APPLY Process" on page 61 and the same three-phase operation:

1. Receive additional data.

   Before starting the ACCEPT process you should obtain any additional HOLDDATA for the function you are installing from CBPDO or the applicable PUTs, or from CSSF Information/Access, SoftwareXcel Extended, or the IBM Support Center. See "Processing HOLDDATA from PSP Files" on page 110 for additional information.

   Note: 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.

2. Run the ACCEPT CHECK job.

   This is a nonupdating mode of ACCEPT. Its purpose is to help resolve any problems that may prevent the ACCEPT from completing processing successfully.

3. Accept the SYSMOD update.

   This installs the new function and service into the distribution library.

Check the Update: The ACCEPT CHECK Process

The ACCEPT CHECK job provides the same function for the distribution libraries that the APPLY CHECK job provided for the target libraries. See "Check the Update: The APPLY CHECK Process" on page 62.

Use the SMP/E dialogs or the sample job shown in Figure 16 on page 67 to do an ACCEPT CHECK for the function and related SYSMODs.

Research the ACCEPT CHECK Reports: The ACCEPT CHECK reports should be researched in the same manner as the APPLY CHECK reports (see "Research the APPLY CHECK Reports" on page 63).

Get Additional SYSMODs: The process of getting additional SYSMODs or APAR fixes for those PTFs being accepted is the same as during the APPLY process (see "Get Additional SYSMODs" on page 64).

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPTCK EXEC SMPPROC
//SMPTLIB  DD UNIT=3380,VOL=SER=TLIB01
//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(ID          /* Bypass all options       */
                IFREQ
                PRE
                REQ
                HOLDSYSTEM
                HOLDERROR
                HOLDUSER
                HOLDCLASS(UCLREL,ERREL)
               )
         .
/*
```

Figure 16. Sample Job for Accepting a Function and Cumulative Service (Check Mode)

Update the Distribution Library: The ACCEPT Process

The ACCEPT process updates the distribution libraries. Use the SMP/E dialogs or the sample job shown in Figure 17 to accept the function and any related SYSMODs.

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPT   EXEC SMPPROC
//SMPTLIB  DD  UNIT=3380,VOL=SER=TLIB01
//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,...) /*               */
               )            /*  reason IDs.             */
         .                  /*                          */
/*
```

Figure 17. Sample Job for Accepting a Function and Cumulative Service (Update Mode)

Note: If you have obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand or specify the APARS and USERMODS operand if **all** applicable APARs and USERMODs are to be installed.

Chapter 5. Installing Preventive Service

This chapter describes the steps for installing preventive service. It discusses these topics:

- An introduction to preventive service
- A summary of the preventive service process
- Preparing your system
- Staging the SYSMODs with the RECEIVE 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:

- A tape that contains the preventive service.

  This may be a CBPDO tape or a program update tape (PUT). CBPDO tapes contain current PUT service, plus approved service that is not yet available on a PUT.

- A well-defined set of steps that should be followed to install each PUT in order to bring your system up to the current service level.

When installing service from a CBPDO tape, you should be aware of how CBPDO tapes handle certain things differently from PUTs.

- In a CBPDO, service and HOLDDATA are shipped by feature. There are features for MVS, NCP, database systems, and CICS. These features correspond to the system release values (SRELs) to which products are applicable. This is different from how service and HOLDDATA are shipped on PUTs.

  - In a CBPDO, you get service and HOLDDATA applicable to all products **within a given feature** for which you are licensed under a single customer number.

  - On a PUT, you get service and HOLDDATA applicable to **all products** for which you are licensed **under a single customer number**. The products are not divided into features.

- CBPDO tapes may contain service from several PUTs, as well as service not yet available on a PUT.

  If you receive service and HOLDDATA from both CBPDO tapes and PUTs, make sure you install them according to the PUT numbers (for example, PUT9001, followed by PUT9002, and so on). On a CBPDO tape, all PTFs that are currently available on a PUT have a source ID that corresponds to the PUT number. Once you have received PTFs from a CBPDO tape, do not try to receive them again from the PUT they are shipped on.

Subsequent sections of this chapter describe the SMP/E preventive service process. For users familiar with previous SMP releases, this process is similar to that defined on the PUT (referred to as the *PUT process*). With SMP/E, some steps in that process are no longer necessary or have been modified to use the functions in SMP/E, and some new steps have been added.

CBPDO Tapes

CBPDO tapes may be ordered periodically, as you decide to update your system. A CBPDO tape contains the PTFs, HOLDDATA, and PSP upgrade files to bring your system up to a service level that you select. A CBPDO is ordered for a particular feature (MVS, NCP, database systems, or CICS). These features correspond to the SRELs to which products are applicable.

You have two options when ordering a CBPDO: you can get products plus service, or service only. (If you are interested just in installing preventive service, you order a service-only CBPDO.) With both these options you automatically receive service for products for which you are already licensed under a single customer number for a single feature.

The amount of service you receive in your CBPDO depends on your selection of a service level and whether this is your first CBPDO order.

- If you select a PUT service level, you get 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 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 service following the PUT service level that was shipped in your previous CBPDO.

Notes:

1. The service level of your system must not be more than two years behind the currently available PUT. Service in a CBPDO does not go back more than two years.
2. A CBPDO order is based on the total set of products for which you are licensed under a single customer number. It does not reflect the contents of any specific system within the establishment defined by that customer number. For example, the same establishment may have MVS/XA on one system and MVS/370 on another system, and be licensed for both under the same customer number. In this case, service for both MVS/XA and MVS/370 is included in the CBPDO order.

The CBPDO tape is a standard label (SL) tape, with files arranged as shown in Table 6.

Table 6. Format of a CBPDO Tape

File Number	Processed by SMP/E	Description of Contents
1	No	Documents
2	No	Installation RIMs
3	Yes	HOLDDATA for exception SYSMODs
4	No	Program directories and PSP information
5	Yes	SMPMCS file (MCS statements for SYSMODs on the tape), PTFs, and cover letters
6-n	Yes	RELFILEs for products on the tape

PUTs

As an IBM customer you may receive, periodically, a customized PUT based on your IBM software distribution profile. This tape has the PTFs, HOLDDATA, and UCLIN data needed to bring your system up to the level defined for that PUT.

Service levels identify when the PUTs are shipped. For example:

Tape	Service Level
9103	Third PUT for 1991
9104	Fourth PUT for 1991
9201	First PUT for 1992

The program update tape is a nonlabeled (NL) tape, with files arranged as shown in Table 7 on page 72.

The optional files in Table 7 can be used with SMP/E; the programs in file 9 can be used to print the information in file 10. They are not necessary to the process, however.

With SMP4, you were expected to use PUT-provided programs to run the job streams supplied on the PUT. These provided the documentation, the PTFs, and control of the process. The PUT, except for the addition of the exception SYSMOD management statements in file 4, has not been changed. For SMP4 users, the existing PUT process can still be used; file 4 will not be referred to in this process. With the addition of the exception SYSMOD statements, and certain RECEIVE, APPLY, and ACCEPT command parameters, however, SMP/E allows you much flexibility in installing preventive service.

Table 7. Format of a PUT

File Number	Processed by SMP/E	Description of Contents
1	Yes	Preventive service PTFs
2	No	Listing of new PTFs shipped
3	No	Reserved
4	Yes	HOLDDATA for exception SYSMODs
5	No	JCLIN file
6	Optional	UCLIN file
7	No	PUTLOADR program
8	No	PUT job streams
9	No	DOCPRINT and PUTEDIT programs
10	No	PUT documents
11	No	PUT XREF file

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 defined in Chapter 6, although the steps within each phase differ. These phases are the basic SMP/E operations to install any SYSMOD.

Each of these phases consists of a series of steps, either SMP/E jobs, research, or system utilities invocations, 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. At times, however, a PTF may require special processing. In these cases, the PTF itself contains instructions telling you how to install it; you should then follow those instructions.

Generally, you should attempt first 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 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.

Prepare Your System

Before you start installing preventive service, you should do the following to make sure your system is ready and that you can recover in case of a serious failure during installation:

- Get the latest HOLDDATA from the IBM Support Center, the Customer Software Support Facility (CSSF) of Information/Access, or SoftwareXcel Extended.

  As described in Chapter 8, there may be PTFs that were discovered to be in error (called PE-PTFs) after your program update tape was shipped to an IBM software distribution center. You can get this information from your IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended, and build the corresponding ++HOLD/++RELEASE statements necessary to prevent introducing problems by processing PTFs with known errors.

  You receive this HOLDDATA during the next phase.

- Make sure you have the publications you need.
- Estimate the time you need for APPLY and ACCEPT processing. Make sure enough time is available to let these jobs run to completion.
- Back up your disk packs.

Note: For more information about how to prepare for installing preventive service obtained from a CBPDO, see *MVS Customer-Built Offerings Planning and Installation.*

Stage the SYSMODs: The RECEIVE Process

After you prepare your target and distribution libraries, you receive the preventive service SYSMODs (PTFs) and the HOLDDATA (including data on the CBPDO or PUT and any obtained from the IBM Support Center) into the SMP/E database (the global zone and the SMPPTS).

- If the service was obtained from a CBPDO, you can use SMP/E dialogs or the RIMLIB job included on the CBPDO tape to receive the service and HOLDDATA shipped on the CBPDO. For more information, see *MVS Custom-Built Offerings Planning and Installation* and *MVS CBPDO Memo to Users Extension* included with your tape.
- If the service was obtained from a PUT, you can use the SMP/E dialogs or the sample job shown in Figure 18 on page 74 to receive the service and HOLDDATA.

```
//RECEIVE  JOB 'accounting info',MSGLEVEL=(1,1)
//RECPUT   EXEC SMPPROC
//SMPPTFIN DD DSN=SMPPTFIN,
//            UNIT=TAPE,
//            VOL=(PRIVATE,RETAIN,SER=xxxxxx),
//            LABEL=(,NL),
//            DCB=(LRECL=80,BLKSIZE=7200,RECFM=FB),
//            DISP=(SHR,PASS)
//SMPHOLD  DD DSN=SMPHOLD,
//            UNIT=TAPE,
//            VOL=(PRIVATE,RETAIN,SER=xxxxxx),
//            LABEL=(4,NL),
//            DCB=(LRECL=80,BLKSIZE=7200,RECFM=FB),
//            DISP=(OLD,PASS)
//SMPCNTL  DD *
SET      BDY(GLOBAL).   /* Set to global zone.       */
RECEIVE  SYSMODS        /* Receive all SYSMODs       */
         HOLDDATA       /*  and any exception data   */
         SOURCEID(      /*  and assign them a common */
           xxxxxxxx)    /*  identifier.              */
         .              /*                           */
LIST     SYSMODS        /* Now list the SYSMOD       */
         MCS            /*  and MCS (cover letter)   */
         SOURCEID(      /*  for only those SYSMODs.  */
           xxxxxxxx)    /*                           */
         .
/*
```

Figure 18. Sample Job for Receiving a PUT (SYSMODs and HOLDDATA)

The SOURCEID operand of the RECEIVE command is used with the LIST command to list the PTF cover letters, rather than the LIST operand of RECEIVE, because the output is in a more usable format. You should substitute a meaningful value for *xxxxxxxx* in each command. The value should be unique and easily tied to a PUT. The recommended format is *PUTyynn*, where *yynn* is the year and the sequence number.

Note: If you are receiving SYSMODs and HOLDDATA from a CBPDO, you can assign the SYSMODs a SOURCEID to identify the source of the PTFs. The recommended format is *PDOyyww*, where *yy* is the year and *ww* the week of the CBPDO tape.

The DCB values for SMPHOLD and SMPPTFIN are those used in the PUTs when this publication was written. If these values change, use the ones defined on the PUT label.

For both CBPDO and PUTs, you should call the IBM Support Center to obtain additional HOLDDATA (unless you just received your CBPDO tape). For additional information see "Processing HOLDDATA from PSP Files" on page 110. The SMP/E dialogs or the job shown in Figure 19 on page 75 can be used to process the data set with the HOLDDATA obtained from the IBM Support Center.

```
//RECEIVE   JOB  'accounting info',MSGLEVEL=(1,1)
//RECESM    EXEC  SMPPROC
//SMPHOLD   DD ...data describing your data set
//*            ...must be RECFM=FB, LRECL=80
//SMPCNTL   DD *
SET      BDY(GLOBAL).    /* Set to global zone.      */
RECEIVE  HOLDDATA.       /* Receive HOLDDATA.        */
/*
```

Figure 19. Sample Job for Receiving HOLDDATA from the IBM Support Center

The HOLDDATA you obtain from the IBM Support Center should be in SMP/E format. If not, or if you are creating your own HOLDDATA, you should see *SMP/E Reference* to review the syntax for the ++HOLD statements.

Update the Target Libraries: The APPLY Process

After you prepare your target and distribution libraries and receive all the necessary PTFs and HOLDDATA, you update the target libraries. Though most PTFs can be installed directly into the target libraries, some PTFs require special processing in order to be installed. Some examples of special processing are:

- You must use a system generation procedure, either to install the PTF or to activate the service after installation.
- You may have to run UCLIN in order to install the PTF.
- The PTF may cause changes that affect your operations.

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; but rather, you should attempt to install all PTFs not requiring such actions and then return to process these. See "Install PTFs That Need Special Processing" on page 81 for additional information regarding these PTFs.

When installing preventive service, you are concerned with two groups of PTFs:

- All PTFs from the CBPDO or PUT 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 iterations through the APPLY process looking for the complete set of PTFs required.
- It reduces exposure to out-of-space conditions, since you are replacing elements in the target libraries less often.
- It decreases overall processing time because of less SMP/E overhead and fewer invocations of the system utilities.

When you are updating 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 PUT 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 has been a delay between the RECEIVE and APPLY staging phase and the target update phase.

   This first step is not discussed further; see "Stage the SYSMODs: The RECEIVE Process" on page 73 if you need to perform this step.

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, when the preventive service is installed into the target libraries.

The following sections describe the last two steps as well as processing of the PTFs that require special processing.

Check the Update (APPLY CHECK)

The purpose of this step is to decide:

- If any errors will occur when you are applying 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)
- If 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.

Use the SMP/E dialogs or the sample job shown in Figure 20 on page 77 to do an APPLY CHECK for preventive service. Figure 20 is an example of an APPLY CHECK job for PTFs on PUT 8905.

The GROUP and GROUPEXTEND operands allow SMP/E to include any PTF that may be required to install PTFs on the current PUT (PUT8905). Some of the PTFs on previous tapes may not have been installed because they were in hold status (PE-PTFs) at the time the PUT was installed. The current PUT may contain fixes for the APARs causing the original PTFs to be held. These PTFs, because they have module intersections with the PE-PTF, must either be prerequisite to or supersede the old PTFs, thus allowing SMP/E to include the old PTFs automatically when the fixing PTF is installed.

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1).         /* Set to target zone.      */
APPLY    SOURCEID(PUT8905)  /* Apply this PUT           */
         GROUPEXTEND        /*  and all requisite PTFs, */
         CHECK              /*  but do not update libs. */
         SELECT(sysmod-id,...) /* Select additional     */
                            /*  service if required.    */
         REDO               /* Use REDO only if any of  */
                            /*  the SYSMODs in the      */
                            /*  SELECT list are to be   */
                            /*  reinstalled.            */
         BYPASS(ID          /* Bypass ID check,         */
                HOLDCLASS(UCLREL) /* UCL                */
                HOLDSYS(reason-id,...) /* hold sys check*/
               )            /* reason IDs.              */
         .                  /*                          */
/*
```

Figure 20. Sample Job for Applying a Single PUT (Check Mode)

You may be able to improve SMP/E performance by including the source IDs for previous PUTs within the SOURCEID operand as shown in Figure 21.

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLYCHK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1).         /* Set to target zone.      */
APPLY    SOURCEID(PUT8905   /* Apply this PUT           */
                  PUT8904   /*  and back-level tapes    */
                  PUT8903   /*  back to some reasonable */
                  PUT8902   /*  level.                  */
                  PUT8901)  /*                          */
         GROUPEXTEND        /* And all requisite PTFs.  */
         CHECK              /* But do not update libs.  */
         SELECT(sysmod-id,...) /* Select additional     */
                            /*  service if required.    */
         REDO               /* Use REDO only if any of  */
                            /*  the SYSMODs in the      */
                            /*  SELECT list are to be   */
                            /*  reinstalled.            */
         BYPASS(ID          /* Bypass ID check,         */
                HOLDCLASS(UCLREL) /* UCL                */
                HOLDSYS(reason-id,...) /* hold sys check*/
               )            /* reason IDs.              */
         .                  /*                          */
```

Figure 21. Sample Job for Applying Multiple PUTs (Check Mode)

Note: This form of the SOURCEID operand can also be used to group PUTs initially in one APPLY command.

If you wish to install preventive service on only selected functional areas of the system, you can also specify the FORFMID operand on the APPLY command, either specifying specific function identifiers (FMIDs) or the name of one or more FMIDSETs.

Research 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 perform further research. Here are some of the errors that may be detected:

- Some DD statement may be missing. Check *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 will 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 these PTFs can be obtained. Some may be on PUTs 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, unlike the previous condition, 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 on a subsequent PUT.

  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 if the condition should be bypassed, using the BYPASS(HOLDERR) operand, if the PTF should not be installed; or if a fix for the APAR should be obtained.

A common cause of regression is user modification. When user modifications 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 user modification. 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 user modification.
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 user modification into the distribution libraries.

When USERMODs are applied on a system, it is up to you to ensure 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 version of the module, macro, and source statement that currently exists in the distribution library.

   Use the SMP/E dialogs or the sample job shown in Figure 22 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.    */
        RETRY           /* Compress, if x37 occurs. */
        .               /*                          */
/*
```

Figure 22. Sample Job for Restoring APARs or USERMODs

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 below. 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.

Figure 23 is an example of the REJECT job:

```
//SMPREJ  JOB 'accounting info',MSGLEVEL=(1,1)
//REJECT  EXEC SMPPROC
//SMPCNTL DD *
SET      BDY(GLOBAL).     /* Set to global zone.   */
REJECT                    /* Remove from PTS and   */
                          /*  global zone          */
         S(AZ00001,       /* these two SYSMODS.    */
           AZ00002)       /*                       */
         .                /*                       */
/*
```

Figure 23. Sample Job for Rejecting APARs or USERMODs

- Receive the USERMODs or APARs.

  The APAR or USERMOD you have updated should now be received.

Get 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.

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 PUT or assign them a unique value and then add that value to the SOURCEID operand.

This process should continue until no new errors are reported.

Update 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 sample job shown in Figure 24 on page 81 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(PUT8905) /* Apply this PUT           */
         GROUPEXTEND       /*  and all requisite PTFs, */
         SELECT(UZ00001    /* Plus two other PTFs.     */
                UZ00002    /*                          */
                AZ12345    /* Plus two APAR fixes.     */
                AZ12346)   /*                          */
                           /*  No check this time.     */
         REDO              /* Use REDO only if any of  */
                           /* the SYSMODs in the SELECT*/
                           /* list are to be           */
                           /* reinstalled.             */
         BYPASS(           /* Bypass only required for */
                HOLDSYS(reason-id,...) /* hold sys check*/
               )           /* reason IDs.              */
         .                 /*                          */
/*
```

Figure 24. Sample Job for Applying Preventive Service (Update Mode)

- If you have obtained additional APAR fixes or USERMODs, you should either specify each of these SYSMODs in the SELECT operand or specify the APARS and USERMODS operands if **all** applicable APARs and USERMODs are to be installed.
- If any of the SYSMODs specified in the SELECT list have already been applied, and you wish to reinstall them, you must also specify the REDO operand on the APPLY command.
- If you wish to install preventive service on only 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.

Install PTFs That Need Special Processing

There are many reasons why a PTF may require special processing, therefore making it 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 shown in Figure 25.

```
++HOLD(sysmod-id)          /* Identify PTF number.     */
      SYSTEM               /* Special processing info. */
      FMID(sysmod-id)      /* Functional owner.        */
      REASON(reason-id)
      COMMENT(....
              ....any amount of comment text
              ....
             )
      .
```

Figure 25. Sample ++HOLD MCS

See *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. You should follow the directions given there.

PTFs That Need UCLIN

For some SYSMODs provided on a PUT, you may have to run UCLIN either before or after installing the SYSMOD (the PTF gives exact directions). The UCLIN to be run should be listed in the PTF cover letter. The UCLIN statements for all PTFs on the PUT are contained in file 6 of the PUT.

This file can be read into a data set or just printed so it is readable. The IEBGENER job shown in Figure 26 reads the UCLIN into a file called UCLFILE on the volume you name.

```
//READUCL  JOB 'accounting info',MSGLEVEL=(1,1)
//READUCL  EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD DSN=UCLIN,
//            UNIT=TAPE,
//            VOL=SER=TPUTxx,
//            LABEL=(6,NL),
//            DCB=(BLKSIZE=7200,LRECL=80,RECFM=FB),
//            DISP=SHR
//SYSUT2   DD DSN=UCLFILE,
//            UNIT=SYSDA,
//            DISP=(NEW,KEEP),
//            DCB=(BLKSIZE=7200,LRECL=80,RECFM=FB),
//            SPACE=(TRK,(10,5))
//SYSIN    DD DUMMY
```

Figure 26. Sample Job for Copying a PUT UCLIN File

As you can see, the UCLIN records are in 80-byte format, with the same DCB parameters as the PTFs and SMPHOLD file. The example uses the same DCB parameters on output and input. You can change the output (SYSUT2) to suit your jobs.

The UCLIN is formatted and tagged as shown in Figure 27. Columns 73 to 80 contain the characters *UCL* followed by the five-number PTF identifier.

```
----+----1----+----2----+...........+----7----+----8
UCLIN  CDS  .                               UCL12345
  DEL MOD(IFBMODXX) .                       UCL12345
ENDUCL .                                    UCL12345
UCLIN  CDS  .                               UCL54321
  DEL MOD(IFAMODYY)  .                      UCL54321
ENDUCL      .                               UCL54321
```

Figure 27. Sample UCLIN for SMP4

Note: The UCLIN is in SMP4 format. If it is necessary to use any of this UCLIN with a SYSMOD, you must do the following:

1. Add the appropriate SET command before the UCLIN command.
2. Use UCL statements as delivered.

3. Put the sets of SMP/E commands and UCL statements in a data set and run SMP/E, having the SMPCNTL DD statement pointing to that data set.

   If the data set operands are left on the UCLIN command, SMP/E ignores these operands if they are syntactically correct for SMP4.

In the previous example, PTFs UZ12345 and UZ54321 would require the UCL described. The UCLIN job for UZ12345, specified for target zone MVSTGT1, might look like that in Figure 28.

```
//PUTUCL    JOB 'accounting info',MSGLEVEL=(1,1)
//PUTUCL    EXEC SMPPROC
//SMPCNTL   DD DSN=UCLFILE,DISP=SHR
/*
```

Figure 28. Sample Job for Running UCLIN

The data set pointed to by the SMPCNTL DD statement would contain the statements shown in Figure 29.

```
----+----1----+----2----+...........+----6----+----7----+----8
SET      BDY(MVSTGT1).      /* Set to target zone.      */
UCLIN    .
DEL      MOD(IFBMODXX).
ENDUCL   .
UCLIN    .                                              UCL54321
  DEL MOD(IFAMODYY)  .                                  UCL54321
ENDUCL      .                                           UCL54321
```

Figure 29. Sample UCLIN for SMP/E

Test 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 those 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 completed until you have run the service in production mode for some period (as determined by your installation requirements). At the conclusion of that period, if no errors are found, you are ready to proceed to the next step: updating your distribution libraries.

Update 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 very critical step because 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 "Update the Target Libraries: The APPLY Process" on page 75, and the same three-phase operation:

1. Receive additional HOLDDATA.

   Before starting the ACCEPT you should contact the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended to obtain any additional HOLDDATA for the service level 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 has been a delay between the RECEIVE staging phase and the ACCEPT DLIB update phase.

   Notes:

   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. See Chapter 13, "Identifying Installed SYSMODs Affected by Error Holds: The REPORT ERRSYSMODS Command" on page 137 for more information.

2. 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.

3. Run the ACCEPT update.

   The third job is the updating mode of ACCEPT, when the preventive service is installed into the distribution libraries.

Note: There may be special processing required during the ACCEPT process. These PTFs should be handled in the same manner as during the APPLY process.

Check 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 "Check the Update (APPLY CHECK)" on page 76.

Use the SMP/E dialogs or the sample job shown in Figure 30 on page 85 to do an ACCEPT CHECK for preventive service. Figure 30 is an example of an ACCEPT CHECK job for PTFs on PUT 8905.

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPTCK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1).        /* Set to DLIB zone.        */
ACCEPT   SOURCEID(PUT8905)  /* Accept this PUT          */
         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.   */
         REDO               /* Use REDO only if any of */
                            /* the SYSMODs in the SELECT*/
                            /* list are to be          */
                            /* reinstalled.            */
         BYPASS(ID          /* Bypass ID check,        */
                HOLDCLASS(UCLREL) /*                   */
                HOLDSYS(reason-id,...) /* hold sys check*/
               )            /* reason IDs.             */
         .                  /*                         */
/*
```

Figure 30. Sample Job for Accepting a PUT (Check Mode)

Note: This example can also be used with CBPDOs, using a different source ID.

If you wish to install preventive service on only 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.

Research the ACCEPT CHECK Reports

The ACCEPT CHECK reports should be researched in the same manner as the APPLY CHECK reports (see “Research the APPLY CHECK Reports” on page 78).

Get 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 “Get Additional SYSMODs” on page 80).

If you obtain additional SYSMODs during the ACCEPT phase, you should process these through the APPLY phase before accepting them.

Update the Distribution Library (ACCEPT)

The ACCEPT command causes SMP/E to update the distribution libraries. Use the SMP/E dialogs or the sample job shown in Figure 31 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(PUT8905)  /* Accept this PUT          */
         GROUPEXTEND(       /*  Include requisite PTFs  */
                NOAPARS     /*  don't include APARs or  */
                NOUSERMODS) /*  USERMODs                */
                            /*  No check this time.     */
         SELECT(sysmod-id,...) /* Select additional     */
                            /*  service if required.    */
         REDO               /* Use REDO only if any of  */
                            /*  the SYSMODs in the      */
                            /* SELECT list are to be    */
                            /* reinstalled.             */
         BYPASS(            /* Bypass only required     */
                HOLDSYS(reason-id,...) /* hold sys check*/
               )            /* reason IDs.              */
         .                  /*                          */
/*
```

Figure 31. Sample Job for Accepting a PUT (Update Mode)

Notes:

1. If you have obtained additional APAR fixes or USERMODs you should either specify each of these SYSMODs in the SELECT operand or specify the APARS and USERMODS operand if **all** applicable APARs and USERMODs are to be installed.
2. If you wish to install preventive service on only 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.

Install PTFs That Need Special Processing

During the ACCEPT process, the same considerations are valid for special processing requirements as were valid during the APPLY process (see "Install PTFs That Need Special Processing" on page 81).

Chapter 6. Installing Corrective Service

This chapter 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
- 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. *SMP/E Diagnosis Guide* provides helpful information on diagnosing and handling SMP/E problems. This may 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 user modifications involved
- The load modules that are affected.

After determining the cause of the error, you want a fix for the problem. There are several possibilities:

1. The problem has been previously reported and a PTF has already been created to fix the module. That PTF may not have been shipped on a PUT.

   If already shipped on a PUT, you should have it in your shop (already received). If not, the IBM Support Center will help you get a copy of the PTF.

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 therefore, 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.

Build or Check the Fix

If the fix is a PTF (either from a PUT or from the IBM Support Center), assume that the construction of that fix is accurate and the material in this section is not applicable.

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 *SMP/E Program Packaging Guide* for rules for constructing USERMOD SYSMODs.

Figure 32 shows the general format of all ++APAR fixes.

```
++APAR(xxxxxxx).            /* APAR identifier          */
++VER (srel)                /* System identifier        */
      FMID(aaaaaaa)         /* Functional area          */
      PRE (                 /* PRE some SYSMODs.        */
           bbbbbbb          /* PRE RMID of element      */
           ccccccc ccccccc  /*  and any UMIDs present.  */
           ccccccc ccccccc  /*                          */
          )                 /*                          */
      SUP (                 /* SUP some SYSMODs.        */
           ddddddd ddddddd  /* Fixes incorporated into  */
           ddddddd ddddddd  /* 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
```

Figure 32. Sample APAR SYSMOD Format

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 check to 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 an absolute requirement, but if 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 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.

Prepare Your System

Corrective service has different characteristics from the installation of a new function or preventive service.

- It usually affects a very limited area of the system.
- It is usually done because there is a severe problem currently affecting the system.
- There is a need for an **immediate** fix.
- The time necessary to install the fix is usually small.

Thus, there is usually no need or time to prepare the system (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.

Stage the SYSMODs: The RECEIVE Process

After verifying that the corrective SYSMOD is syntactically correct and specifies the proper set of functions and PTFs, you receive that SYSMOD (APAR or PTF) into the SMP/E database, so that you can then install it into the target libraries.

Because you are dealing with a very small number of SYSMODs (often only one), the job for receiving it is very simple. Use the SMP/E dialogs or the sample job shown in Figure 33 to receive the corrective service.

```
//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.  */
         .                   /*                          */
/*
```

Figure 33. Sample Job for Receiving Corrective Service

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 contained only those SYSMODs that you are installing for this corrective service problem, you can omit the SELECT operand, in which case SMP/E attempts to process all SYSMODs in the SMPPTFIN input data set.

Update the Target Libraries: The APPLY Process

After receiving the corrective service, you are ready to install it into the target libraries. You should not attempt to install the SYSMODs without first verifying them. If you have done all the proper checking before this time, the SYSMODs should be installed correctly. However, if you have overlooked something, the direct installation may cause unexpected results.

Check the Update (APPLY CHECK)

The purpose of this job is to verify that the SYSMOD(s) can be installed correctly and that you understand what libraries and load modules in the system are affected. Use the SMP/E dialogs or the sample job shown in Figure 34 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. */
                XXXXXXX     /* Specify SYSMOD name here.*/
               )            /*                          */
         CHECK              /* In check mode only.      */
         REDO               /* Only if SYSMOD is being  */
                            /* reinstalled.             */
                            /*                          */
```

Figure 34. Sample Job for Applying Corrective Service (Check Mode)

Normally, you do not need the check run because you are installing only a few SYSMODs in response to a system problem.

Research 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.
- Are any SYSMODs going to be regressed? If so, determine how to resolve the problems.
- Are there any other areas of the system that are affected?

Using HOLDDATA to Assist in Identifying Fixes: If the SYSMOD you are installing is a PTF type SYSMOD (obtained from a CBPDO, a PUT, or directly from the IBM Support Center), SMP/E may have some HOLDDATA stored that is applicable to that PTF. If so, the reports will indicate all the reason IDs that prevent PTF installation. You should use these reason IDs to determine what the errors are. This can be done by:

1. Listing the SYSMOD and MCS entries for the PTF.
2. Looking at the ++HOLD statements that are listed.
3. Using 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, you should contact the IBM Support Center to obtain further information.
4. Determining if 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 existing problem, you should perform this step at some later date.

Get Additional SYSMODs

If the research of the APPLY CHECK reports indicate that additional SYSMODs are required, these should be obtained in the same manner as the original corrective SYSMOD.

Update the Target Library (APPLY)

Once the APPLY CHECK has run to your satisfaction, you are ready to install the fix. Use the SMP/E dialogs or the sample job shown in Figure 35 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. */
               xxxxxxx      /* Specify SYSMOD name here.*/
               )            /*                         */
         REDO               /* Only if SYSMOD is being  */
                            /* reinstalled.             */
                            /* Note no check operand.   */
         .                  /*                         */
```

Figure 35. Sample Job for Applying Corrective Service (Update Mode)

Test the Corrective Service

The testing done after installing a corrective fix depends on the type of problem that you encountered, ranging from no testing to running a job that was producing the error.

Update the Distribution Libraries: The ACCEPT Process

Once you have installed the corrective service into the target libraries, you must decide if you want to update the distribution libraries. This decision is up to you, based on the products involved and 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. Once you have accepted 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 more SYSMODs that have been applied but not accepted, the more work must be done to perform the restore.** 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 steps should be used, assuming that you have decided to accept some corrective service.

Check 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. Use the SMP/E dialogs or the sample job shown in Figure 36 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. */
                xxxxxxx     /* Specify SYSMOD name here.*/
               )            /*                          */
         CHECK              /* In check mode only.      */
         .                  /*                          */
```

Figure 36. Sample Job for Accepting Corrective Service (Check Mode)

Research the ACCEPT CHECK Reports

You should research the ACCEPT CHECK reports in the same manner as you did in the APPLY process (see "Research the APPLY CHECK Reports" on page 91).

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. Additional information may be reported if you have processed any HOLDDATA between the APPLY and ACCEPT. This information should be handled in the same manner as the APPLY information.

Get Additional SYSMODs

If additional SYSMODs are required in order to ACCEPT the corrective service, you should obtain them in the same manner that you obtained the original corrective service SYSMOD.

Note: If you obtain additional SYSMODs, you should make sure you process them through the APPLY and test phases before accepting them.

Update 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 sample job shown in Figure 37 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. */
                xxxxxxx     /* Specify SYSMOD name here.*/
               )            /*                         */
                            /* Note no check operand.  */
         .                  /*                         */
```

Figure 37. Sample Job for Accepting Corrective Service (Update Mode)

Chapter 7. Installing a User Modification

This chapter describes the steps for installing a user modification. It describes:

- An introduction to USERMODs
- Preparing your system
- Staging the user modification with the RECEIVE command
- Updating the target libraries with the APPLY command
- Testing the user modification
- Updating the distribution libraries with the ACCEPT command.

Introduction

A USERMOD-type SYSMOD is a modification that you make 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 user modification should not be confused with an APAR-type SYSMOD (corrective fix), even if the initial version of that fix has been built by you in order to fix a problem immediately. A description of how to construct USERMOD-type SYSMODs can be found in Chapter 16. *SMP/E Program Packaging Guide* contains additional information on when a USERMOD should be used with function SYSMODs and detailed packaging rules for properly constructing a USERMOD. The remainder 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.

Prepare Your System

You must determine the amount of system preparation necessary for your user modification. If it is extensive and affects critical components of the system, you should perform the same tasks as defined under "Prepare Your System" on page 57 or "Prepare Your System" on page 73. If it is a minor change, affecting very few modules and not critical to the operation of the system, no preparation is needed.

Stage the SYSMODs: The RECEIVE Process

Because user-modification SYSMODs are generally processed as a single SYSMOD, processing is very similar to that for corrective service (that is, they are received by use of the SELECT option). Use the SMP/E dialogs or the sample job shown in Figure 38 on page 96 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.*/
               xxxxxxx       /* Specify USERMOD number.  */
               )             /*                          */
         SYSMODS             /* Only process SMPPTFIN -
                                do not look at SMPHOLD.  */
         .                   /*                          */
/*
```

Figure 38. Sample Job for Receiving a User Modification

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 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 those USERMODs that you want to receive now, you can omit the SELECT operand, in which case SMP/E attempts to process all SYSMODs in the SMPPTFIN input data set.

Update the Target Libraries: The APPLY Process

After having received the user modification, you are ready to install it into the target libraries. You may be tempted to install the SYSMODs without first having performed 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.

Check 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 sample job shown in Figure 39 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. */
               xxxxxxx       /* Specify SYSMOD name here.*/
               )             /*                          */
         CHECK               /* In check mode only.      */
         .                   /*                          */
```

Figure 39. Sample Job for Applying a User Modification (Check Mode)

Note: At times it may be necessary to reinstall a user modification, 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, depending on the relationship between your user modification and the PTF that was installed.

Research 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, and thus 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.

- Are any SYSMODs going to be regressed? If so, determine how to resolve the problems.

Update the Target Library (APPLY)

Once the APPLY CHECK runs to your satisfaction, you are ready to install the user modification. Use the SMP/E dialogs or the sample job shown in Figure 40 to apply the USERMOD.

```
//APPLY    JOB 'accounting info',MSGLEVEL=(1,1)
//APPLY    EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(TGT1).          /* Set to target zone.      */
APPLY    SELECT(             /* Install selected SYSMOD. */
               xxxxxxx       /* Specify SYSMOD name here.*/
              )              /*                          */
                             /* Note no check operand.   */
         .                   /*                          */
```

Figure 40. Sample Job for Applying a User Modification (Update Mode)

Test the User Modification

The amount of testing needed after the installation of a user modification depends on the changes that you are making.

You may want to review the recommendations found under "Test the New Function" on page 65 and "Test the New Service Level" on page 83.

When originally constructing your user modification, 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 modification 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 modification.

Update the Distribution Libraries: The ACCEPT Process

Once you have installed the user modification into the target libraries, you must decide if you want to update the distribution libraries. This decision is up to you, based on the products involved and your processing requirements.

The following is a consideration **for not accepting** user modifications:

> If a problem is encountered in the modified modules, you may be asked to re-create the problem, using an unmodified version of the module. If you have accepted the user modification, you cannot create an unmodified version of the module, unless you are also maintaining a separate set of distribution libraries without the user modifications.

The following are some of the considerations **for accepting** user modifications:

- If you do not accept the USERMOD and you perform a system generation, that modification is lost and must be reinstalled.
- If you must restore a SYSMOD, the more SYSMODs that are in an APPLY-only state (that is, not yet accepted), the more work you must do to perform the restore (all intersecting SYSMODs must be restored, and then all but the desired SYSMODs must be reapplied). This is especially true for source-modified products.

The following steps should be used, assuming that you have decided to accept the user modification.

Check the Update (ACCEPT CHECK)

The ACCEPT CHECK job provides the same function for the distribution libraries that the APPLY 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. Use the SMP/E dialogs or the sample job shown in Figure 41 to do an ACCEPT CHECK for the USERMOD.

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPTCK EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1).        /* Set to DLIB zone.        */
ACCEPT   SELECT(            /* Install selected SYSMOD. */
                xxxxxxx     /* Specify SYSMOD name here */
               )            /*                          */
         CHECK              /* In check mode only.      */
         .                  /*                          */
```

Figure 41. Sample Job for Accepting a User Modification (Check Mode)

Research the ACCEPT CHECK Reports

You should research the ACCEPT CHECK reports in the same manner as you did the APPLY reports (see "Research the APPLY CHECK Reports" on page 97).

Update 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 sample job shown in Figure 42 to accept the USERMOD.

```
//ACCEPT   JOB 'accounting info',MSGLEVEL=(1,1)
//ACCEPT   EXEC SMPPROC
//SMPCNTL  DD *
SET      BDY(DLIB1).       /* Set to DLIB zone.        */
ACCEPT   SELECT(           /* Install selected SYSMOD. */
              xxxxxxx      /* Specify SYSMOD name here */
             )             /*                          */
                           /* Note no check operand.   */
         .                 /*                          */
```

Figure 42. Sample Job for Accepting a User Modification (Update Mode)

Chapter 8. Managing Exception SYSMODs

This chapter 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. However, for some SYSMODs this is not possible. Examples of such SYSMODs include:

- SYSMODs that were sent out to correct a problem but that either have not fixed the problem or have introduced a new problem. These are called *PTFs in error*, or *PE-PTFs*.
- SYSMODs that require special SMP/E processing, such as UCLIN.
- SYSMODs that require special installation processing, such as an I/O generation or full SYSGEN, or 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*. SMP4 had no way of managing these SYSMODs—you had to manage them yourself. 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 any type of 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 **must** be built as part of the held PTF. ++RELEASE statements and ++HOLD statements for error or user holds may be in the SMPHOLD data set. ++HOLD and ++RELEASE statements identify the following:

- The SYSMOD ID of the exception SYSMOD.
- 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.
  - For **user**-category exception SYSMODs, SMP/E makes no assumption about what the reason ID represents.

    For more information about reason IDs, see *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, an alternative way to release a held SYSMOD. For example, a SYSMOD may require UCLIN changes unless a SYSGEN is done. The ++HOLD statement for that SYSMOD would have a SYSTEM reason ID of UCLIN and a CLASS of SYSGEN.

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 sections of this chapter 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 *SMP/E Reference* 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 specified by the SMPHOLD DD statement.

You may specify one or both operands on the RECEIVE command. If neither operand is specified, SMP/E attempts to receive both 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 an exact copy 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 the SMPHOLD data set. The name of the entry is 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. The appropriate BYPASS operand should be used 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.

In order for an error category 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 may specify BYPASS(HOLDCLASS) on APPLY to indicate that you will be using an alternative way to resolve the reason ID.

See the BYPASS operand in the chapter on the APPLY command in *SMP/E Reference* for additional information.

System and user category reason IDs are resolved by specifying BYPASS(HOLDSYSTEM) or BYPASS(HOLDUSER) on the APPLY command. If there is a HOLD CLASS associated with the reason ID, you may specify BYPASS(HOLDCLASS) on APPLY to indicate that you will be 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 *SMP/E Reference* for more information on the reason ID naming conventions.

If all reason IDs are resolved, SMP/E allows the SYSMOD to be applied. If any reason IDs remain unresolved, SMP/E prevents the SYSMOD, and any other SYSMODs dependent on this one, from being installed.

SMP/E does not remove the reason IDs from the global zone SYSMOD entry when the SYSMOD is applied, so that if it is applied on another system later, the same checking will be done on that system. If the information had been deleted during the first APPLY, SMP/E would not recognize the problem when 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 very important that SMP/E have the most current exception SYSMOD information available (see "How to Process the HOLDDATA" on page 107 for more information on the importance of having current HOLDDATA and what you **must do** to provide that information to SMP/E).

Updating Distribution Libraries: ACCEPT

Exception SYSMOD processing is the same when accepting a SYSMOD as when applying a SYSMOD, except that the appropriate distribution zone is used to determine if 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.

After a Successful ACCEPT

When accepting a SYSMOD, if SMP/E determines that the global zone SYSMOD entry and the SMPPTS MCS entry are to be deleted, then SMP/E deletes any HOLDDATA associated with those SYSMODs. Once you purge the SYSMOD and MCS entries you will probably not install them on any other systems, so you would not need the HOLDDATA again.

During RESTORE Processing

HOLDDATA is never deleted during RESTORE processing, under the assumption that you may later want to reapply the SYSMODs that 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 notify SMP/E that besides deleting the global zone SYSMOD and SMPPTS MCS entries, any associated HOLDDATA is also to be deleted.

Sources of HOLDDATA

Besides the data you create, these are the main sources of HOLDDATA provided by IBM:

- CBPDO tapes
- Program update tapes (PUTs)
- Preventive service planning (PSP) information from the CSSF files
- Cumulative service (CUM) tape.

This section describes these sources.

CBPDO Tapes

One of the primary means of obtaining HOLDDATA is CBPDO tapes. IBM custom-builds these tapes on the basis of the products and service you request, and on whether this is your first CBPDO order.

- If you select a PUT 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 PUT service level that was shipped in your previous CBPDO.

The HOLDDATA on a CBPDO tape 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 tape should be processed immediately on receipt of the tape. You may use either the SMP/E dialogs or the RIMLIB jobs provided with the CBPDO tape to receive the HOLDDATA. (See *MVS Custom-Built Offerings Planning and Installation* and *CBPDO Memo to Users Extension* for more information.)

PUTs

Another means of obtaining HOLDDATA is the program update tapes (PUTs). IBM creates these tapes periodically and ships them from software distribution to all users. The tapes contain all the preventive service (PTFs) and HOLDDATA eligible for inclusion since the last tape was built.

Table 7 on page 72 defines the contents of each file on the PUT.

Note: For this discussion, file 4 of the PUT contains the HOLDDATA.

The HOLDDATA on your tape has been customized for your installation. That is, it contains only data applicable to PTFs for those products you have defined in your software distribution profile. Therefore, it is very important to keep your profile current. Your IBM representative can help you with any questions you may have regarding your software distribution profile.

The HOLDDATA on the PUT should be processed immediately on receipt of the tape. (See "Processing HOLDDATA from a PUT" on page 109 and "Example of Processing HOLDDATA" on page 112 for more information on processing the PUT.)

PSP Information

Once a PUT has been shipped to an IBM software distribution center for distribution, there is no further opportunity to change the HOLDDATA on that tape even though new errors are reported. It is, however, important that you have this information when you are about to install a CBPDO or PUT. Otherwise, you may direct SMP/E to install a PE-PTF, thereby introducing a problem besides fixing one. To solve this problem, PSP files have been set up to hold this additional HOLDDATA. One file exists for each PUT.

Once a PUT is shipped, a PSP file is created. As problems (APARs) are reported, appropriate ++HOLD statements are added to the applicable PSP files. IBM determines the applicable PSP files as follows:

1. Determine the PTF that introduced the APAR.
2. If the PTF is not yet available on a PUT, add a ++HOLD statement to the CORPE PSP file.

   Otherwise, continue with the next step.
3. If the PTF is available on a PUT, determine the PUT on which that PTF was first shipped.
4. Add a ++HOLD statement to the PSP file corresponding to that PUT.
5. Add a ++HOLD statement to each PSP file corresponding to any PUTs shipped after the one determined in the previous step, up to the current PUT.
6. Add a ++HOLD statement to file 4 of the next PUT shipped.
7. No further PSP files will be updated.

Before installing a CBPDO tape, PUT, or PTFs in corrective mode, use CSSF Information/Access or SoftwareXcel Extended, or contact the IBM Support Center to get the information in the applicable PSP file. See "Processing HOLDDATA from PSP Files" on page 110 and "Example of Processing HOLDDATA" on page 112 for more information on how to use the PSP information.

CUM Tapes

There may also be HOLDDATA on the CUM tape that you may receive when you order a new function. Normally, when a developer first delivers a product to an IBM software distribution center, there are no PTFs applicable to the product, and hence no HOLDDATA. Therefore, in general, early users of a product receive only the function tape.

As the product is used and problems are reported, PTFs are developed in response to those problems. These PTFs, in turn, can have problems reported against them, resulting in the creation of exception SYSMOD data.

Now, users who order the product receive not only the function tape but also a tape containing any PTFs applicable to the function. This tape is called the *CUM tape*. The tape contains not only the PTFs but also the HOLDDATA applicable to those PTFs. The format of the tape is described in Table 5 on page 60. The format is the same as the first six files of a normal PUT. Therefore, the method of processing the HOLDDATA for the CUM tape is the same as that for the PUT. See "Processing HOLDDATA from PSP Files" on page 110 and "Example of Processing HOLDDATA" on page 112 for more information on how to use the CUM tape HOLDDATA.

Periodically, the PTFs on the CUM tape are integrated into the product tape itself. This is done by replacing the elements in the product tape with the corresponding elements from the PTF with the highest service level and then updating the SMP/E install logic to reflect the integration of this service. SMP/E is used to do this service integration, thereby making sure no PTFs are integrated without having resolved any HOLDDATA. When a PTF is integrated, any HOLDDATA associated with that PTF is deleted from the exception SYSMOD file of the CUM tape.

Note: To integrate a PTF with HOLDDATA, that exception condition must have been resolved as described under "What SMP/E Does with the HOLDDATA" on page 102 and is thus no longer required.

How to Process the HOLDDATA

The management of exception SYSMODs is a very 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 time frame in 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, PUT, or CUM tapes **in PUT level order**. Remember to do the following:
   - Receive all new products as you get them. This allows you to process exception SYSMODs for them.
   - Receive HOLDDATA as soon as you get your CBPDO, PUT, or CUM tapes.
3. Before doing preventive service, do the following:
   a. Get the PSP file associated with the last PUT level for which you processed HOLDDATA, and receive this additional HOLDDATA for your products.

   b. Get the CORPE PSP file. This contains PE-PTFs that are available correctively but are not yet available on a PUT.

   c. 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 helps makes sure all available service is installed when you do preventive service.

Be sure that you review "Example of Processing HOLDDATA" on page 112. This example shows why you should follow the procedures defined.

Processing HOLDDATA from a CBPDO Tape

When you receive a CBPDO tape, 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 tape **as soon as you get the tape**. Use the SMP/E dialogs or the RIMLIB job provided with the CBPDO tape.

   By receiving the HOLDDATA as soon as possible, you make sure SMP/E has the most current information available. Therefore, if you try to install any PTF in response to a problem on your system, and that PTF is in error, SMP/E will know this and warn you so that you can assess the effect of installing the known problem as opposed to fixing the problem you have encountered.

2. Receive the SYSMODs from the CBPDO tape **as soon as you get the tape**. Use the SMP/E dialogs or the RIMLIB job provided with the CBPDO tape.

   This makes sure 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 may 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 versus system programmer time as increased research time looking for the PUT with the PTF you need and fixing problems caused by installing PE-PTFs.

One important part of this procedure is that the HOLDDATA on each CBPDO and PUT **must be received in chronological order**. SMP/E processes the ++HOLD and ++RELEASE statements in the order they are encountered. Therefore, there could be an exposure if you receive the data out of sequence. For instance, the tapes may be set up so one tape contains a ++HOLD for a PTF and a subsequent tape contains a ++RELEASE for the same PTF. If the tapes are processed in the wrong order, the RELEASE statement will be processed first, then the HOLD statement. As a result, the PTF will remain held.

Processing HOLDDATA from a PUT

When you receive your preventive service tape (also called the **program update tape**, or **PUT**), one of the files (file 4) contains the HOLDDATA generated during the previous service period (see "PUTs" on page 106 for additional information). The HOLDDATA on that tape pertains both to the PTFs actually on that tape and to PTFs shipped on previous tapes.

In previous SMP releases, you were not required to do anything with the PUTs until you were actually ready to install it (possibly several months after initial receipt). This was true because once you received the PUT, to mass-apply one PUT, you had to mass-apply all PUTs that had been received. However, with the SMP/E SOURCEID operand (on the RECEIVE, APPLY, and ACCEPT commands) this restriction has been eliminated.

Although you can still process the PUTs in this manner (that is, store them until ready to actually install), SMP/E and the PUT exception SYSMOD support have been designed to work most efficiently and effectively if you adhere to the following processing guides:

1. Receive the HOLDDATA file (file 4) of the PUT **as soon as you get the tape**. Use the SMP/E dialogs or the sample job shown in Figure 43.

```
//RECHOLD  JOB 'accounting info',MSGLEVEL=(1,1)
//RECEIVE  EXEC SMPPROC
//SMPHOLD  DD ...data describing PUT exception file
//SMPCNTL  DD *
SET      BDY(GLOBAL).       /* Set to global zone.   */
RECEIVE  HOLDDATA.          /* Receive only exception
                               SYSMOD data.          */
/*
```

Figure 43. Sample Job for Receiving HOLDDATA from a PUT

   By receiving the HOLDDATA as soon as possible, you make sure SMP/E has the most current information available. Therefore, if you try to install any PTF in response to a problem on your system, and that PTF is in error, SMP/E will know this and warn you so that you can assess the effect of installing the known problem as opposed to fixing the problem you have encountered.

2. Receive the SYSMOD file (file 1) of the PUT **as soon as you get the tape**. Use the SMP/E dialogs or the sample job shown in Figure 44.

```
//RECPTFS  JOB 'accounting info',MSGLEVEL=(1,1)
//RECEIVE  EXEC SMPPROC
//SMPPTFIN DD ...data describing sysmods file
//SMPCNTL  DD *
SET      BDY(GLOBAL).        /* Set to global zone.   */
RECEIVE  SYSMODS             /* Receive only SYSMODs. */
         SOURCEID(PUTxxxx).  /* Assign appropriate
                                source ID value.      */
LIST     SYSMODS             /* Now list the SYSMODs  */
         MCS                 /* including SMP MCS     */
         SOURCEID(PUTxxxx).  /* for those SYSMODs just
                                received.             */
/*
```

Figure 44. Sample Job for Receiving SYSMODs from a PUT

This makes sure 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.

Although separate jobs are shown to receive the SYSMOD and HOLDDATA file, they can be done in one RECEIVE command, by specifying both the SYSMODS and HOLDDATA operands. In fact, this is the preferred method and is the default if neither operand is specified.

By following these procedures you are essentially making a trade-off: system resources as increased DASD space for the SMPPTS versus system programmer time as increased research time looking for the PUT with the PTF you need and fixing problems caused by installing PE-PTFs.

One important part of this procedure is that the statements in the HOLD file on each PUT **must be received in chronological order**. SMP/E processes the ++HOLD and ++RELEASE statements in the order that they are encountered. Therefore, there could be an exposure if you receive the data out of sequence. For instance, the tapes may be set up so one tape contains a ++HOLD for a PTF and a subsequent tape contains a ++RELEASE for the same PTF. If the tapes are processed in the wrong order, the RELEASE statement will be processed first, then the HOLD statement. As a result, the PTF will remain held.

Processing HOLDDATA from PSP Files

Another source of exception SYSMOD data is the PSP file, available through CSSF Information/Access, through SoftwareXcel Extended, or by request from your IBM Support Center. For each PUT that is applicable to a specific environment, there is a PSP file that contains additional HOLDDATA. This file contains all the ++HOLD and ++RELEASE statements applicable to PTFs either on that PUT or on PUTs before that one. You should process this PSP file before you install a PUT, or a CBPDO tape that includes PTFs from that PUT.

When you are ready to install a CBPDO or PUT you must do the following:

1. Make sure you have received the HOLDDATA from, at minimum, all the CBPDOs and PUTs up to the PUT level of the tape you are installing. You should receive the HOLDDATA from **all the available tapes** to reduce the amount of data that you have to get from PSP.
2. Contact the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended, to get the latest CORPE PSP file, as well as the PSP file associated with the **latest PUT level for which you have received HOLDDATA** (not the PSP file for the PUT level that you are installing).
3. Create a data set containing the ++HOLD statements obtained from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.
4. Receive that data set, using the SMP/E dialogs or the sample job shown in Figure 45.

```
//RECHOLD  JOB 'accounting info',MSGLEVEL=(1,1)
//RECEIVE  EXEC SMPPROC
//SMPHOLD  DD ...data describing your data set
//SMPCNTL  DD *
SET      BDY(GLOBAL).        /* Set to global zone.   */
RECEIVE  HOLDDATA.           /* Receive only exception
                                SYSMOD data.          */
/*
```

Figure 45. Sample Job for Receiving HOLDDATA from a PSP File

You should also use the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended to get PSP information whenever you are installing corrective service. Before installing a PTF in corrective mode, determine the PUT on which it was initially shipped. Then contact the IBM Support Center to get the PSP data associated with that PUT level to determine if there are any ++HOLD statements for that PTF. If so, process them, and install the PTF.

Processing HOLDDATA from a CUM Tape

When you order a new software product from IBM you receive two tapes:

- The function tape itself
- The CUM tape for the function.

The CUM tape contains any PTFs applicable to the function that have not already been integrated into it by the service update process (SUP process). The format of this tape is the same as a normal PUT, except that the PUT documentation and program files are empty.

As soon as you receive your new function order, you should do the following:

1. Either receive the function or use the UCLIN command to add the FMID of the new function into the global zone GLOBALZONE FMID subentry list. This allows you to receive service applicable to that function.
2. Receive file 4 of the CUM tape to provide SMP/E with all available HOLDDATA for the new function. This step is required.
3. Receive file 1 of the CUM tape to make all the current PTFs for the function available to SMP/E.

If the new function arrives at the same time as a PUT, the new function package should be processed first so that any PTFs on the PUT can also be received.

As in processing the PUT, the purpose here is to make sure the HOLDDATA is kept as up-to-date as possible.

Example of Processing HOLDDATA

Assume you are ready to actually install a CBPDO or PUT. The following example may help you understand the reasons for the recommendations made in this chapter. In this example:

1. Table 8 on page 113 shows the exception SYSMOD information. The PTFs and PSP files are as follows:
   a. Column 1 lists the three PUT service levels involved in this example.
   b. Column 2 lists the five SYSMODs contained in the PUT service level.
   c. Column 3 lists the ++HOLD statements contained on the CBPDO/PUT.

      For simplicity, there are no PE-PTFs before the first PUT service level in the example (PUT9101). The exact syntax and APAR numbers for the ++HOLD are not significant for this example.
   d. Column 4 lists the ++HOLD statements contained in the PSP file associated with each of the PUT service levels. The exact syntax and APAR numbers for the ++HOLD are not significant for this example.
2. The SYSMODs have been marked PE as follows:
   - As of the PUT9101 service level, there were no PTFs in error.
   - Between the PUT9101 and the PUT9102 service levels, PTFs UR00002 and UR00003 were marked as PE.
   - Between the PUT9102 and the PUT9103 service levels, PTF UR00005 was marked PE.
   - PTF UR00001 was marked as PE at the PUT9103 service level.

3. Table 8 shows the contents of each of the files at some time after the shipment of PUT9103.

Table 8. CBPDO/PUT/PSP HOLDDATA Example

Service Level	PTFs Per Service Level	PTFs with HOLDDATA	HOLDDATA in PSP File for the Source ID Service Level
PUT9101	UR00001 UR00002 UR00003 UR00004 UR00005		UR00001 UR00002 UR00003 UR00005
PUT9102	UR00006 UR00007 UR00008 UR00009 UR00010	UR00002 UR00003	UR00001 UR00005
PUT9103	UR00011 UR00012 UR00013 UR00014 UR00015	UR00005	UR00001

4. You are now trying to install PTFs at service level PUT9101. Depending on what you had done with PTFs in PUT9102 and PUT9103, the amount of processing you have to do before installing PUT9101 level PTFs varies.

   - If you have received the HOLDDATA (file 4) from service levels 9101, 9102, and 9103, you just have to process the one ++HOLD statement for PTF UR00001 from the PSP file for PUT9103.
   - If you have received only the HOLDDATA from PUT9102, you have to process the two ++HOLD statements for PTFs UR00001 and UR00005 from the PSP file for PUT9102.
   - If you had decided not to process any data on the service level tapes until actually ready to install them (that is, you had done nothing with PUT9102 and PUT9103), you have to process the four ++HOLD statements for PTFs UR00001, UR00002, UR00003, and UR00005 from the PSP file for PUT9101.

   Note: In each case, you used the PSP file associated with the last service level from which you received the HOLDDATA, but if you had kept current in processing the exception SYSMOD files from the service levels, there would have been less information that had to be obtained from the IBM Support Center.

In this example, the number of PTFs and HOLDDATA was small and thus the data seems manageable. However, with a real service level with hundreds of PTFs, the amount of manual work involved in getting the ++HOLD statements from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended and then keying them into a data set and receiving them could be very time consuming. So, normally, the cost of the increased DASD space necessary to store the HOLDDATA each month is paid back in increased programmer productivity when the service level is to be installed.

Chapter 9. Creating Target Zones After Using a Special Generation Procedure

This chapter discusses how to create the appropriate entries in a target zone after you have installed SYSMODs using one of the special generation methods described in Chapter 4. The following sections tell how to initialize entries when you are:

- Defining a new target zone
- Updating an existing target zone.

When Defining a New Target Zone

You may need to define a new target zone in these cases.

- When performing a full-system generation
- When installing a product after system or subsystem generation
- When using the GENERATE command
- When reinstalling the entire product.

After Full-System Generation

After performing a full-system generation, you must create a new target zone describing the set of target libraries created during the system generation process. The examples on the next pages show the steps necessary to do this, using one of the following:

- The ZONECOPY command
- The ZONEEXPORT and ZONEIMPORT commands
- The ZONERENAME and ZONEMERGE commands.

Here are some considerations as to which method to use:

- ZONECOPY can be used only when the distribution zone and the previous target zone reside in an SMPCSI data set different from that of the new target zone. ZONECOPY provides the best performance and also creates a backup copy of the zone.
- ZONEEXPORT and ZONEIMPORT can be used regardless of where the distribution zone and old and new target zones reside. ZONEEXPORT and ZONEIMPORT, in addition to copying the zones, also provide you with a backup copy of the exported zone.
- ZONERENAME and ZONEMERGE can be used when the target and distribution zones are in the same SMPCSI data set.

For more information about the zone utility commands, see the chapters on the ZONECOPY command, the ZONEIMPORT command, the ZONEEXPORT command, and the ZONEMERGE command in *SMP/E Reference*.

Note: After doing a full-system generation, in addition to allocating a new target zone and copying the distribution zone to the new target zone, you must also allocate new SMPMTS and SMPSTS data sets. You must also allocate new data sets unique to the target zone, such as SMPLOG, SMPLOGA, and SMPSCDS. If you do not start your new system with an empty SMPMTS and SMPSTS, you may inadvertently regress your new system when service is applied.

Example 1: Using the ZONECOPY or ZONEEXPORT and ZONEIMPORT Commands

1. Define the new target zone. Assume:

   - The previous target zone was named MVSTGT1.
   - The new target zone is named MVSTGT2 and was from distribution zone MVSDLIB.
   - The OPTIONS entry used is MVSOPT.
   - MVSTGT2 is created in SMPE.SMPCSI.CSI.

   Note: If the SMPCSI you are importing to is new, you have to allocate the CSI and prime it with GIMZPOOL.

   The following SMP/E commands define the new target zone:

```
SET      BDY(GLOBAL).       /* Set to global.            */
UCLIN    .                  /* UCLIN to def new zone.    */
ADD      GZONE              /*                           */
         ZONEINDEX(         /* Define zone in CSI.       */
                   (MVSTGT2,SMPE.SMPCSI.CSI,TARGET) /*   */
                  )         /*                           */
         .                  /* Define new zone.          */
ENDUCL   .                  /*                           */
```

2. Copy the information from the distribution zone into the new target zone. This can be done in one of two ways:

   - If the new target zone, MVSTGT2, and the distribution zone exist in different SMPCSI data sets you can use the ZONECOPY command as follows:

```
SET      BDY(MVSTGT2).      /* Set to target zone.     */
ZONECOPY (MVSDLIB)          /* Copy DLIB zone          */
         INTO(MVSTGT2)      /*  into new target zone.  */
         RELATED(MVSDLIB)   /* Same info as in         */
         OPTIONS(MVSOPT).   /*  old zone.              */
```

   - If the new target zone, MVSTGT2, and the distribution zone exist in the same SMPCSI data set, you must use the ZONEEXPORT and ZONEIMPORT commands as follows:

```
SET      BDY(MVSDLIB).     /* Set to DLIB zone.         */
ZEXP     (MVSDLIB)         /* Export DLIB zone          */
         OFILE(DD1).       /*  to DD1 sequential file.  */
SET      BDY(MVSTGT2).     /* Set to target zone.       */
ZIMP     (MVSDLIB)         /* Import DLIB zone          */
         INTO(MVSTGT2)     /*  into new target zone     */
         IFILE(DD1)        /*  from DD1 sequential file.*/
         RELATED(MVSDLIB)  /* Same info as in           */
         OPTIONS(MVSOPT).  /*  old zone.                */
```

At this point, your new target zone contains all the information about the function and service levels of the elements in your new target libraries.

3. Copy the target library information from your old target zone into your new target zone. This can be done using the ZONEMERGE command as follows:

```
SET      BDY(MVSTGT2).     /* Set to new zone.        */
ZONEMERGE(MVSTGT1)         /* Merge from MVSTGT1      */
         INTO(MVSTGT2)     /*  into new zone MVSTGT2. */
         DEFINITION        /* Definition only         */
         REPLACE           /*  with replace option.   */
         .                 /*                         */
```

The only entries that will be copied in this case are the DDDEF entries. These entries should not have changed during the system-generation process.

If you have changed the unit or volume serial number of the target volumes, and the DDDEF entries describing those libraries also contain the UNIT or VOLUME information (that is, the DDDEF entries did not assume that the data sets were cataloged), these entries must be modified with either ZONEEDIT or UCLIN to reflect the new data.

Note: Use ZONEEDIT or the dialogs to make a mass change rather than using several UCLIN commands.

The following is an example of how to modify the DDDEF entries using ZONEEDIT to change both the UNIT and VOLUME information.

```
SET      BDY(MVSTGT2).     /* Set to new target zone. */
ZONEEDIT DDDEF.            /* All DDDEF entries       */
CHANGE   UNIT(*,3380)      /*  now on UNIT 3380s      */
         VOLUME(*,TGTPCK). /*  and VOLUME TGTPCK.     */
ENDZONEEDIT.               /*                         */
```

4. Now that the target zone is defined and primed with the nonstructure entries from the previous target zone, prime the new target zone with the structure information about the entries in the new target libraries. Do this by using the JCLIN command of SMP/E with the Stage 1 generation output being used as input. If the Stage 1 generation output JCL was saved in data set STG1.MVSTGT2.CNTL, and you have an SMPPROC similar to the one described under "Calling SMP/E" on page 35, the following job will prime the new target zone:

```
//JOB1     JOB 'accounting info',MSGLEVEL=(1,1)
//STEP1    EXEC SMPPROC
//SMPJCLIN DD DSN=STG1.MVSTGT2.CNTL,DISP=SHR
//SMPCNTL  DD *
SET      BDY(MVSTGT2).      /* Set to new target.       */
JCLIN    .                  /* Update zone.             */
LIST     .                  /* List zone.               */
/*
```

5. The new target zone, MVSTGT2, is now ready to be used for the installation of service or new functions. When this system has been tested and the old level, MVSTGT1, is no longer required, delete it using the ZONEDELETE command. This can be done as follows:

```
SET      BDY(MVSTGT1).      /* Set to old target.       */
ZONEDELETE                  /*                          */
         TZONE(MVSTGT1).    /* Delete it.               */
```

Example 2: Using the ZONERENAME and ZONEMERGE Commands

This example creates a new target zone to replace an existing target zone. Example 1 used SMP/E zone commands to create a new target zone; this example uses access method services.

1. Create a total copy of the SMPCSI data set containing the distribution zone describing the distribution libraries used to create the new target libraries. Assume:

   - The previous target zone was named MVSTGT1.
   - The new target zone is named MVSTGT2 and is generated from distribution zone MVSDLIB, which exists in SMPE.DLIBCSI.CSI.
   - The new target zone is to exist in data set SMPE.TGTCSI.CSI.

   Define a new VSAM data set and use the access method services REPRO command as follows:

   ```
   //REPRO    JOB 'accounting info',MSGLEVEL=(1,1)
   //STEP1    EXEC PGM=IDCAMS
   //OLDCSI   DD DSN=SMPE.DLIBCSI.CSI,DISP=OLD
   //NEWCSI   DD DSN=SMPE.TGTCSI.CSI,DISP=OLD
   //SYSPRINT DD SYSOUT=A
   //SYSIN    DD *
     REPRO                  -
           INFILE(OLDCSI)  -
           OUTFILE(NEWCSI)
   /*
   ```

   You now have a copy of the SMPCSI data set containing the distribution zone used to generate your new target libraries. The objective now is to transform the copy of the distribution zone into a target zone that actually describes the function, service level, and system structure of the new target libraries.

2. Make the copied distribution zone a target zone and connect that target zone through the global zone ZONEINDEX entries. This can be done using the ZONERENAME command as follows:

   ```
   SET      BDY(GLOBAL).       /* Set to global zone.     */
   ZONERENAME(MVSDLIB)         /* Rename DLIB zone        */
            NEWDATASET(        /*  existing in the copy of */
               SMPE.TGTCSI.CSI /*  the CSI data set.      */
                          )    /*                         */
            TO(MVSTGT2)        /* New target name and     */
            TOTYPE(TARGET)     /*  change type to target  */
            RELATED(MVSDLIB)   /*  and connect to old DLIB.*/
            OPTIONS(MVSOPT)    /* Use different           */
                               /*  options entry.         */
                               /*                         */
   ```

   After this command runs, the global zone will contain a ZONEINDEX entry for MVSDLIB indicating that it still resides in the original SMPCSI data set, an entry for MVSTGT1, and an entry for the new zone MVSTGT2, indicating that the zone exists in data set SMPE.TGTCSI.CSI.

3. Delete any other zones that may be in the copied SMPCSI data set. This can be done by also renaming those zones, similar to the way MVSDLIB was renamed, and then using the ZONEDELETE command to delete those zones from the copied SMPCSI data set. When renaming the extra zones, remember to use names that are not currently defined in the global zone ZONEINDEX subentries.

4. Copy the DDDEF entries from your old target zone to your new target zone. This can be done using the ZONEMERGE command as follows:

```
SET      BDY(MVSTGT2).       /* Set to new zone.          */
ZONEMERGE(MVSTGT1)           /* Merge from MVSTGT1        */
         INTO(MVSTGT2)       /*  into new zone MVSTGT2.   */
         DEFINITION          /* Definition information.   */
                             /*                           */
```

   If any of the DDDEF entries copied during the ZONEMERGE operation had the unit and volume specified, those entries should be modified to reflect the unit and volume of the new target libraries.

```
SET      BDY(MVSTGT2).       /* Set to new target zone  */
ZONEEDIT DDDEF.              /* all DDDEF entries       */
CHANGE   UNIT(*,3380)        /* now on UNIT 3380s       */
         VOLUME(*,TGTPCK).   /* and VOLUME TGTPCK.      */
ENDZONEEDIT.                 /*                         */
```

   Your new target zone, MVSTGT2, now contains information about the function and service levels of each element of your new target libraries (because that information was in the distribution zone that you used to create the new target zone). What is missing is the information describing how the elements (modules, macros, and source code) from the installed products are installed in the new target libraries. This information is added with the JCLIN command, as defined in the previous example.

   Your new target zone is now ready for use.

Installation of a Product after System or Subsystem Generation

Some products are not included in a system or subsystem generation procedure. Reinstallation of these products after a complete system generation (both Stage 1 and Stage 2) can be a long and error-prone process. If the product was accepted, the problems created are further complicated by incomplete data in the target zone. When the distribution zone is copied to the target zone, the product that was accepted to the distribution zone now appears to be installed in the system, when in fact it is not.

There are two methods of getting the product installed into the new target libraries:

- Using the GENERATE command
- Reinstalling the entire product.

Note: It is recommended that the first method be used because it is less prone to error.

Using the GENERATE command

GENERATE eliminates the need to reinstall products that are not included by a generation procedure. See "RECEIVE-ACCEPT-Stage 1 Generation-JCLIN-GENERATE Method" on page 52 for the steps you follow.

Reinstalling the Entire Product

1. Obtain the function tape.
2. Make sure the function has been rejected.
3. Obtain all service tapes pertaining to the product that will bring it up to the same service level that existed before the system generation.
4. Build an SMP/E job to receive the function and all service.
5. Build an SMP/E job to apply the product selectively using the REDO and BYPASS keywords.

When Updating an Existing Target Zone

You may need to update an existing target zone after a partial system generation.

After Partial System Generation

If an I/O or nucleus generation is performed, see the applicable system generation manual for further directions on required SMP/E processing.

I/O (Device) Generation

Note: Accept or restore all service and products **before** an I/O generation to avoid regression.

After an I/O (device) system generation, the Stage 1 generation output JCL must be used as input for JCLIN processing to ensure that:

- The module, macro, and load module entries in the target zone are updated.
- The new assembler entries are stored with the new assembler input in the target zone.
- The linkage editor control statements for load module entries are replaced, except for linkage editor CHANGE and REPLACE control statements that were carried over to the updated version.

Chapter 10. Displaying the Data Managed by SMP/E: The LIST Command

This chapter discusses the LIST command. It discusses these topics:

- An introduction to the LIST command
- 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 will be 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. If recently, that change may have caused the problem.
- After reporting the problem, you may 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 having determined the problem in the module, you may want to know if any other parts of the system are affected by that module.

All of this information is available in the SMP/E database. This information can be obtained by using the SMP/E dialogs as you are debugging the problem. You can also use the SMP/E LIST command to create hard-copy listings of the information.

The LIST command allows you to display all entries of a specified type or specific entries. The following sections demonstrate the flexibility of the LIST command. See the chapter on the LIST command in *SMP/E Reference* for a complete description of all the LIST command operands.

Listing All the SMP/E Data

When 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 using the SMP/E dialogs during the debugging process. However, if the system is not running, the information will not be 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 hard-copy listing. The data can be listed by individual zone. Figure 46 shows 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.         */
/*
```

Figure 46. Sample Job for Listing All Data in One Zone

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 job in Figure 47 will list 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.        */
/*
```

Figure 47. Sample Job for Listing All Data in the Global Zone

If you do not require a listing of the SYSMOD and MCS entries you can use the LIST operands that allow you to list only specific entry types. See "Listing by Specific Entry Type" on page 123 for additional information.

SMP/E also provides support to list all the entries for all the zones defined in the GLOBALZONE entry. This allows you to display all data for all systems with one SMP/E command. Figure 48 shows 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.           */
/*
```

Figure 48. Sample Job for Listing All Data in All Zones

Notes:

1. The ALLZONES operand should be used with caution, as a significant amount of output may be produced.
2. This function can also be qualified by other LIST operands to limit the entries listed from each zone. For example, the job in Figure 49 on page 123 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.           */
/*
```

Figure 49. Sample Job for Listing All SYSMODs in 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 that exist in the global zone so that you do not duplicate an entry that already exists.

SMP/E supports various operands on the LIST command to allow you to list all the entries for one or more entry types. The example in Figure 50 shows the use of the DDDEF, OPTIONS, and UTILITY 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.     */
         .                  /*                          */
/*
```

Figure 50. Sample Job for Listing All Entries of One Type

See the chapter on the LIST command in *SMP/E Reference* for a complete list of all the operands corresponding to the various entry types.

Note: Not all entry types are valid for each zone type. For example, requesting a listing of the OPTIONS entries in a target zone will result in an error, as OPTIONS entries exist only in the global zone.

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 may be done is the SYSMOD entries. There are numerous operands to qualify SYSMOD entries. The chapter on the LIST command in *SMP/E Reference* describes all these operands in detail.

One of the most common uses of this function is to determine the functions (or FMIDs) that have been installed. The job in Figure 51 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.        */
         .                  /*                          */
/*
```

Figure 51. Sample Job for Listing All SYSMODs As Qualified by LIST Operands

Listing Specific Entries

When you encounter a problem in your system and contact the IBM Support Center to debug 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, this information can be obtained from that listing. This information can also be obtained through the SMP/E dialogs, as you are talking to the IBM Support Center.

SMP/E also provides additional LIST functions that allow you 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 job in Figure 52 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.         */
         .                  /*                          */
/*
```

Figure 52. Sample Job for Listing Selected Entries

You will get a listing of the required information for the two modules. If SYSMOD UR12345 was installed, it should be listed; otherwise, you will get a message indicating 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. Figure 53 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
            )
           .
/*
```

Figure 53. Sample Job for Listing Selected PTF Cover Letters

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. The FORFMID operand can be used in conjunction with the various entry-type operands to limit the entries processed. Figure 54 shows an example of listing:

1. All the entries associated with function HXY1100.
2. All the MAC entries associated with function HXZ2100.
3. 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.            */
         .                  /*                          */
/*
```

Figure 54. Sample Job for Listing by FMID

Note: The names within the FORFMID operand can be names of FMIDSET entries, in which case SMP/E will list all the entries associated with any of the FMIDs defined in the FMIDSET entry.

Listing to Compare Two Zones

If you have multiple zones there may be times when you want to determine the functional and service differences between those zones. 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. See Chapter 15 for more information.

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.

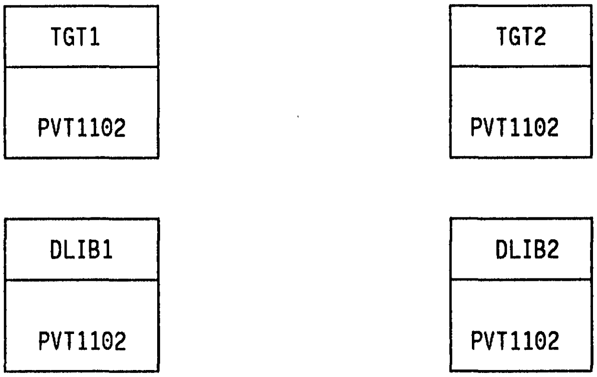
This example compares two target zones, TGT1 and TGT2. The commands in Figure 55 will perform the comparison for you.

```
//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.         */
         .                  /*                          */
/*
```

Figure 55. Sample Job for Comparing Two Target Zones

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 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 could use the commands in Figure 56.

```
//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. */
```

Figure 56. Sample Job for Comparing Two Distribution Zones

Similarly, to compare the service records for the target zone copies of PVT1102, you would use LIST with the NOAPPLY operand.

Summary

The LIST command allows you to:

- Compare two target zones, using the NOAPPLY operand, as in Figure 55 on page 126.
- Compare two distribution zones, using the NOACCEPT operand in place of the NOAPPLY operand, as in Figure 56 on page 127.
- 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 zones 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 11. Changing the Data SMP/E Manages: The UCLIN Command

This chapter discusses these topics:

- An introduction to the UCLIN command
- When to use UCLIN
- How to use UCLIN.

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. Subsequently, you may need to modify this 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. The chapters on the UCLIN command and SMP/E data-set entries in *SMP/E Reference* provide that level of detail. Read those chapters before trying to run any UCLIN commands. The following sections describe, at a very high level, what UCLIN is.

When to Use UCLIN

The ability, using UCLIN, to modify almost all the data in the SMP/E database is a very powerful function that must be used with extreme caution. When you are modifying an entry, SMP/E makes sure the data within that one entry is consistent (that is, 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 will occur when you attempt to use that OPTIONS entry. This is a very simple example of inconsistent data across entries that will not result in a serious error.

However, UCLIN modifications to other entries (such as element, LMOD, SYSMOD) may not be detected as error conditions during processing, but incorrect processing will occur (such as failing to link a module, updating the wrong library, or installing a SYSMOD that should not be installed).

In general, you should consider the following before making the UCLIN update:

1. Determine if there is a better method that can be used to obtain the same result. Table 9 shows where to find more information about alternatives to UCLIN.

Table 9. 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	*SMP/E Reference*: Chapter on the ZONEEDIT command
Add, rename, or delete a zone	*SMP/E Reference*: Chapters on the following commands: • **Adding a zone:** To add a zone, you may use the following commands, depending on the particular situation: ZONECOPY ZONERENAME ZONEEXPORT, UCLIN for the ZONEINDEX, and ZONEIMPORT • **Renaming a zone:** To rename a zone, you must use the following command: ZONERENAME • **Deleting a zone:** To delete a zone, you may use the following commands, depending on the particular situation: ZONEDELETE ZONEXPORT
Change the structure of your system (For example, to add, delete, move, or rename elements or their aliases)	*SMP/E: Program Packaging Guide*: 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 doing 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. Once you have done all of the above, if you have been given directions for installing the UCLIN (for example, within 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 IEHPROGM and IMASPZAP utilities are used to update the system libraries. That is, it allows you to:

- Add new information
- Delete existing information
- Replace existing information with new information.

Not all functions are supported for all entries or all data sets. Table 10 shows the UCL statements that may be used for each data set.

Table 10. UCL Statements for SMP/E Data Sets

Data Set	ADD	DEL	REP
SMPCSI	Yes	Yes	Yes
SMPMTS		Yes	
SMPSCDS		Yes	
SMPSTS		Yes	

The chapters on the UCLIN command and SMP/E data set entries in *SMP/E Reference* contain more detailed definitions of which entry for each data set is supported, 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(xxxxxxx).       /* 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
indicates the action to be taken on the entry or operands specified.

In general, ADD indicates 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. See the chapter on the UCLIN command in *SMP/E Reference* for a more detailed description of the functions provided by ADD, REP, and DEL.

type(name)
indicates the entry type (such as MOD, MAC, SRC, data element type, SYSMOD) and name of the entry (such as GIMMPDRV, HELP, MYSRCMOD, MYCLIST, UR12345).

operand
indicates the individual data items within 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 12. 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:

- An introduction to the REPORT CROSSZONE command
- How to define a ZONESET
- 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 may be specified on a conditional requisite (++IF) modification control statement in the PTFs. During APPLY, ACCEPT, and RESTORE processing, SMP/E automatically checks the requisites if the products are in the same zone. However, if the products are in separate zones, the requisites are not automatically checked in any other zones.

To make sure these requisites are installed where they are needed, you must:

1. Define a set of zones to be checked for conditional requisites. This is done with the ZONESET entry in the global zone.
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.

Note: If you just want to check service levels for products, you should use the REPORT SYSMODS command or the LIST command. See Chapter 15 and "Listing to Compare Two Zones" on page 126 for more information.

Define a ZONESET

To tell SMP/E which zones it should check for missing requisites, you must define a global zone ZONESET entry. You may have one or more ZONESETs to describe groups of products that might have dependencies on each other.

For example, assume you have a system that supports MVS/370 and MVS/ESA. You might have one zone, MVS370, for the base MVS/370 function, and another zone, PROD370, for a dependent function. Likewise, you might have a zone MVSESA for the base MVS/ESA function, and another zone, PRODESA, for a dependent function. The dependent functions are different versions of the same product, and they must be synchronized with each other and with their base functions. You could set up two ZONESETs to help keep these products at the same service level.

These are the commands you could use to define the ZONESETs:

```
//UCL      JOB 'accounting info',MSGLEVEL=(1,1)
//UCL      EXEC SMPPROC
//SMPCNTL  DD  *
SET      BDY(GLOBAL).     /* Set to global zone.  */
UCLIN    .                /*                      */
ADD      ZONESET(S370)    /* Create ZONESET S370. */
         ZONE(MVS370,     /* Include these zones. */
              PROD370,    /*                      */
              PRODESA)    /*                      */
         .
ADD      ZONESET(SESA)    /* Create ZONESET SESA. */
         ZONE(MVSESA,     /* Include these zones. */
              PRODESA,    /*                      */
              PROD370)    /*                      */
         .
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.
- The same zone may be part of more than one ZONESET.
- A ZONESET can contain both target and distribution zones.

See *SMP/E Reference* for more information on defining the ZONESET entry.

Run the REPORT CROSSZONE Command

After you have defined the appropriate ZONESET, you can run the REPORT CROSSZONE command to get a list of all the SYSMODs that must be installed and which zones in the ZONESET need them. This list is the Cross-Zone Requisite SYSMOD report, which also indicates 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 in a ZONESET are of the same type, SMP/E determines the type of report to be generated. For example, a ZONESET that contains only target zones results in a Cross-Zone Requisite SYSMOD report for APPLY processing. On the other hand, your ZONESET may be mixed (that is, the ZONESET may contain both target and distribution zones). If this is the case, you need to specify which type of Cross-Zone report you want to be 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 those SYSMODs needed in specific zones in the ZONESET
- FORFMID: to list only those SYSMODs needed for specific FMIDs in the ZONESET zones
- DLIBZONE or TARGETZONE: to tell SMP/E which zones you want a report on (if the zones contained in the zone set specified by ZONESET are mixed).

  Note: DLIBZONE and TARGETZONE are mutually exclusive operands.

See *SMP/E Reference* for more information on the REPORT CROSSZONE command.

To continue the example for this chapter, assume you applied a number of SYSMODs to the zones in ZONESET S370. Some of these SYSMODs named conditional requisites. To find out which zones are affected by these requisites, you could 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(S370).    /* ZONESET S370.        */
/*
```

Install the SYSMODs

Once 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.
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.
3. Rerun the REPORT command using the same ZONESET 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, you would follow these steps for ZONESET S370 and ZONESET SESA, since both contain zone PRODESA. You would continue to run the REPORT CROSSZONE command and install SYSMODs until all the needed SYSMODs were installed in both ZONESETs.

Chapter 13. Identifying Installed SYSMODs Affected by Error Holds: The REPORT ERRSYSMODS Command

This chapter contains information about using the REPORT ERRSYSMODS command to check for installed SYSMODs affected by error holds that were subsequently received. It discusses these topics:

- An introduction to the REPORT ERRSYSMODS command
- How to run the REPORT ERRSYSMODS command
- How to install the SYSMODs using the output from the REPORT ERRSYSMODS command.

Introduction

In the course of maintaining your system, you may apply or accept service and later receive HOLDDATA that affects that service. If any of that HOLDDATA was for an error reason ID, you would want to 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 subsequently 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.

Run 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 indicates if any resolving SYSMODs have been received for these holds and if 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 Release 5, or later, on or after the specified date
- ENDDATE: to check HOLDDATA entries for error reason IDs that were received by SMP/E Release 5, or later, on or before the specified date
- FORFMID: to list only those SYSMODs owned by specific FMIDs.

See *SMP/E Reference* for more information on the REPORT ERRSYSMODS command.

For example, assume that you have just received some HOLDDATA, and you need to know if 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(05 01 91) /* for HOLDDATA received   */
       ENDDATE(06 01 91).  /* between these dates     */
```

Install 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 update the selection list so that the held resolving SYSMODs will be processed.
3. Use the SMPPUNCH output to install the resolving SYSMODs in DZONE1.

Chapter 14. 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:

- An introduction to the REPORT SOURCEID command
- 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 could 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.

Run the REPORT SOURCEID Command

You can use the REPORT SOURCEID command to get a list of all the source IDs that are 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 details on the REPORT SOURCEID command, see *SMP/E Reference*.

For example, 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.
```

List 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.

Chapter 15. 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:

- An introduction to the REPORT SYSMODS command
- 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, you may have installed the same products in different zones and you 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 which SYSMODs that are installed in the first zone are applicable to, but not yet installed in, the second zone.
2. Install the applicable SYSMODs in the second zone.

Run 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 indicates which of these SYSMODs are applicable to the second zone, shows if the SYSMODs have been received, and indicates 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 details on the REPORT SYSMODS command, see *SMP/E Reference*.

For example, assume you have two MVS/ESA systems. The target zones that control these systems are MVSESA1 and MVSESA2, and they are serviced from the same global zone. You wish to determine which SYSMODs are installed in MVSESA1 and are not installed in, but are applicable to, MVSESA2. You can use the following commands:

```
SET    BDY(GLOBAL).          /* process global zone    */
REPORT SYSMODS               /* report on sysmods      */
       INZONE(MVSESA1)       /* input zone MVSESA1     */
       COMPAREDTO(MVSESA2).  /* comparison zone MVSESA2 */
```

Install 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:

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 wish 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 and run the commands to install the desired SYSMODs.

Chapter 16. Building a User Modification

This chapter discusses steps and considerations for building a USERMOD. It provides the following information:

- How to choose between building a SYSMOD as a user modification and building it as a function
- How to create modification control statements
- Examples of USERMOD type SYSMODs.

Choose between a USERMOD and a Function SYSMOD

Software products available from IBM provide you with many functions. However, there are times when these functions may not exactly meet your processing requirements. Many of these products provide interfaces such as user exit routines or dummy modules that let you customize the functions to your needs. Sometimes, however, you may need to make significant changes to a function. 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 11 on page 144 lists considerations that will help you make this decision.

Table 11. 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 will report on changes attempted by PTFs, APARs, or USERMODs.	SMP/E will not report on changes attempted by PTFs, APARs, or USERMODs.
Other SYSMODs for the same function could update the elements.	Since 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.

Create the MCSs

This section describes some of the considerations for building the MCSs for a USERMOD SYSMOD. See *SMP/E Program Packaging Guide* for more information on packaging rules and *SMP/E Reference* for more information on MCS syntax.

The ++USERMOD MCS

The ++USERMOD statement identifies this SYSMOD as a user modification and assigns a 7-character identifier to the SYSMOD.

The format of the ++USERMOD statement is:

```
++USERMOD(xxxxxxx).         /*                          */
```

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 ddddddd /* Other USERMODs incorp-   */
            ddddddd ddddddd /* 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 value that is different from that specified in the ++VER statement, that element will not be selected for installation during APPLY or ACCEPT processing, and message GIM45401 will be issued. This condition is a SYSMOD construction error. SMP/E supports a SYSMOD construction that assumes that this condition will occur regularly, and that there will be an ++IF statement in the SYSMOD that indicates another SYSMOD that will supply the proper functional version of the element.

Note: As indicated earlier, it is a good idea to construct a user modification such that each SYSMOD contains only one element. This construction technique will eliminate 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 that it is a prerequisite of the USERMOD 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.
2. If the element entry in your target zone has any UMID values, you should first check to make sure the user modification itself was constructed so it will work 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 not done, SMP/E will issue 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.
3. If you do not specify the requisites that previously replaced an element, SMP/E will not allow your USERMODs to be installed unless BYPASS(ID) is specified on the APPLY or ACCEPT commands.

4. If you want this SYSMOD to be 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 will 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 will allow the SYSMOD to be installed but will issue a warning message for each requisite that has not been specified in the PRE list.
   - What this means is that 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 user modification, 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 user modifications, designed to provide some user function. Since 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 indicates to 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 modules or change the link-edit characteristics or link-edit control cards for existing 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 to be used to perform the installation.

The general format of the ++JCLIN statement is:

```
++JCLIN   .                 /* Installation data        */
```

++MOD and ++ZAP MCSs

The ++MOD and ++ZAP statements are used to indicate a distribution library module replacement and update.

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 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 will determine all the load modules into which this distribution module was link-edited and then call 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 indicate a distribution library macro replacement and update.

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 had 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 is done to allow 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 indicate a distribution library source replacement and update.

The format of the MCSs and the data format and restrictions are similar to those for macros.

Data Element MCSs

Data element MCSs are used to define replacements for data elements, which are elements other than macros, modules, 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 chapter on MCSs in *SMP/E Reference*.) If you have installed a product using SMP/E Release 6, 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
...
```

In order 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 will be changed back to its original format. For more information about using GIMDTS, see *SMP/E Reference*.

Examples of USERMODs

These examples of user modifications show you how to do the following:

- Update a module
- Replace a module
- Add a new module
- Replace a macro or source
- Update a macro or source.

Example 1: USERMOD to Update a Module

The following is an example of a user modification to update a module. In this example, the module to be updated is GIMMPDFT (the module providing dynamic allocation support for SYSOUT data sets).

```
++USERMOD(MY00001).        /* My USERMOD              */
++VER(Z038)                /* MVS                     */
     FMID(HMP1600)         /* SMP/E function ID       */
     .                     /*                         */
++ZAP(GIMMPDFT)            /* SMP/E module name       */
     DISTLIB(AOS12)        /* DLIB ddname             */
     .                     /*                         */
  NAME GIMMPDFT
* Verify existing ddname
  VER  0050  4040404040404000
* Verify existing SYSOUT class
  VER  0057  40
* Verify existing terminal assignment
  VER  0058  00
* Rep with new data
*            M Y P R I N T  = ddname
  REP  0050  D4E8D7D9C9D5E3
*            A              = SYSOUT class
  REP  0057  C1
*            00             = do not assign to terminal
  REP  0058  00
```

Notes:

1. You should verify each location that you are going to update.
2. Only one name is specified in the NAME statement.
3. The changes made by the REP statements are explained by the preceding comment lines.

Example 2: USERMOD to Replace a Module

The following is an example of a user modification to replace a module. In this example, the module to be replaced is GIMMPUXD (the driver for the SMP/E installation exit routines). Assume that you have written and assembled your version of GIMMPUXD and that you now wish to create a user modification to install it.

```
++USERMOD(MY00002).          /* My USERMOD            */
++VER(Z038)                  /* MVS                   */
     FMID(HMP1600)           /* SMP/E function ID     */
     .                       /*                       */
++MOD(GIMMPUXD)              /* SMP/E module name     */
    DISTLIB(AOS12)           /* DLIB ddname           */
     .                       /*                       */
...
... Object module for your version of GIMMPUXD
...
```

Note: There are no PRE operands on the ++VER, because this module has not been modified since its initial installation; thus the FMID and RMID subentries in the MOD entry are equal, and only the ++VER FMID operand is required.

Example 3: USERMOD to Add a New Module

The following is an example of a user modification to add a new module. For this example, assume:

- You have written three new modules, SMPMOD01, SMPMOD02, and SMPMOD03, that are link-edited to create a new load module, named SMPEXT01. This load module is an installation exit routine that gets called by GIMMPUXD.
- SMPMOD01 is the entry point into the new load module.
- The new load module is to exist in SYS1.LINKLIB.

You now want to install this new module using SMP/E.

```
++USERMOD(MY00003).         /* My USERMOD               */
++VER(Z038)                 /* MVS                      */
     FMID(HMP1600)          /* SMP/E function ID        */
     PRE(MY00002)           /* Need GIMMPUXD installed
                               for exit to get called.  */
     .                      /*                          */
++JCLIN.                    /* JCLIN to install exit    */
//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)             /* My SMP/E exit routine    */
     DISTLIB(PVTDLIB1)      /* in my DLIB               */
     .                      /*                          */
...
... Object module for SMPMOD01
...
++MOD(SMPMOD02)             /* New SMP/E exit routine   */
     DISTLIB(PVTDLIB1)      /* in my DLIB               */
     .                      /*                          */
...
... Object module for SMPMOD02
...
++MOD(SMPMOD03)             /* New SMP/E exit routine   */
     DISTLIB(PVTDLIB1)      /* in my DLIB               */
     .                      /*                          */
...
... Object module for SMPMOD03
...
```

Notes:

1. This SYSMOD has SYSMOD MY00002 as a prerequisite because it is not called until the new version of GIMMPUXD is installed.
2. The ++JCLIN statement was required because these are new modules and a new load module.
3. This SYSMOD could have been packaged as a ++FUNCTION because each of the modules is user-written.
4. This user modification could have been packaged as three separate SYSMODs with one module in each. Then they would have had the ++VER REQ operand specified so that they could be installed as a set.

Example 4: USERMOD to Replace a Macro or Source

The following is an example of a user modification to add a new macro. For this example, assume:

- The macro is to be put into both a target and distribution library.
- Source module, USRSRC01, is to be assembled when the macro is installed.

```
++USERMOD(MY00004).         /* My USERMOD              */
++VER(Z038)                 /* MVS                     */
     FMID(HJE2102)          /* JES function ID         */
     .                      /*                         */
++MAC(USRMAC01)             /* My user macro           */
     DISTLIB(USRMACLB)      /* in my DLIB              */
     SYSLIB (MACLIB)        /* but in system MACLIB    */
     ASSEM  (USRSRC01)      /* Reassemble this source. */
     .                      /*                         */
...
... Macro replacement
...
```

Notes:

1. No JCLIN is required to install a new macro; all necessary information can be specified on the ++MAC statement.
2. This was an example for a macro replacement but could be equally valid for a source module replacement, with the following exceptions:
   a. The ASSEM operand is not required for a source module. After replacing the source, SMP/E automatically attempts to assemble the new version of the source module, assuming that SMP/E understands how the assembled module is to be installed into the system.
   b. The ++JCLIN statement should be used to define to SMP/E how the assembled module installs. The data should be similar to that ir the example for adding a new module.

Example 5: USERMOD to Update a Macro or Source

If, at some later date, this macro had to be updated, you could create the following user modification:

```
++USERMOD(MY00005).         /* My USERMOD                */
++VER(Z038)                 /* MVS                       */
     FMID(HJE2102)          /* JES function id           */
     PRE (MY00004)          /* Last replacement          */
     .                      /*                           */
++MACUPD(USRMAC01)          /* My user macro update      */
     DISTLIB(USRMACLB)      /* in my DLIB                */
                            /* SYSLIB remembered by SMP */
     ASSEM(USRSRC01)        /* Reassemble this source.  */
     .                      /*                           */
./ CHANGE NAME=USRMAC01
...
... macro changed lines here
... with sequence numbers in
... columns 73 thru 80
...
```

Notes:

1. The ++VER PRE operand was used to specify that the update was for the previous replacement level of the macro.
2. This example could be equally applicable to a source update. Again, the ASSEM operand is not supported, and since you have already defined how the assembled module installs (when you first added the source code), there is no need for a ++JCLIN statement here.

Chapter 17. Processing Inline JCLIN at ACCEPT Time

This chapter provides you with information on using inline JCLIN to update the distribution zone during ACCEPT processing. It discusses:

- What is meant by inline JCLIN
- The benefit of processing inline JCLIN at ACCEPT time
- Restrictions to remember.

What Is Meant by Inline JCLIN

JCLIN input can be either **inline** or **out-of-stream**. Inline JCLIN is identified by a ++JCLIN MCS. Out-of-stream JCLIN is in the data stream identified by the SMPJCLIN DD statement. A SYSMOD does not have to be packaged in a particular way to contain inline JCLIN. SYSMODs packaged as relative files, inline, and in indirect libraries can all have inline JCLIN.

Reasons for Processing Inline JCLIN at ACCEPT Time

Saving inline JCLIN for products not included by a system or subsystem generation procedure makes building a new system or subsystem easier. When SMP/E processes the JCLIN before it processes the elements, the distribution zone is initialized with JCLIN information for products not included by a system or subsystem generation procedure. Later, when SMP/E builds a system or subsystem using the existing distribution libraries and the GENERATE command, this JCLIN data is copied into the target zone. This eliminates the need for a separate step to obtain JCLIN information for products not included by a system or subsystem generation procedure.

Restrictions

Remember the following restrictions when you plan to save inline JCLIN at ACCEPT time:

- You can save inline JCLIN only for products whose distribution libraries were initially built using SMP/E Release 3, or higher. You cannot save inline JCLIN for products that were installed with an earlier release of SMP/E.
- Before you begin building the distribution libraries, set the ACCJCLIN indicator in the associated DLIBZONE entry. This tells SMP/E to save inline JCLIN in the distribution zone.

  If ACCJCLIN is not set when you initially build the distribution libraries, it cannot be set until the next time the libraries are built.

  Note: With MVS CBIPO, it is by default that the delivered system is set up using ACCJCLIN.
- After you install a product using SMP/E Release 3, or higher, and the ACCJCLIN indicator, you must keep ACCJCLIN set in the DLIBZONE entry. This makes sure that any time you accept service for that product, its JCLIN is updated in the distribution zone.

- The only way to save inline JCLIN in the distribution zone is through the ACCEPT command. The JCLIN command does not update the distribution zone.
- Saving JCLIN at ACCEPT does **not** take the place of a stage 1 system generation for products that **are** included in a system or subsystem generation procedure.
- Because additional data is being saved in the distribution zone, the SMPCSI data set that contains the distribution zone may require more DASD space.

Inline JCLIN is not processed for superseded or deleted SYSMODs.

The NOJCLIN operand on the ACCEPT command prevents the processing of inline JCLIN. NOJCLIN can be used if the JCLIN contains data that would overlay user-modified entries in the distribution zone.

If you specify NOJCLIN without an operand list, inline JCLIN is not processed for any SYSMOD that was selected and that contained a ++JCLIN statement. If you specify NOJCLIN with an operand list, inline JCLIN is not processed for the specified SYSMODs.

For more information about JCLIN processing, see *SMP/E Reference*.

Chapter 18. Determining Which SYSMODs Led Others to Fail: The Causer SYSMOD Summary Report

Change for Root Cause SPE

The causer SYSMOD information is issued only when the root cause SPE (PTF UR90259 for FMID HMP1600 and, if the Japanese feature is installed, PTF UR90260 for FMID JMP1611) is installed.

This chapter 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

To reduce the work needed to determine which errors caused SYSMODs to fail, SMP/E performs root cause analysis for the ACCEPT, APPLY, and RESTORE commands to identify causer SYSMODs. A causer SYSMOD is a SYSMOD whose failure 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: x37 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 checking the SYSMOD Status report to determine the results of processing, you use the Causer SYSMOD Summary report to determine what 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 indicate that several of the SYSMODs failed. You want to resolve all the indicated errors and install all the SYSMODs in the CBPDO.

1. Go directly to the Causer SYSMOD Summary report to identify the causer SYSMODs and determine how to resolve the errors indicated in the report.
2. Resolve the indicated errors.
3. Rerun the command that failed. If you find more errors on this pass, repeat the process for these errors.

Resolving Errors for a Single SYSMOD that Failed

Suppose you are installing a group of SYSMODs, but the reports indicate 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 directly to the Causer SYSMOD Summary report, and determine how to resolve the errors for the causer SYSMOD.
3. Resolve the indicated errors.
4. Rerun the command to install the SYSMOD that previously failed. If you find more errors on this pass, repeat the process for these errors.

Example

Suppose you ran the following command:

```
APPLY  S(UR00001) GEXT CHECK.
```

and the SYSMOD Status report included the results shown in Figure 57.

```
Page nnnn  - NOW SET TO TARGET ZONE nnnnnnn  DATE mm/dd/yy  TIME hh:mm:ss GIMSMP LVL 16.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 57. 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 58.

```
Page nnnn  - NOW SET TO TARGET ZONE nnnnnnnn  DATE mm/dd/yy  TIME hh:mm:ss GIMSMP LVL 16.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 58. Causer SYSMOD Summary Report: Sample Report for APPLY

That report indicates that APPLY processing failed because error hold AR00003 was not resolved for SYSMOD UR00002. By resolving this hold, you will fix the errors indicated in the SYSMOD Status report.

Table 12 on page 160 shows the steps involved in figuring out which SYSMOD led others to fail. The first column outlines the new method you can follow, using causer SYSMOD information, and the second column outlines the old method you used to have to follow before causer SYSMOD information was available.

This example, being fairly simple, may not make it immediately obvious just how much time the causer SYSMOD information can save you. Try to imagine using the steps in the *Before Causer SYSMOD Information* column to figure out the causer for numerous other SYSMODs by analyzing all their related error messages. The time saved by using the causer SYSMOD information could be considerable.

Table 12. Figuring Out the Causer SYSMOD (With and Without the Causer SYSMOD Information)

With Causer SYSMOD Information	**Before Causer SYSMOD Information**
1. Do one of the following: • Check the SYSMOD Status report for the causers; then look them up in the Causer SYSMOD Summary report. • Go directly to the Causer SYSMOD Summary report to determine the causers. (UR00002 is the causer SYSMOD in the example.)	1. Check the SYSMOD Status report. In this example, UR00001 and UR00002 have a status of HELD, and AR00001 has a status of NOGO. • APPLY processing failed for UR00001 because error hold AR00001 was not resolved. • AR00001 should have been superseded by UR00002, but UR00002 had an unresolved error hold. • APPLY processing failed for UR00002 because error hold AR00003 was not resolved.
2. Read the error description in the Causer SYSMOD Summary report to determine the problem. The report tells us that error hold AR00003 for UR00002 was not resolved.	2. Look up the error messages in the SMPOUT output. There are one severe message, two error messages, and three informational messages: • GIM30206E—APPLY processing failed for SYSMOD UR00001. • GIM35901I—error hold AR00001 was not resolved. • GIM302096—APPLY processing failed for SYSMOD UR00002. • GIM35901I—error hold AR00003 was not resolved. • GIM24801S—no SYSMODs satisfied the APPLY command. • GIM20501I—APPLY processing is complete. SYSMOD UR00001 failed because error hold AR00001 was not resolved. SYSMOD UR00002 would have superseded AR00001, but UR00002 failed because error hold AR00003 was not resolved. So we need to resolve AR00003.
3. The report summarizes the error messages and the possible causes for a SYSMOD failure. In most cases, you should not need to look up additional information in the messages manual and the SMPOUT output.	3. Look up GIM35901 to find out how to resolve error hold AR00003. • If you want to resolve the reason ID, determine if AR00003 or a superseding SYSMOD is available to be received. If it is not, you need to get a copy from IBM. • If you do not want to resolve the reason ID, you need to bypass it.
4. Take the appropriate action: • To resolve the reason ID, receive AR00003 or a superseding SYSMOD, and rerun the APPLY command. • To bypass the reason ID, specify `BYPASS(HOLDERR(AR00003))` on the APPLY command, and rerun the command.	4. Take the appropriate action: • To resolve the reason ID, receive AR00003 or a superseding SYSMOD, and rerun the APPLY command. • To bypass the reason ID, specify `BYPASS(HOLDERR(AR00003))` on the APPLY command, and rerun the command.

Part 3. Appendixes

Appendix A. SYSMOD Installation Checklists

This appendix contains a checklist for the installation of each type of SYSMOD:

- New function (page 164) as described in Chapter 4
- Preventive service (page 166) as described in Chapter 5
- Corrective service (page 168) as described in Chapter 6
- User modification (page 169) as described in Chapter 7.

This appendix also contains a checklist (on page 170) for rebuilding target zones with SYSGEN- and non-SYSGEN-supported products. You may want to use the lists as a guide during the installation of any of these SYSMOD types. The list should be filed in a place accessible to all the system programmers in your installation, so that if they have any questions as to the status of that installation, the information is available.

The SMP/E dialogs provide a similar service through the SYSMOD Management dialog, which leads you step by step through the installation procedures.

Permission is granted to reproduce the blank checklists in this appendix.

New Function Installation Checklist

Table 13 (Page 1 of 2). New Function Installation Checklist

Step	Date Completed	By Whom	Description	Your Notes
1.00			PREPARE YOUR SYSTEM.	
1.01			Read the program directory.	
1.02			Compress libraries.	
1.03			Allocate libraries.	
1.04			Check SMP/E data-set free space.	
1.05			Back up volumes.	
Note: If you are installing preventive service along with a new function and its CUM tape, see Step 1.05 in Table 14 on page 166.				
1.06			Get documentation.	
1.07			Estimate the time required.	
2.00			STAGE THE SYSMODS.	
2.01			Receive the function.	
2.02			Receive CUM tape.	
Note: If you are installing preventive service along with a new function and its CUM tape, see Steps 2.01 and 2.02 in Table 14 on page 166.				
3.00			UPDATE TARGET LIBS.	
3.01			Obtain the latest exception SYSMOD data from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.	
3.02			Run APPLY CHECK.	
3.03			Research the APPLY CHECK reports.	
3.04			Obtain additional SYSMODs if required.	
3.05			Run the APPLY job.	
Note: If you are installing preventive service along with a new function and its CUM tape, see Steps 3.06 and 3.07 in Table 14 on page 166.				
4.00			TEST THE SYSTEM.	
4.01			Back up volumes.	
4.02			IPL the system.	
4.03			Run the function IVP.	
4.04			Run your IVP.	
5.00			UPDATE DLIBS.	
5.01			Obtain latest exception SYSMOD data from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.	
5.02			Run ACCEPT CHECK.	

Table 13 (Page 2 of 2). New Function Installation Checklist

Step	Date Completed	By Whom	Description	Your Notes
5.03			Research the ACCEPT CHECK reports.	
5.04			Obtain additional SYSMODs if required.	
5.05			Run the ACCEPT job.	

Note: If you are installing preventive service along with a new function and its CUM tape, see Step 5.06 in Table 14 on page 166.

Preventive Service Installation Checklist

Table 14 (Page 1 of 2). Preventive Service Installation Checklist

Step	Date Completed	By Whom	Description	Your Notes
1.00			PREPARE YOUR SYSTEM.	
1.01			Compress libraries.	
1.02			Check SMP/E data-set free space.	
1.03			Bring DLIBs to the proper level.	
1.04			Back up volumes.	
1.05			Obtain the latest exception SYSMOD data from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.	
1.06			Get documentation.	
1.07			Estimate time required.	
2.00			STAGE THE SYSMODS.	
2.01			Receive PTFs and exception SYSMOD data.	
2.02			List the PTF cover letters.	
2.03			Obtain the latest exception SYSMOD data from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.	
3.00			UPDATE TARGET LIBS.	
3.01			Obtain the latest exception SYSMOD data from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.	
3.02			Run APPLY CHECK.	
3.03			Research the APPLY CHECK reports.	
3.04			Obtain additional SYSMODs if required.	
3.05			Run the APPLY job.	
3.06			Install PTFs that require special processing.	
3.07			Get any necessary UCLIN.	
4.00			TEST THE SYSTEM.	
4.01			Back up volumes.	
4.02			IPL the system.	
4.03			Run selected function IVPs.	
4.04			Run your IVP.	
5.00			UPDATE DLIBS.	

Table 14 (Page 2 of 2). Preventive Service Installation Checklist

Step	Date Completed	By Whom	Description	Your Notes
5.01			Obtain the latest exception SYSMOD data from the IBM Support Center, CSSF Information/Access, or SoftwareXcel Extended.	
5.02			Run ACCEPT CHECK.	
5.03			Research the ACCEPT CHECK reports.	
5.04			Obtain additional SYSMODs if required.	
5.05			Run the ACCEPT job.	
5.06			Install PTFs that require special processing.	

Corrective Service Installation Checklist

Table 15. Corrective Service Installation Checklist

Step	Date Completed	By Whom	Description	Your Notes
1.00			BUILD OR CHECK SYSMOD.	
1.01			Determine problem to the module level.	
1.02a			If possible, obtain fixing PTF or APAR from IBM. For APARs, check the accuracy of the fix.	
1.02b			Construct the SYSMOD.	
2.00			PREPARE YOUR SYSTEM (as required for this SYSMOD).	
3.00			STAGE THE SYSMODS.	
3.01			Receive corrective service.	
4.00			UPDATE TARGET LIBS.	
4.01			Run APPLY CHECK.	
4.02			Research the APPLY CHECK reports.	
4.03			Obtain additional SYSMODs if required.	
4.04			Run the APPLY job.	
5.00			TEST THE SYSTEM (test according to the original problem).	
6.00			UPDATE DLIBS.	
6.01			Evaluate DLIB update.	
6.02			Run ACCEPT CHECK.	
6.03			Research the ACCEPT CHECK reports.	
6.04			Obtain additional SYSMODs if required.	
6.05			Run the ACCEPT job.	

User Modification Installation Checklist

Table 16. User Modification Installation Checklist

Step	Date Completed	By Whom	Description	Your Notes
1.00			PREPARE YOUR SYSTEM (based on complexity and individual requirements).	
2.00			STAGE THE SYSMODS (receive USERMOD).	
3.00			UPDATE TARGET LIBS.	
3.01			Run APPLY CHECK.	
3.02			Research the APPLY CHECK reports.	
3.03			Run the APPLY job.	
4.00			TEST THE SYSTEM.	
4.01			Test based on user modification requirements.	
4.02			Document the function.	
5.00			UPDATE DLIBS.	
5.01			Evaluate DLIB update.	
5.02			Run ACCEPT CHECK.	
5.03			Research the ACCEPT CHECK reports.	
5.04			Run the ACCEPT job.	

Rebuild Checklist: Target Zone with SYSGEN- and Non-SYSGEN-Supported Products

Table 17. Rebuild Checklist: Target Zone with SMP/E GENERATE Command

Step	Date Completed	By Whom	Description	Your Notes
1.00			Make the distribution libraries used during SYSTEM match the old target zone from which you will do the GENERATE.	
2.00			Do a Stage 1 system generation to get the JCLIN for the SYSGEN-supported products.	
3.00			Use GENERATE with the FORFMID operand to specify all non-SYSGEN-supported products.	
4.00			Allocate a new target zone.	
5.00			Copy the distribution zone to the new target zone.	
6.00			Do additional initialization on the new target zone.	
7.00			Run JCLIN using the SYSGEN output as input for the JCLIN.	
8.00			Run JCLIN using the GENERATE output as input for the JCLIN. (See note.)	
9.00			Run the SYSGEN output jobs.	
10.00			Run the GENERATE output jobs.	

Note: For SYSMODs with inline JCLIN, you might be able to skip this step.

Appendix B. Abbreviations

ACDS	Alternate control data set
ACRQ	Alternate conditional requisite queue
APAR	Authorized program analysis report
CBIPO	MVS Custom-Built Installation Process Offering
CBPDO	MVS Custom-Built Product Delivery Offering
CDS	Control data set
CICS	Customer Information Control System
CORPE	PSP upgrade for PE-PTFs available correctively but not yet available on a PUT
CRQ	Conditional requisite queue
CSI	Consolidated software inventory
CSSF	Customer Software Support Facility
DASD	Direct access storage device
DD	Data definition
DLIB	Distribution library
EC	Engineering change
IMS	Information Management System
IMSGEN	IMS generation
I/O	Input or output
ISPF	Interactive System Productivity Facility
ISPF/PDF	Interactive System Productivity Facility/Program Development Facility
IVP	Installation Verification Procedure
JES	Job Entry Subsystem
MCS	Modification control statement
MVS	Multiple Virtual Storage
MVS/ESA	Multiple Virtual Storage/Enterprise Systems Architecture (MVS/SP* Version 3 and MVS/ESA SP Version 4)
MVS/SP	Multiple Virtual Storage/System Product
MVS/XA	Multiple Virtual Storage/Extended Architecture (MVS/SP Version 2)
MVS/370	Multiple Virtual Storage for System/370 (MVS/SP Version 1)
NCP	Network Control Program
OS/VS	Operating System/Virtual Storage (VS1 or VS2)
OS/VS1	Operating System/Virtual Storage 1
OS/VS2	Operating System/Virtual Storage 2
PDF	Program Development Facility
PE-PTF	Program error PTF
PSP	Preventive Service Planning
PTF	Program temporary fix
PTS	PTF temporary store
PUT	Program update tape
RACF	Resource Access Control Facility
RAS	Reliability, availability, and serviceability
RIM	Related installation material
SMP	System Modification Program
SMP4	System Modification Program Release 4
SMP/E	System Modification Program Extended
SMPCSI	System Modification Program consolidated software inventory

MVS/SP is a trademark of the IBM Corporation.

SMPMTS	SMP/E Macro Temporary Store
SMPPTS	System Modification Program PTF temporary store
SMPSCDS	SMP/E Save Control Data Set
SMPSTS	SMP/E Source Temporary Store
SYSMOD	System modification
TGTLIB	Target library
USERMOD	User modification
VSAM	Virtual Sequential Access Method
VS1	Operating System/Virtual Storage 1
VS2	Operating System/Virtual Storage 2

Index

Communicating Your Comments to IBM

System Modification Program Extended
User's Guide
Release 6

Publication No. SC28-1302-11

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States & Canada: 914+385-0662
  - Other Countries: (country code +1)+914+385-0662
- If you prefer to send comments electronically, use this network ID:
  - IBMLINK: KGNRCF at KGNVMC
  - IBM Mail Exchange: USIB27BQ at IBMMAIL
  - Internet: KGNRCF@KGNVMC.vnet.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Readers' Comments

System Modification Program Extended
User's Guide
Release 6

Publication No. SC28-1302-11

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ______________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain)

[ ] As a text (student)
[ ] As a text (instructor)

____

____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

Name	Address
Company or Organization	
Phone No.	

Readers' Comments
SC28-1302-11

IBM®

Cu
Al

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 52QA MS 911
NEIGHBORHOOD ROAD
KINGSTON NY 12401-0000

Fold and Tape | **Please do not staple** | Fold and Tape

C
A

Readers' Comments

System Modification Program Extended
User's Guide
Release 6

Publication No. SC28-1302-11

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ______________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ]	As an introduction	[ ]	As a text (student)
[ ]	As a reference manual	[ ]	As a text (instructor)
[ ]	For another purpose (explain)		

____

____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

______________________ ______________________
Name Address

______________________ ______________________
Company or Organization

______________________ ______________________
Phone No.

Readers' Comments
SC28-1302-11

Cu
Alc

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 52QA MS 911
NEIGHBORHOOD ROAD
KINGSTON NY 12401-0000

Fold and Tape **Please do not staple** Fold and Tape

C
A

Readers' Comments

System Modification Program Extended
User's Guide
Release 6

Publication No. SC28-1302-11

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ____________________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain)
[ ] As a text (student)
[ ] As a text (instructor)

____

____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

____
Name

____
Company or Organization

____
Phone No.

____
Address

____

____

Readers' Comments
SC28-1302-11

IBM®

Cu
Alc

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 52QA MS 911
NEIGHBORHOOD ROAD
KINGSTON NY 12401-0000

Fold and Tape **Please do not staple** Fold and Tape

IBM®

File Number: S370-34
Program Number: 5668-949

Printed in USA

SC28-1302-11