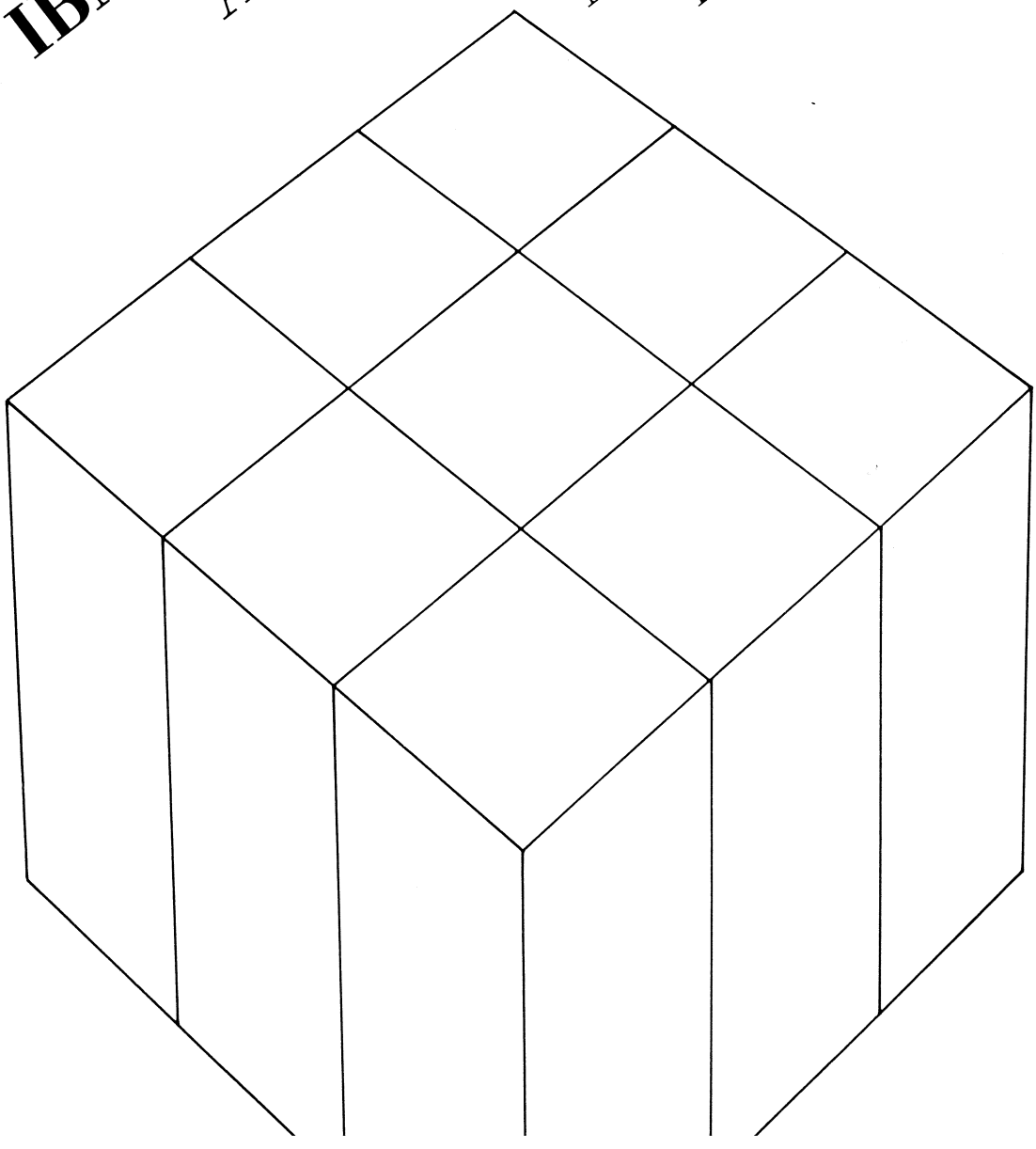
IBM Virtual Storage Extended Advanced Functions

Maintain System History Program Reference



IBM Virtual Storage Extended/ Advanced Functions

Maintain System History Program Reference

Program Number 5666-301

Order Number SC33-6199-02
File No. S370/4300-34

Third Edition (March 1987)

This edition is a major revision of SC33-6199-1. This edition applies to Version 3, Release 1 of IBM Virtual Storage Extended/Advanced Functions, Program Number 5666-301, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes or additions to the text and illustrations are indicated by a vertical bar to the left of the change. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30XX and 4300 Processors Bibliography,* GC20-0001, for the editions that are applicable and current.

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

Publications are not stocked at the addresses given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

IBM Corporation
Dept. 6R1BP
180 Kost Road
Mechanicsburg, PA 17055, USA

or to:

IBM Deutschland GmbH
Dept. 3164
Schoenaicher Strasse 220
D-7030 Boeblingen, Federal Republic of Germany

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This manual contains a complete description of all control statements of the Maintain System History Program (MSHP). Information on when and how to use these statements in the installation and service process is contained in *VSE/Advanced Functions, Planning and Installation*, SC33-6193.

The manual is intended as a reference source in cases where more detailed control information is needed than is given in the Planning and Installation manual. The manual provides the reference information needed for installing and servicing a product, and also describes the control statements needed by IBM personnel (product owners) when preparing a programming package for shipment.

The contents of this manual is divided into the following chapters:

Chapter 1, "Introduction" contains a summary of all MSHP control statements and the rules for writing these statements.

Chapter 2, "Function Control Statements" describes, in alphabetic order, all the MSHP function control statements.

Chapter 3, "Detail Control Statements" describes, also in alphabetic order, the related detail control statements.

Related Publications

VSE/Advanced Functions, Planning and Installation, SC33-6193:

> Contains, in the form of procedures, all the information needed for installing a system or product and also for installing PTFs and local or APAR fixes.

VSE/Advanced Functions, System Management Guide, SC33-6191:

> Contains a short description of the operating environment of MSHP and discusses the required logical unit assignments for VSE/Advanced Functions programs that may be invoked by MSHP.

VSE/System Package Guide for Solving Problems, SC33-6311:

> Contains information on how to manipulate and display information from the history file.

Preface

Contents

Figures

Summary of Amendments

Version 2, Release 1

This publication has the same overall organization as the MSHP Reference manual available with VSE/Advanced Functions 1.3. However, the publication had to be revised almost entirely to cover the installation and service changes implemented for VSE/Advanced Functions 2.1.

The major changes are:

- New shipment package format - The shipment package is adjusted to the new VSE Librarian. The installation process has been significantly improved.
- New service tape format - MSHP supports the new 'SYSIN' service tape format for preventive and corrective PTF service.
- Multiple-target library service - Service changes can be installed on two or more sublibraries in one service run.
- Multiple-level installation - More than one level of a program product can be installed and serviced under control of MSHP.
- Product backup - You can create a backup copy of a product, together with the related history file entries.
- Requirement checking for local or APAR fixes - The CORRECT function checks for any prerequisites needed before a fix can be applied.
- Extension of the MSHP history file.
- A PATCH function for altering phases without updating the history file.

Summary of Amendments

Chapter 1. Introduction

The control statements of the Maintain System History Program (MSHP) are of two types:

- Function control statements (summarized in Figure 1 on page 2), which are used to define to MSHP the required function.
- Detail control statements (summarized in Figure 2 on page 3), which are used to provide further details about the requested function.

To use an MSHP function, build a job or job step comprising:

- Job control ASSGN statements or commands for the necessary logical units as given in the description of the function control statement;
- The job control statement or command:

  ```
  [//] EXEC MSHP
  ```

- The applicable MSHP function control statement;
- Depending on the requested function, one or more detail control statements.

The two types of MSHP statement are described under separate headings and in alphabetical order. Each description includes:

1. The syntax notation.
2. A general description of the purpose and function of the statement, plus any special considerations and restrictions.
3. For function control statements, the system and programmer logical unit assignments required for the specific MSHP function. These assignments include work files used by MSHP or by any other system program that is invoked by MSHP (for example, assembler, librarian, or linkage editor). MSHP requires the same logical unit and extent information as the called program.
4. For function control statements, any required or optional detail control statements.
5. A detailed explanation of the statements' operands, together with any restrictions and the applicable default values.

Control Statement Overview

MSHP control statements may be entered from SYSIN or SYSLOG.

Figures 1 and 2 give an overview of the function control statements and detail control statements, respectively. In these figures, the shortest valid form of each operation code is shown in capital letters. The effect of the statements is summarized under "Purpose."

Function Control Statement	Purpose
APply	Install a PTF and record it in the system history file of the operational system.
ARChive	Enter information relating to products, components, PTFs and local or APAR fixes into the history file.
BACKup	Copy an auxiliary or system history file from disk to magnetic tape for backup purposes.
COPy	Copy a history file from disk to disk.
CORrect	Install a local or APAR fix.
CReate	Pre-format a history file and reserve space for the PERsonalize function (see below).
DUMP	Produce a formatted printout of a system or auxiliary history file.
INCorporate	Install a component distributed in SYSIN format.
INSTall SYsres/ PRoduct	Install a system (SYsres) or product (PRoduct).
INSTall SErvice/ BAckout	Apply preventative and corrective service from the service tape (SErvice) or a backout tape (BAckout).
LIST	Retrieve information from a service tape and write this information to SYSLST.
Lookup	Display on SYSLOG selected information from the system history file.
MERge	Insert entries of one history file into another history file.
PERsonalize	Identify the system history file in relation to a specific user.
REMove	Erase entries from the system history file.
RESIdence	Specify the names of the sublibraries in which a product resides.
RESTore	Restore a complete shipment package or a history file from magnetic tape to disk.
RETRace	Retrieve information from the system history file and print the information on SYSLST.

Figure 1 (Part 1 of 2). Function Control Statements - Overview and Purpose

Function Control Statement	Purpose
REVoke	Restore an operational system to the status that existed before the installation of a PTF.
SELect	Select individual tailor jobs from the generation file (for retailoring).
TAILor	Identify and initiate the generation (or re-generation) of a sublibrary member.
UNdo	Remove an initiated local or APAR fix to re-establish the previous library status.

Figure 1 (Part 2 of 2). Function Control Statements - Overview and Purpose

Detail Control Statement	Purpose
AFFects	Specify the sublibrary members that are affected by a PTF or local fix application.
ALter	Specify text modifications for sublibrary members.
COMPAtible	Indicate the products that are compatible with the shipped product.
COMPRises	Identify the component, phases, modules, and/or macros that comprise a product, and enter the information in the history file.
DATA	Delimit input to the LIBR and LNKEDT programs.
DEFine	Create label/extent definitions for the history file.
DELete	Specify the lines to be deleted from a source book when applying a local fix.
EXCLude	Exclude one or more products, components, or PTFs from a service application.
EXECute	Call one or more system programs (for example, assembler) required for tailoring.
GENerate	Specify a phase, module, or macro for regeneration.
INCLude	Include one or more products, components, or PTFs in a service application.
INFluences	Identify those generated phases, modules, or macros that are affected by a PTF or local/APAR fix and that have to be regenerated if the fix is applied.
INsert	Specify the lines to be inserted in a source book when applying a local fix.

Figure 2 (Part 1 of 2). Detail Control Statements - Overview and Purpose

Detail Control Statement	Purpose
INVolves	Explicitly request link-editing when installing a product or applying service.
OR	Delimit a set of requirements (initiated with the REQuires statement) and test the requirements.
PTF	List the PTFs whose cover letters are to be printed.
REPlace	Delimit where replacement lines for local or APAR fixes must begin and end; initiate the replacement of the source text.
REQuires	Specify the requirements for successfully installing a shipment package or applying service.
RESolves	Associate a comment with a PTF, a product, an APAR or a local fix, or a generated member.
RESTart	Indicates, for macro updates, that a new sequence number series starts after the specified statement.
SCan	Scan a phase for a specified string, or display 16 bytes of a phase.
SUPersedes	Record the PTFs that are superseded by a given PTF.
VERify	Specify where a verification is to be made for a local or APAR fix correction.

Figure 2 (Part 2 of 2). Detail Control Statements - Overview and Purpose

Rules for Writing MSHP Control Statements

- With one EXEC MSHP statement or command, any number of function control statements can be specified.
- The function control statement you use determines which detail control statement(s) must or may follow.
- Detail control statements can follow only a function control statement or another detail control statement.

  If detail control statements must be submitted in a specific sequence, this is noted in the description for the function control statement.
- MSHP control statements are of free form. The operation codes may begin in any position of the input line.
- An input line for MSHP control statements represents the first 72 characters of a card image input record, or 120 characters for console input.
- Operation codes and operand keywords may be abbreviated. In the statement descriptions, permissible minimum abbreviations are shown as uppercase character strings, followed by the remainder of the

keyword in lowercase. For example, INSTall may be coded as INST, INSTALL, or anything within these limits.

- A value contained within brackets [...] may be included or omitted, depending on the requirements of the program. Two or more values contained within brackets and separated by an | sign represent alternatives, one (and only one) of which may be chosen. For example:

  ```
  [IRRevokable|REVokable]
  ```

  If you specify nothing, MSHP assumes the *underlined* option as default.

- Options contained within braces {...} and separated by an | sign represent alternatives, one of which *must* be chosen. For example:

  ```
  {PRODuct|SYSres}
  ```

- The operands of a statement are separated from one another by:

  - One or more blanks
  - A comment (which is text within /* and */)
  - A comma (which may be surrounded by one or more blanks or comments)

- An all-comment input line is allowed. However, it must not begin in column 1.

- Words given in all lowercase letters represent information that must be supplied by the user.

- The equal sign (=), the plus sign (+), the colon (:), and the single quotes (' ') must be coded as shown; they may be surrounded by one or more blanks, except for the (+) sign, which must *not* be preceded or followed by a blank.

- An ellipsis (a series of three periods) indicates that a list of up to 100 items (such as PTF numbers) may be specified within parentheses. For example:

  ```
  (UD27484,UD13528,...)
  ```

  However, a single item does not have to be enclosed in parentheses.

- The individual values in a list can be separated from each other by:

  - One or more blanks
  - A comment (text within /* and */)
  - A comma (which may be surrounded by one or more blanks or comments)

  Commas and blanks as separators may be intermixed in a given list.

- A control statement (function or detail) ends with the end of the input line, unless it is explicitly continued by means of a dash (-), followed by at least one blank. It may also end with a semicolon.

The continuation dash must also be preceded by at least one blank, except after a

- Comma;
- Parenthesis;
- Equal sign;
- Comment;
- Quoted string.

For function control statements, not more than six continuation lines may be specified.

A pair of values connected by a colon, as in

```
APPLY 5666-301-06-H07:UD12345
```

cannot be broken by a line end; nor can a keyword itself, a number, or a string (with or without quotes) be continued on a subsequent line.

- From the console, MSHP control statements may be entered in uppercase or in lowercase.
- An MSHP statement entered from the console may be canceled by entering two question marks (??).

Coding Conventions for Frequently Used MSHP Operands

component

The term 'component' stands for the component identification number (or program number) of a component. Example:

```
5666-301-06
```

A component may occur in more than one product, in which case it is further qualified by a 'level' indication.

level

This is a string of three alphameric characters which identifies the component uniquely if, for example, the component is shared by several different licensed programs.

To indicate that a component belongs to a certain program, the level number of that program is hyphenated with the component name.

For example:

```
5666-301-06-H07
```

identifies component 5666-301-06 at level H07.

product

The term 'product' stands for the 6-character identification number of a licensed program, for example:

```
301H07
```

The first three characters (301 in this example) are the product code. This is derived from the program number. The remaining characters (H07 in this example) are the level number of the program, formerly known as feature number or release number.

This level number is also the level number of any component(s) belonging to the program.

MSHP supports multiple levels of a program. A component can be installed several times with an identical product code but different levels.

apar-number

A string of seven characters consisting of one or two alphabetic characters, followed by five digits. Example:

```
DY12345
```

ptf-number

A string of seven characters consisting of two alphabetic characters, followed by five digits. Example:

```
UD12345
```

Note: The lib and sublib operands follow the syntax of the librarian program (for details, refer to the manual VSE/Advanced Functions System Control Statements).

lib

The name of a library.

sublib

the name of a sublibrary.

Note: Do not use $$MSHPxx as a sublibrary name.

member-name

The name of a sublibrary member (phase, module, or macro). It consists of one to eight alphameric characters, the first of which must be alphabetic.

member-type

The type of a sublibrary member. MSHP requires a member-type specification only when processing a macro.

Chapter 2. Function Control Statements

APPLY

Operation	Operands
APply	component[-level] : ptf-number [IRRevokable | REVokable] [INDirect]

The APPLY statement is used to install a single PTF to your system and to record the installation in the system history file.

Logical Unit Assignments

Required:

SYS001 Work file used by MSHP.

SYSLNK Linkage editor input file; needed to catalog phases supplied by IBM in object format.

SYSLST System printer.

Optional:

SYSPCH Needed if a backout PTF is to be generated (via the REVOKABLE operand).

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

AFFECTS
DATA
RESOLVES 'comment' APARS=(apar-number,...)

Optional:

DEFINE
INFLUENCES
INVOLVES
OR
REQUIRES
SUPERSEDES

Description of Operands

component[-level]:
Specifies the component to which the PTF is to be installed.

Level specifies the level number (formerly release number) of the applicable component.

If the level number is not specified, application of the PTF depends on how many levels of the component are installed. If there is only one level installed, the PTF is applied to this one; otherwise, MSHP informs you which levels are installed and asks you to which one you want to apply the PTF.

ptf-number:
Specifies the number of the PTF to be installed.

IRRevokable:
Specifies that, when installing the PTF, MSHP will not produce any backout PTF jobs. The PTF cannot be revoked, that is, the status before the installation of the PTF cannot be recreated at a later point in time.

REVokable:
Specifies that, when installing the PTF, a backout PTF job (with a REVOKE statement) is to be generated on SYSPCH. This allows you to recreate the status of your system as it existed before the installation of the PTF.

Restrictions:

- Do not specify REVOKABLE for a PTF that is a pre- or co-requisite for other PTFs or has comparable local/APAR fix dependencies.

- Do not specify REVOKABLE if the PTF contains new or additional modules or macros that are not part of the current component release.

INDirect:
Specifies a PTF for indirect application via the Service Dialog of VSE/SP. The operand indicates to the INSTALL SERVICE SD function that the sublibrary members affected by the service application are first to be applied to a reserved sublibrary $$MSHPIL before they are finally moved to the system sublibrary IJSYSRn.SYSLIB. This is to protect the IPLed SYSLIB in case the PTF application fails.

Example

APPLY 5666-301-02-H07 : UD19345

Detail Control Statements:

```
RESOLVES 'comment' APARS=(DY50001,DY50010)
AFFECTS MODULES=(IKRUPGR,IKRINSTL)
DATA
```

Note: For compatibility reasons, the APPLY statement is still accepted in the following format:

Operation	Operands
APply	component[(level)] : ptf-number [RELease=(release-number,...)]

(level):
Indicates the old three-character alphameric feature identifier of the component.

If this operand is specified, any following release information is ignored.

RELease = (release-number,...):
Specifies the release(s) of the component to which the PTF is to be installed.

This operand applies to old-format statements only and is ignored if level was specified (see above). If level was not specified, MSHP converts the release number into a level number.

If neither level nor release is specified, application of the PTF depends on how many levels (releases) of the component are installed. If there is only one level installed, the PTF is applied to this one; otherwise, MSHP informs you which levels are installed and asks you to which one you want to apply the PTF.

ARCHIVE

Operation	Operands
ARChive	product
ARChive	component-level [PTF=ptf-number| APAR=apar-number| SOFTreject]

The ARChive statement is used to make entries in the system history file.

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

The detail control statements related to the operands of the ARCHIVE statement are listed in the following table. Required detail statements are marked with an 'R', optional detail statements with an 'O':

ARCHIVE...	Prod.	Comp.	PTF	APAR
AFFECTS			R	R
ALTER				O
COMPRISES	R			
DEFINE	O	O	O	O
DELETE				O
INSERT				O
INVOLVES	O	O	O	
OR	O	O	O	
REPLACE				O
REQUIRES	O	O	O	
RESOLVES	R		R	O
SUPERSEDES			O	

Figure 3. Detail Control Statements Related to ARCHIVE Operands

Description of Operands

product:
Specifies that an entry for the named product is to be made in the system history file.

component-level:
Specifies that an entry for the named component is to be made in the system history file.

If a PTF or APAR is specified, 'component' identifies the component to which the particular PTF, local fix, or APAR fix to be archived applies.

PTF = ptf-number:
Identifies the PTF for which an entry is to be made in the history file.

APAR = apar-number:
Identifies the local or APAR fix for which an entry is to be made in the history file.

SOFTreject:
Specifies that a PTF which may have to be installed to the named component is to be installed even if, as a result, a local or APAR fix would be partially overwritten. (The same applies to a PTF that may have to be revoked.) For a component that is archived without SOFTREJECT specified, MSHP automatically rejects the installation (revocation) of a PTF that partially overwrites a local or APAR fix.

Use the option only if the result of a partial overwrite does not cause an immediate compatibility problem (as, for example, the replacement of asynchronously executed phases).

Example

ARCHIVE 5666-301-02-H07 PTF = UD23453

Detail Control Statements:

```
AFFECTS MODULE=MODAAA
RESOLVES 'comment' APAR=DY32555
```

BACKUP HISTORY

Operation	Operands
BACKup	History [SYStem|AUXiliary]

The BACKUP HISTORY statement requests MSHP to copy a history file located on disk onto magnetic tape.

Logical Unit Assignments

Required:

SYS006 The tape onto which MSHP writes the backup copy of the history file.

SYSLST System printer.

Required for BACKUP HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required for BACKUP HISTORY SYSTEM if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required: none

Optional: DEFINE

Description of Operands

SYStem:
Specifies that the system history file is to be copied to tape.

AUXiliary:
Specifies that the auxiliary history file is to be copied to tape.

Example

BACKUP HISTORY SYSTEM

BACKUP PRODUCT

Operation	Operands
BACKup	PRoduct=(product,...) ID='tapefile-id' [HEADer=member-name] [PRODuction] [GENEration]

The BACKUP PRODUCT statement is used to produce, on magnetic tape, a backup copy of the named program(s), referred to in this section as product(s). This backup copy consists of the production and generation sublibraries of the product(s), together with the pertinent history file containing product-related entries. You can later re-install the product(s) with the INSTALL PRODUCT statement.

Logical Unit Assignments

Required:

SYS006 The tape onto which MSHP writes the backup copy of the named product(s).

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Related Detail Control Statements

Required:

DEFINE HISTORY AUXILIARY

Optional:

DEFINE HISTORY SYSTEM

Description of Operands

PRoduct=(product,...):
Specifies the product(s) for which a backup copy is to be produced.

All requested products must reside in the same set of production and generation sublibraries, since MSHP copies only entire sublibraries.

ID='tapefile-id':
Specifies the identifier of the backup file. MSHP uses this identifier to locate the backup file during RESTORE. The tapefile-id can be one to 16 alphanumeric characters, enclosed in quotes; it must not contain any quotes.

HEADer=member-name:
Specifies an additional sublibrary member that is to be written as a header file onto the backup tape (as the very first file created with this BACKUP statement). This header file can be used to write some informational text, or job control statements, or copyright information in front of the backup file.

MSHP searches for the denoted member under type 'Z' in the production sublibrary of the product to be backed up.

'member-name' denotes the name of the sublibrary member containing the header file information.

PRODuction:
Specifies that only the production sublibrary of the named product(s) is to be copied.

GENEration:
Specifies that only the generation sublibrary of the named product(s) is to be copied.

If neither PRODUCTION nor GENERATION has been specified, both the production and the generation sublibraries are copied.

Example

```
BACKUP PRODUCT=(XE3G34,XE3G36) ID='POWER...2.2.0'
PRODUCTION
```

COPY HISTORY

Operation	Operands
COPy	History {SYStem AUXiliary |AUXiliary SYStem}

The COPY HISTORY statement requests MSHP to copy a history file from disk to disk.

Logical Unit Assignments

Required:

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY

Description of Operands

SYStem AUXiliary:
Creates a copy of the system history file for use as an auxiliary history file.

AUXiliary SYStem:
Copies an auxiliary history file to the system history file.

Note:

If the new (copied) history file extent is to reside on a newly defined VM minidisk, this minidisk must have been initialized by:

- *CMS command FORMAT, followed by*
- *VM disk initialization program IBCDASDI, or*

- *Device Support Facilities INIT command with parameter 'Mimic(Mini(n))'.*

Example

COPY HISTORY SYS AUX

CORRECT

Operation	Operands
CORrect	component[-level] : apar-number [REVokable|IRRevokable]

The CORRECT statement specifies that a local or APAR fix is to be installed to a component.

Note: To avoid an unintended removal of a fix due to linkage editor or assembly runs after the application of the fix, a correction should always be made in all applicable sublibrary members (phases, modules, macros).

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSLNK Linkage editor input file; needed if the correction requires link-editing.

SYSPCH Needed when correcting a macro and REVOKABLE is specified.

SYS001, SYS004 Needed as work files if the correction involves:
Modules,
Expandable phases, or
Macros.

SYS002, SYS003 Needed as work files if corrections to macros are involved.

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

AFFECTS

Optional:

ALTER
DEFINE HISTORY SYSTEM
DELETE
INFLUENCES

INSERT
INVOLVES
REPLACE
REQUIRES
OR
RESOLVES 'comment'
RESTART
SCAN
VERIFY

The detail control statements must be entered in the following sequence:

1. DEFINE HISTORY SYSTEM
2. REQUIRES, OR
3. RESOLVES
4. AFFECTS
5. ALTER, DELETE, INSERT, REPLACE, RESTART, SCAN, and VERIFY, if used, must be coded after the AFFECTS statement.
6. INFLUENCES, INVOLVES.

Description of Operands

component[-level]:
Specifies the component that is to be corrected by the local or APAR fix.

If level is not specified, the application of the fix depends on how many levels of the component are installed. If only one level is installed, MSHP applies the fix to this one; otherwise, MSHP informs you which levels are installed and asks you for the requested one.

apar-number:
Specifies the number of the local or APAR fix that contains the correction(s).

REVokable:
Specifies that corrections made to phases or modules can be removed by using the UNDO function.

For corrections to macros, the REVOKABLE option causes a job to be created on SYSPCH with the initiating control statement:

```
UNDO component:apar-number
```

The correction data consists of catalog requests for the unaltered version of the macro(s), enclosed in DATA statement delimiters.

IRRevokable:
Specifies that corrections cannot be revoked.

Example

CORRECT 5666-301-02-H07 : DY21001

Detail Control Statements:

```
AFFECTS PHASE=MAINPHA
SCAN 0730 ARG=4130A346
ALTER 0730 4130A346 : 47F0C71C
```

Note: For compatibility reasons, the CORRECT statement is still accepted in the following format:

Operation	Operands
CORrect	component[(level)] : apar-number ...

(level):
Indicates the old three-character alphameric feature identifier of the component.

If **level** is not specified, the application of the fix depends on how many levels of the component are installed. If only one level is installed, MSHP applies the fix to this level; otherwise, MSHP informs you which levels are installed, and asks you specify the level to which the fix is to be applied.

CREATE HISTORY

Operation	Operands
CReate	History [SYStem|AUXiliary]

The CREATE HISTORY statement requests MSHP to initialize a history file. (For information on creating extent definitions, refer to the DEFINE HISTORY detail control statement.)

Logical Unit Assignments

Required:

SYSLST System printer.

Required for CREATE HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required for CREATE HISTORY SYSTEM if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE

Description of Operands

SYStem:
Specifies that a system history file is to be initialized.

AUXiliary:
Specifies that an auxiliary history file is to be initialized.

Note: If the new (copied) history file extent is to reside on a newly defined VM minidisk, this minidisk must have been initialized by:

- *CMS command FORMAT, followed by*
- *VM disk initialization program IBCDASDI, or*
- *Device Support Facilities INIT command with parameter 'Mimic(Mini(n))'.*

Example

CREATE HISTORY SYSTEM

DUMP HISTORY

Operation	Operands
DUMP	History [SYStem|AUXiliary]

The DUMP HISTORY statement requests MSHP to produce a formatted hexadecimal printout of a history file on SYSLST.

This statement is provided primarily as an aid for program service. IBM service personnel may ask you to use it to produce a dump for diagnosis purposes.

Logical Unit Assignments

Required:

SYSLST System printer.

Required for DUMP HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required for DUMP HISTORY SYSTEM if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE

Description of Operands

SYStem:
Specifies that the system history file is to be dumped.

AUXiliary:
Specifies that the auxiliary history file is to be dumped.

Example

DUMP HISTORY AUX

INCORPORATE

Operation	Operands
INCorporate	component[-level] [RELease=release-number]

The INCORPORATE statement is used to install a component distributed in SYSIN format.

Logical Unit Assignments

Required:

SYSLNK Linkage editor input file.

SYS001 Linkage editor work file.

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

DATA

Optional:

DEFINE
INVOLVES
OR
REQUIRES

Description of Operands

component[-level]:
Identifies the component to be incorporated.

If you specify a level, any following release information is ignored. If you specify component without level, you must indicate the release number of the component in the RELEASE operand.

RELease = release-number:
Specifies the release of the component to be incorporated (only applicable if 'level' is not specified in the component operand). MSHP converts the release number into a level number.

Example

INCORPORATE 5666-301-06-H07

INSTALL PRODUCT/SYSRES

Operation	Operands
INSTall	{PRoduct|SYsres} [FROMTape ID='tapefile-id'] {INto=lib| [PRODuction INto=lib[.sublib]] [GENEration INto=(lib[.sublib],...)]}

The INSTALL statement requests MSHP to install either a licensed program (referred to as product in this section), such as VSE/VSAM, or a SYSRES package, such as VSE/Advanced Functions.

The history file that accompanies the programming support reflects the contents of the shipment package; it may contain information on any requirements that have to be met prior to installation (for example, pre-requisite components or PTFs). When executing the INSTALL function, MSHP informs you of any missing requirements.

MSHP restores the history file from the distribution tape into an auxiliary history file. You may either use the standard SYS002 work file for the auxiliary history file or define it with a DEFINE HISTORY AUXILIARY statement.

With the restored auxiliary history file, checking for pre-, co-, and negative-requisites is performed. If all checks and verifications prove satisfactory, the distribution libraries are restored into the specified target libraries, and the restored distribution history file is merged with the current system history file.

MSHP also determines (by analyzing the shipment history file and your system's history file) which of the products already installed in your system are compatible with the shipped product and which are superseded:

- Products that are based on the same base products are usually *compatible* with each other, as, for example, VSAM Space Management and VSAM Backup/Restore, which both require VSE/VSAM as a base product. This relationship can also be explicitly defined via the COMPATIBLE detail control statement.

- An installed product is *superseded* when you install a follow-on release of that product. In that case MSHP (1) informs you that the new shipment package supersedes the current level of the product and (2) asks you whether you want to keep the old version of the product or delete it.

Logical Unit Assignments

Required:

SYS006 Distribution tape.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY
COMPATIBLE

Description of Operands

PRoduct:
Specifies that a product (non-SYSRES package) is to be installed.

SYSres:
Specifies that a SYSRES package is to be installed.

FROMTape ID='tapefile-id':
Indicates that the distribution tape is to be searched for the specified tapefile-id. This must be identical with the tapefile-id specified in the BACKUP statement. If the tape is not correctly positioned, it is scanned (forward only) for the specified ID, and correctly positioned before installation.

The tapefile-id can be one to 16 alphameric characters.

If the operand is omitted, it is assumed that the tape is correctly positioned to the product to be installed.

{INto=lib|
[PRODuction INto=lib[.sublib]]
[GENEration INto=(lib[.sublib],...)]}

Specifies the names of the libraries and sublibraries into which the members from the distribution tape are to be copied. Via the parameters PRODUCTION and GENERATION you may indicate that you want to install either the production part or the generation part of the shipment package only, or install the two parts into separate libraries.

Defaults:

- If, for INSTALL SYSRES, you do not specify a sublibrary (sublib) name, MSHP uses the name of the shipment sublibrary, which is SYSLIB. If the target libraries (lib) do not exist, MSHP creates them. However, you must provide the necessary label information (DLBL/EXTENT) for the libraries.

- For INSTALL PRODUCT, you need not specify the names of the target libraries and sublibraries, provided these libraries exist (on-line) and label information is available in the label area. If you specify neither a library nor a sublibrary, MSHP takes one of the following as target library:

  1. If the shipped product is already installed, as, for example, in the case of a refresh installation, the library and sublibrary of the installed product.

  2. If the shipped product supersedes another one, the library and sublibrary of the superseded product.

  3. Otherwise, the library and sublibrary of a compatible product, if there exists any.

  4. If none of the above, MSHP notifies you and terminates installation.

  If you specify a target library, but no sublibrary, MSHP takes one of the following:

  1. The sublibrary of any superseded product with the same library.

  2. Or the sublibrary of any compatible product with the same library.

  3. If none of the above, the sublibrary name of the shipped production and/or generation sublibrary.

INto = lib:
Specifies, for INSTALL PRODUCT only, installation of both the production and the generation part of the shipment package into the library denoted by 'lib'.

Note: This operand cannot be used for INSTALL SYSRES, because the generation part and the production part must be installed into different target libraries.

PRODuction INto = lib[.sublib]:
Specifies installation of the executable (production) part of the shipment package, which consists of all phases, procedures (and some modules/macros) needed for daily operation of your system or product. The production part must be installed before the generation part.

MSHP merges the members of the production shipment sublibrary into the target sublibrary indicated by lib[.sublib] or, if sublib has been omitted, into the default sublibraries established by MSHP (see above, under "Defaults").

For INSTALL SYSRES, however, the name of the production library must always be specified as IJSYSRn, n being a digit from 1 to 9. The generation part must be installed into a different library than the production part.

GENEration INto=(lib[.sublib],...):
Specifies installation of the generation part of the shipment package, which contains those modules and, possibly, macros that are needed for the regeneration of the product.

MSHP merges the members of the shipped generation sublibrary into the target sublibrary (or sublibraries) indicated by (lib[.sublib],...), or if sublib has been omitted, into the default sublibraries established by MSHP (see above, under "Defaults").

Examples

```
INSTALL PRODUCT FROMTAPE ID='LM4E11' -
        INTO=USER01

INSTALL PRODUCT FROMTAPE ID='LM4E11' -
    PRODUCTION INTO=USER01.PRODALL

INSTALL SYSRES FROMTAPE ID='301H07' -
        PRODUCTION INTO=IJSYSR1 -
        GENERATION INTO=GENLIB
```

INSTALL SERVICE/BACKOUT

Operation	Operands
INSTall	{SErvice[REVokable|REStart] [TAPes={1|no.-of-tapes}] [SD] |BAckout[REStart]}

The INSTALL SERVICE statement requests MSHP to install PTFs from one or more service tapes.

The INSTALL BACKOUT statement requests MSHP to install one or more backout PTFs, which means recataloging the sublibrary member(s) replaced by installing the corresponding PTF(s). The statement works in the same way as the INSTALL SERVICE statement, except that it reads the PTF information from the backout tape, which is created when you specify the REVOKABLE operand in the INSTALL SERVICE statement.

Via the INCLUDE and EXCLUDE detail control statements you can specify that only certain products, components, or PTFs are to be included or excluded during the service application. If you omit the EXCLUDE and INCLUDE statements, all service tape PTFs which are applicable to your system will be selected for service installation.

MSHP prints a list of all PTFs that are to be installed and asks you for confirmation before it replaces the affected members in your sublibraries and updates the history file.

Logical Unit Assignments

Required:

SYS006 Service/backout tape.

SYS001,

SYS002,

SYS003 Work files used by MSHP.

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

SYS004 Needed if backout PTF jobs are to be created (by specifying the REVOKABLE operand).

Related Detail Control Statements

Required:

none

Optional:

INCLUDE
EXCLUDE (for INSTALL SERVICE only)

Description of Operands

SErvice:
Specifies that MSHP is to install PTFs from the service tape(s), as detailed in any INCLUDE or EXCLUDE detail control statements.

REVokable:
Specifies that backout jobs are to be created for all PTFs that are to be installed. The backout jobs are MSHP jobs with the REVOKE function control statement included. The backout jobs are written in blocked format onto a tape which must be assigned as SYS004. This tape can later be used as input for the INSTALL BACKOUT statement to re-install the PTFs, if necessary.

BAckout:
Specifies that MSHP is to install one or more backout PTFs from the backout tape which is produced by the INSTALL SERVICE function with the REVOKABLE option. The PTFs to be installed can be selected via the INCLUDE statement.

PTFs with common sublibrary members are grouped together into one single PTF which contains all the members of the PTFs. In that case, a separation via the INCLUDE statement is not possible at INSTALL BACKOUT time.

REStart:
Requests MSHP to restart a previous INSTALL SERVICE/BACKOUT or APPLY/REVOKE job whose final link step failed. MSHP scans the history file entries for those PTFs that were correctly cataloged, but not yet linked, and invokes the linkage editor to complete the final link step.

TAPes = no.-of-tapes:
Required only if two or more tapes are to be processed. Indicates to MSHP the number of tape volumes that have to be scanned for the particular service installation. If you know that prerequisite PTFs exist on other service tapes and that these PTFs are not yet installed, have MSHP scan these additional tape volumes for the prerequisite PTFs and have them retrieved for installation.

Mount the first tape on the tape drive assigned to SYS006 before you enter the EXEC MSHP command or statement. MSHP scans this tape and then issues message M363D, prompting you to mount the next tape on the same tape drive. When the last tape has been scanned, MSHP processes it and

then issues message M363D again. You must now mount the first tape again, this time for processing. After processing each tape, MSHP issues message M363D, prompting you for the next tape.

For example, if you specify TAPES=3, the sequence of events is:

Mount tape 1;
MSHP scans tape 1;
Mount tape 2;
MSHP scans tape 2;
Mount tape 3;
MSHP scans *and processes* tape 3;
Mount tape 1 again;
MSHP processes tape 1;
Mount tape 2 again;
MSHP processes tape 2.

The maximum number that can be specified is 9. If the operand is omitted, *one* tape volume is assumed.

SD:
This operand indicates that service is to be applied via the VSE/SP Service Dialog. For those PTFs that are flagged with the INDIRECT option (in the APPLY statement), the members affected by the service application are first applied to a reserved sublibrary $$MSHPIL before they are finally moved into the system sublibrary IJSYSRn.SYSLIB. This is to protect the IPLed SYSLIB in case the PTF application fails.

Examples

INSTALL SERVICE

INSTALL SERVICE RESTART (no detail control statements needed)

INSTALL BACKOUT

Detail Control Statements

EXCLUDE PRODUCT=(301H07)

INCLUDE PTF=(UD24500,UD34000)

LIST

Operation	Operands
LIST	SERVICEtape [DOCument | NODOCument] [XREF | NOXREF] [COVer [CONTinuous | SEParate] | NOCOVer]

The LIST statement requests MSHP to print, on SYSLST, information from a service tape.

Logical Unit Assignments

Required:

SYS001 Work file used by MSHP.

SYS003 Work file used by MSHP.

SYS006 Service tape.

SYSLST System printer.

Optional:

SYS002 Needed as work file if XREF is specified.

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

PTF (if COVER is specified)

Description of Operands

SERVICEtape:
Specifies that information from a service tape is to be printed.

DOCument:
Specifies printing of the service tape documentation, which contains information on how to apply corrective and preventive service from the service tape.

NODOCument:
Suppresses the DOCUMENT function.

XREF:
Specifies printing of the cross-reference list of all PTFs and APARs shipped on the service tape.

NOXREF:
Suppresses the XREF option.

COVer:
Specifies printing of the cover letters of those PTFs that are listed on an associated PTF detail control statement. If no PTF statement is given, the cover letters of all PTFs on the service tape are printed. The following is printed for all requested PTFs:

- Job control statements (including comments)
- MSHP control statements
- Librarian commands
- Linkage editor control statements

NOCOVer:
Suppresses the COVER function.

CONTinuous:
Specifies that the cover letters of the PTFs are to be printed without starting a new page for each PTF.

SEParate:
Causes a new page to be started for each PTF cover letter that is to be printed.

Example

LIST SERVICETAPE COVER SEPARATE

Detail Control Statement:

```
PTF=(UD34200,UD34201,UD34202)
```

LOOKUP

Operation	Operands
LOokup	PRoduct=product
LOokup	[component[-level]] [PTF=ptf-number| APAR=apar-number| PHAse=member-name[DATA]| MODule=member-name[DATA]| MACro=member-name[TYpe={E|member-type}][DATA]]

The LOOKUP statement requests MSHP to display, on SYSLOG, selected information from the system history file.

Logical Unit Assignments

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY

Description of Operands

PRoduct = product:
Indicates that the following information is to be displayed for the specified product-id:

- Date of installation.
- Requirements to be met for installation.
- Component(s) contained in the product.
- Comments, if any.

component[-level]:
Specifies the component for which information is to be displayed.

If level is omitted and more than one level of the component is installed, all levels of the component information will be displayed.

If component is specified without any further operands, the following history file information is displayed:

- Component identifier plus release level.
- Latest service: number of the most recently applied PTF and its application date, or NO PTF applied.
- Latest APAR or latest fix application: number of the most recently applied local or APAR fix and its application date.
- Invalidated APARs: a list of local and/or APAR fixes that have been invalidated by the application of a PTF.
- Incomplete APARs: a list of local and/or APAR fixes whose application is incomplete.

PTF=ptf-number:
Indicates that, for the given ptf-number, the following history file information is to be displayed:

- PTF number.
- Applied / Not applied / Revoked.
- Date of application (if applied).
- Superseded by / Not superseded.
- Affected component.
- Resolved APAR(s).
- Affected phases, modules, or macros.
- Prerequisites.
- Involved link-edits.
- Comments, if any, included in the PTF.

APAR=apar-number:
Indicates that, for the given apar-number, the following history file information is to be displayed:

- APAR number (also for local fixes).
- Fixed / Not fixed by PTF / Local fix (if fixed).
- Date of correction (if fixed).
- Affected component.
- Affected phases/modules; if locally fixed and fix is recorded: alterations.
- Affected macros; if locally fixed and fix is recorded:
  - Insertions
  - Deletions
  - Replacements
- Comments, if any, included in the APAR.

PHAse=member-name:
Indicates that, for the given phase-name, the following history file information is to be displayed:

- Phase name.
- Not affected / Affected by PTF.
- Date when affecting PTF was applied by local fix.
- Date when local/APAR fix was made; if local fix was recorded: alterations.

MODule = member-name:
Indicates that, for the given module-name, the following history file information is to be displayed:

- Module name.
- Not affected / Affected by PTF.
- Date when affecting PTF was applied by local fix.
- Date when local/APAR fix was made; if local fix was recorded:
  - CSECT
  - Expansion
  - Alterations

MACro = member-name:
Indicates that, for the given macro-name, the following history file information is to be displayed:

- Macro name.
- Not affected / Affected by PTF.
- Date when affecting PTF was applied by local fix.
- Date when local/APAR fix was made; if local/APAR fix was recorded:
  - Insertions
  - Deletions
  - Replacements

TYpe = member-type:
Indicates the type of the specified macro. Member-type may be one character only.

If the operand is omitted, type E is assumed.

DATA:
Specifies that the source data from which the phase/module/macro was generated (with TAILOR KEEPDATA) is to be displayed.

Examples

```
LOOKUP 5666-301-02-H07 PTF=UD00001
LOOKUP PRODUCT=301H07
LOOKUP PTF=UD34500
LOOKUP APAR=DY34200
LOOKUP PHASE=$$A$SUP1
```

MERGE HISTORY

Operation	Operands
MERge	History {SYStem AUXiliary |AUXiliary SYStem}

The MERGE HISTORY statement requests MSHP to insert entries of one history file into another history file.

The sequence of the keywords SYSTEM and AUXILIARY defines the direction of the merge operation. The first keyword specifies the source history file, and the second the target history file. The two keywords must be specified adjacent to each other.

Restriction: Both the source and the target history files must reside on disk.

Logical Unit Assignments

Required:

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE

Description of Operands

SYStem AUXiliary:
Specifies that entries from the system history file are to be merged into an auxiliary history file.

AUXiliary SYStem:
Specifies that entries from an auxiliary history file are to be merged into the system history file.

Example

MERGE HISTORY SYS AUX

PATCH

Operation	Operands
PAtch	Sublibrary=lib.sublib

The PATCH control statement allows you to change (patch) a phase stored in a sublibrary. MSHP does not record the change in the history file.

The phase you patch may or may not be under control of MSHP. If the phase is MSHP-controlled, MSHP issues a warning message at the console.

Logical Unit Assignments

When entered at the console, none.

Required when entered from SYSIPT:

SYSLST System printer

Related Detail Control Statements

Required:

AFFECTS

Optional:

ALTER

SCAN

The AFFECTS statement must precede any optional detail control statements.

Description of Operands

Sublibrary = lib.sublib
For lib in lib.sublib, supply the name of the library that is to be accessed.

For sublib in lib.sublib, supply the name of the sublibrary in which the affected phase is stored.

Example

Assuming that the phase to be changed resides in sublibrary PAYSUBL of library WEEKLIB, then your specification is:

```
sublibrary=weeklib.paysubl
```

PERSONALIZE

Operation	Operands
PERsonalize	['customer-name'] [ADDRess='customer-address'] [PHone='phone-number'] [PRogrammer='programmer-name'] [ENVironment='description']

The PERSONALIZE statement is used to identify a history file and relate it to a specific user.

Restrictions:

- To personalize your system's history file, MSHP needs at least one operand.
- If the history file has not been personalized before, specification of customer-name and customer-address is mandatory.
- The first personalization of a history file changes the dates of all PTF entries to the date when the PERSONALIZE statement is given.

Logical Unit Assignments

Required:

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY

Description of Operands

'customer-name':
Specifies the user's name that is to be entered in the history file.

The name is a string of one to twenty characters, enclosed in quotes. If fewer than 20 characters are specified, the entry in the history file is padded with trailing blanks.

Restriction: If the history file has not been personalized before, customer-name must be specified.

ADDRess='customer-address':
Specifies the address that is to be entered in the history file.

The address is a string of 1 to 45 characters, enclosed in quotes. If fewer than 45 characters are specified, the string is padded with trailing blanks.

Restriction: If the history file has not been personalized before, customer-address must be specified.

PHone='phone-number':
Specifies the phone-number that is to be entered in the history file.

The phone number is a string of 1 to 17 characters, enclosed in quotes. If fewer than 17 characters are specified, the string is padded with trailing blanks.

A null string (two consecutive quotes) is accepted; it erases a previously specified number.

PRogrammer='programmer-name':
Specifies the programmer name that is to be entered in the history file.

The programmer name is a string of 1 to 24 characters, enclosed in quotes. If fewer than 24 characters are specified, the string is padded with trailing blanks.

A null string (two consecutive quotes) is accepted; it erases a previously specified name.

ENVironment='description':
Specifies any additional information (for example, the release level) that is to be entered in the history file.

The operand is a string of 1 to 62 characters, enclosed in quotes. If fewer than 62 characters are specified, the string is padded with trailing blanks.

A null string (two consecutive quotes) is accepted; it erases a previously specified description.

Example

```
PERSONALIZE 'M. Morris' -
ADDRESS='60 Water St., New York, N.Y.' -
PHONE='633 4537'
PROGRAMMER='JOHN'
ENVIRONMENT='REL2.1'
```

REMOVE

Operation	Operands
REMove	product
REMove	component-level [PTF=ptf-number| APAR=apar-number| PHAse=member-name| MODule=member-name| MACro=member-name [TYpe=member-type]]

The REMOVE statement requests MSHP to erase entries from the system history file. The space of the removed history file entries is freed for future use.

Note: MSHP does not remove an APAR that was archived as a 'resolved' APAR in conjunction with a PTF.

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY

Description of Operands

product:
Indicates that the entry for the specified product is to be removed from the history file.

component-level:
Indicates that the entry for the specified component is to be removed (if no further operand follows).

If followed by another operand, 'component' indicates the component to which the specified PTF, APAR, or member-name refers.

PTF = ptf-number:
Indicates that the entries associated with the specified PTF number are to be removed.

APAR = apar-number:
Indicates that the entry for the indicated APAR/local fix is to be removed.

PHAse = member-name:
Indicates that the entry for the specified phase name is to be removed.

MODule = member-name:
Indicates that the entry for the specified module name is to be removed.

MACro = member-name:
Indicates that the entry for the specified macro name is to be removed.

TYpe = member-type:
Indicates the type of the macro to be removed. Member-type may be one character only.

If the operand is omitted, type E is assumed.

Example

REMOVE 5666-301-02-H07 PTF = UD12345

RESIDENCE

Operation	Operands
RESIdence	PRODuct=(product,...) PRODuction=lib.sublib [GENEration=(lib.sublib,...)]

The RESIDENCE statement defines the names of the production and generation sublibraries in which the named products are to reside. This information is recorded in the history file for any follow-on activities, such as service applications, tailoring, installation, or product backup.

Any sublibrary names previously recorded in the history file (via another RESIDENCE or an INSTALL statement) are erased.

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY SYSTEM

Description of Operands

The operands PRODuct= and PRODuction= must be entered in the sequence as shown.

PRODuct=(product,...):
Specifies the name(s) of the product(s) whose residence is to be defined.

PRODuction=lib.sublib:
Indicates that the production part of the product(s) is to reside in the specified sublibrary.

GENEration = (lib.sublib,...):
Indicates that the generation part of the product(s) is to reside in the specified sublibrary.

Example

```
RESIDENCE PRODUCT = LM4E11 PRODUCTION = LIB01.PR$E11 -
                    GENERATION = LIB02.G1$E11
```

RESTORE PRODUCT/SYSRES

Operation	Operands
RESTore	{PRoduct|SYsres} {INto=lib| PRODuction INto=lib[.sublib] GENEration INto=(lib[.sublib],...)} [ID='tapefile-id']

The RESTORE statement is used to restore a complete shipment tape (production part, generation part, and shipment history file) onto disk; however, without any checks or updates of the system history file. The history file included in IBM's shipment tape is restored into an auxiliary history file.

Logical Unit Assignments

Required:

SYS006 Distribution tape.

SYSLST System printer.

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY AUXILIARY

Description of Operands

PRoduct:
Specifies that a non-SYSRES package is to be restored.

SYsres:
Specifies that a SYSRES shipment package is to be restored.

INto = lib:
Specifies, for RESTORE PRODUCT only, that both the production and the generation part of the shipment package are to be restored into the library denoted by lib.

Note: Cannot be used for RESTORE SYSRES, because the production part and the generation part must be restored into different target libraries.

PRODuction INto = lib[.sublib]:
Specifies that the production part of the shipment package is to be restored to the named library (and sublibrary).

For RESTORE PRODUCT, the target library must exist (on-line), and label information must be available for it in the label area.

For RESTORE SYSRES, the name of the target library for the production part must be IJSYSRn, n being a digit from 1 to 9. (The name of the target library for the generation part must be different.) IJSYSRn will be created by MSHP if it does not exist.

If you omit 'sublib', MSHP takes the name of the shipment sublibrary as default.

GENEration INto = (lib[.sublib],...):
Specifies that the generation part of the shipment package is to be restored to the named library or libraries.

For RESTORE PRODUCT, the target libraries must exist (on-line), and label information must be available in the label area.

For RESTORE SYSRES, the target library is created by MSHP if it does not exist (label information must be available in the label area).

If you omit 'sublib', MSHP takes the name of the shipment sublibrary as default.

ID = 'tapefile-id':
Indicates that the shipment tape is to be searched for the denoted tapefile-id, which was specified in the BACKUP statement. If the tape is not correctly positioned, it is scanned (forward only) for the specified tapefile-id and correctly positioned.

The tapefile-id can be 1 to 16 characters.

If you omit the operand, the tape is assumed to be correctly positioned.

Example

```
RESTORE PRODUCT INTO=PRODLIB
```

RESTORE HISTORY

Operation	Operands
RESTore	History [SYStem|AUXiliary]

The RESTORE HISTORY statement requests MSHP to write a history file located on magnetic tape onto disk. If the tape containing the history file was not written using the BACKUP HISTORY statement, the tape must be positioned at the file containing the history file before you issue the RESTORE statement.

Logical Unit Assignments

Required:

SYS006 The tape containing the history file.

SYSLST System printer.

Required additionally for RESTORE HISTORY AUXILIARY:

SYSyyy The device on which the auxiliary history file resides. This can be either SYS002 or the device specified in the UNIT operand of the DEFINE HISTORY AUXILIARY statement.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Optional:

DEFINE HISTORY

Description of Operands

SYStem:
Specifies that a history file on tape is to be copied to the system history file (a disk file with the file name IJSYSHF).

AUXiliary:
Specifies that a history file on tape is to be copied to the auxiliary history file (a disk file with the file name IJSYS02).

Example

RESTORE HISTORY AUX

RETRACE

Operation	Operands
RETRace	[PRODucts| COMPonents[IDentifier=component[-level]]| PTFs| APARs| MEMbers]

The RETRACE statement requests MSHP to print information from the system history file on SYSLST. The listings produced are identified with the PERSONALIZE information (contained in the history file header record).

On a system running under VSE/System Package Version 3, Release 1, listing includes the refresh level indication of the system.

Defaults: If RETRACE is specified without any keywords, MSHP writes, to SYSLST, a report on the system's current service level; this report contains:

1. A list of all products and components installed.
2. A combined list of all local fixes, sorted by APAR number, and of all applied and not superseded PTFs, sorted by PTF number.
3. An APAR cross-reference list. It lists, for all APARs that have been applied to the system (in APAR number sequence), whether a particular APAR was corrected locally (local fix) or whether it has been resolved by a PTF and, if so, by which PTF.
4. A member cross-reference list. It lists, for all sublibrary members that were affected by a PTF or local fix, the affecting PTF or APAR number. This listing is in alphabetical order by member name (without respect to member type).

Logical Unit Assignments

Required:

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

none

Optional:

DEFine HISTORY

Description of Operands

PRODucts:
Requests MSHP to print a list of all products installed, together with the following information:

- Date of installation (or personalization)
- Component(s) contained in the product
- Comments, if any

COMPonents:
Requests MSHP to print a list of all component records from the history file. The printout includes (for each component):

- Component identifier plus release level
- Date of installation (or personalization)
- All PTFs applied to the particular component (in PTF number sequence)
- All APARs and local fixes applied to the particular component (in APAR number sequence)
- All generated members (for the TAILOR function)

COMPonents IDentifier=component[-level]:
Requests MSHP to print information for the specified component only. If level is omitted, MSHP prints information for all installed levels of the component.

PTFs:
Specifies that all applied PTFs are to be listed (in PTF number sequence). For each PTF the following information is printed:

- PTF number
- Whether or not the PTF was revoked
- Component to which the PTF applies
- Affected members
- Resolved APARs
- Prerequisites, corequisites, and also negative prerequisites
- PTFs which this PTF supersedes
- The PTF that supersedes this PTF, and the date of application of the superseding PTF

APARs:
Specifies that all APARs are to be listed (in APAR number sequence) which were corrected by a PTF or local/APAR fix. For each APAR the following information is printed:

- APAR number
- Component to which the APAR applies
- PTF number, if the APAR has been resolved by a PTF
- Date of correction
- If locally corrected:
  - Affected modules
  - What the fix consisted of

MEMbers:
Specifies that all phases, modules, and macros that are affected by a PTF or local fix are to be listed. For each member the following information is printed:

- Member name
- Component to which the member belongs
- Date of PTF or local fix application
- PTF number, if affected by a PTF
- APAR number; if locally corrected, also what the fix consisted of

Note: Since RETRACE MEMBERS does not indicate whether an APAR, PTF, or component is incorrect or incomplete, use RETRACE APARS|PTFS|COMPONENTS instead.

Examples

```
RETRACE PRODUCTS
RETRACE COMP ID=5666-301-01-H07
RETRACE PTFS
```

REVOKE

Operation	Operands
REVoke	component[-level] : ptf-number

The REVOKE statement initiates a backout PTF job that contains the phases, modules, and macros as they were before the named PTF was installed. This backout PTF job (with the initial REVOKE statement) is generated by the APPLY or INSTALL SERVICE statements if REVOKABLE was specified.

By bringing the backout PTF back onto the system (with the INSTALL BACKOUT statement), MSHP restores the system to the status that existed before the original PTF was installed. MSHP also flags the history file entry for the PTF as revoked.

A PTF cannot be revoked if it is a prerequisite for another PTF that has not been revoked previously.

Logical Unit Assignments

Required:

SYSLNK Linkage editor input file.

SYS001 Linkage editor work file.

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

DATA

Optional:

none

Description of Operands

component[-level]:
Identifies the component for which the backout PTF was generated.

If level is not specified and only one level of the specified component is installed, the REVOKE job is applied to this one. If two or more levels are installed, MSHP informs you which levels are installed and asks you to which one the REVOKE job applies.

ptf-number:
Identifies the PTF that is to be revoked. The number to be specified is that of the originally applied PTF (the one that proved to be unsatisfactory). The PTF is flagged as 'revoked' in the system history file.

Example

```
REVOKE 5666-301-02-H07 : UD00001
```

SELECT

Operation	Operands
SELect	GENFile COMPonent=component[-level]

The SELECT statement identifies the generation file, from which individual phases, modules, or macros can be regenerated (with the GENERATE detail control statement) after a service application.

The generation file is a set of MSHP tailor jobs, each of which must be preceded by a // JOB statement and followed by a /& statement. The generation file may reside on tape, disk, or diskette. The records of the file can be 80 or 81 bytes long. The generation file can also be a card deck, in which case it must be terminated by a /* card, immediately followed by the last /& card.

Logical Unit Assignments

Required:

SYS005 Generation file.

SYSLNK Linkage editor input file.

SYS001 Linkage editor work file.

SYSLST System printer.

Optional:

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

GENERATE

Optional:

DEFINE HISTORY

Description of Operands

GENFile:
Indicates the generation file.

COMPonent = component[-level]:
Identifies the component to which the members to be regenerated belong.

The level indication must be the same as that of the corresponding TAILOR job. If level was not specified during tailoring or retailoring, and multiple levels of the affected component are installed, MSHP asks you for which level the generation is to be done.

Example

SELECT GENFILE COMPONENT = 5666-301-02-H07

TAILOR

Operation	Operands
TAILor	component[-level] {PHAse=member-name| MODule=member-name| MACro=member-name [TYpe=member-type]} [KEEPdata]

The TAILOR statement (together with the EXECUTE detail control statement) is used to generate (tailor) sublibrary members of components that are shipped in source-macro format and that have to be assembled and link-edited according to the specific needs of your installation (for example, supervisor macros).

Instead of invoking the assembler or linkage editor directly, use the EXECUTE statement, which causes MSHP to call the specified program(s) and to record the processed data in the system history file.

If later a PTF is installed to a generated sublibrary member, you may have to regenerate (retailor) the member (also with the TAILOR statement) in order to make the applied fix active for the system. Note, however, that it is not possible to retailor a sublibrary member directly after having tailored it, that is, without a prior service application.

Logical Unit Assignments

Required:

SYSLNK Linkage editor input file.

SYS001 Linkage editor/assembler work file.

SYS002, SYS003 Assembler work files.

SYSLST System printer.

Optional:

SYS004 Work file needed by MSHP if MODULE= or MACRO= is specified.

SYSxxx Required if the device on which the system history file resides is specified in the UNIT operand of the DEFINE HISTORY SYSTEM statement.

Related Detail Control Statements

Required:

EXECUTE

Optional:

RESOLVES
DEFINE

Description of Operands

component[-level]:
Identifies the component containing the macro, module, or phase to be tailored.

If level is omitted and only one level of the affected component is installed, MSHP generates the sublibrary member for this one; if two or more levels are installed, MSHP informs you which levels are installed and asks you for which one you want to generate the member.

PHAse = member-name:
Specifies the name of the phase to be generated. (See Note, below.) For retailoring, generic names such as DFH* are allowed.

MODule = member-name:
Specifies the name of the module to be generated. (See Note, below.) For retailoring, generic names such as DFH* are allowed.

MACro = member-name:
Specifies the name of the macro (definition) to be assembled. (See Note, below.) For retailoring, generic names such as DFH* are allowed.

Note: The operands PHASE=, MODULE=, or MACRO= do not generate any PHASE or CATALOG statements. You have to include these statements after the EXECUTE detail control statement. MSHP uses the information to:

- *Check if a specified member already exists in the history file.*
- *Compare the PHASE, MODULE, or MACRO operands with the corresponding PHASE or CATALOG statements given.*
- *Enter the member name into the history file.*

TYpe = member-type:
Indicates the type of the macro to be assembled. Member-type may be one character only.

If the operand is omitted, type E is assumed.

KEEPdata:
Specifies that the source code processed by the invoked control program(s) is to be stored in the system history file. MSHP uses that code if retailoring is to be done later on. Instead of resubmitting the original tailor job, it is sufficient to invoke MSHP and initiate, for example, the function

```
TAILOR component PHASE=member-name
```

without an EXECUTE detail control statement.

MSHP fetches, from the system history file, all information and data belonging to the generated phase and reassembles (and recatalogs) it.

For those tailor jobs that are too large to be kept in the history file, do not specify KEEPDATA. Instead, create a sequential generation file (with the filename GENFILE) and put the original tailor job(s) into it. To create this file, you can use a program of your own or an IBM-supplied program such as OBJMAINT or DITTO (Data Interfile Transfer, Testing, and Operations utility).

Each individual tailor job on the generation file must be preceded by a // JOB statement and followed by a /& statement. The generation file must be assigned to SYS005; it can be on tape, disk, or diskette, and the records must be 80 or 81 bytes long. The generation file can also be a card deck, in which case the last card must be a /* card, immediately followed by the last /& card.

You can then retailor individual members from the generation file by invoking MSHP with the SELECT GENFILE function control statement and a GENERATE detail control statement for each member to be retailored.

Example

TAILOR 5666-301-02-H07 PHASE=$$A$SUP8 KEEPDATA

UNDO

Operation	Operands
UNdo	component[-level] : apar-number

The UNDO statement is used to re-establish the status of a sublibrary member as it existed before a local or APAR fix was applied with the CORRECT...REVOKABLE statement.

Restriction: If a phase has been expanded by a local or APAR fix, this expansion cannot be removed. The phase remains expanded.

For phases and modules, MSHP may be invoked with an UNDO statement that refers (by component and apar-number) to the correction as specified in the CORRECT statement; MSHP uses the information to remove the correction from the respective library and the system history file. For macros, the UNDO statement is included in the job created (on SYSPCH) by CORRECT...REVOKABLE.

Logical Unit Assignments

Same as for CORRECT.

Related Detail Control Statements

Required:

none

Optional:

DEFINE HISTORY SYSTEM
DATA

Description of Operands

component[-level]:
Specifies the component from which the local or APAR fix (initiated by CORRECT) is to be removed. If level is omitted and only one level of the component is installed, MSHP removes the fix from this one. If two or more levels are installed, MSHP informs you which levels are installed and prompts you for the level of the applicable component.

apar-number:
Specifies the local or APAR fix (initiated by CORRECT) that is to be removed.

Example

UNDO 5666-301-02-H07 : DY19227

Chapter 3. Detail Control Statements

AFFECTS

Operation	Operands
AFFects	[PHAses=(member-name,...) [EXPand=size-increment]] [MODules=(member-name,...) [LIOCS] [ESDid={001|esd-number}] [EXPand=size-increment]] [MACros=(member-name,...) [TYpe={E|member-type}]]

The AFFECTS statement identifies the phases, modules, and macros that are affected by a PTF or local fix application.

Restrictions:

One AFFECTS statement may not refer to more than a total of 100 phases, modules, and macros.

If AFFECTS is used as a detail control statement to CORRECT, or when archiving a local/APAR fix and the fix information itself, only one phase, module or macro may be specified.

Description of Operands

PHAses = (member-name,...):
Specifies the affected phase(s).

EXPand = size-increment:
Indicates that the specified phase or module (see below) is to be made larger by the number of bytes specified in size-increment, so that fix code can be added at the end of the phase or module. Size-increment is a decimal number of one to six digits.

EXPAND may be specified only when applying a local/APAR fix (with CORRECT) or archiving a local/APAR fix.

MODules = (member-name,...):
Specifies the affected module(s).

LIOCS:
Indicates that a LIOCS module is affected by the PTF. However, only the macro needed to generate the module is distributed in the PTF, but not the affected module itself.

ESDid = esd-number:
Indicates that a change applies to the specified ESD.

Default: If not specified, ESDID = 001 is assumed, that is, the first ESD.

Restriction: ESDID may be specified only when correcting a component (CORRECT) or archiving (ARCHIVE) a local/APAR fix.

For esd-number specify one to three hexadecimal digits. If less than three digits are specified, the number is padded with leading zeros.

MACros = (member-name,...):
Specifies the affected macro(s).

TYpe = member-type:
Indicates the type of the affected macro. The member-type can be one character only.

If the operand is omitted, type E is assumed.

Example

AFFECTS MACRO = GENAB

Note: For compatibility reasons, the operand CSect = csect-number is still accepted (for ESDid).

ALTER

Operation	Operands
ALter	address old-text : new-text

The ALTER statement identifies the modifications that are to be made to a phase or module. This includes verification of the alteration for phases and, optionally, for modules.

Description of Operands

address:
Specifies the (relative) address where the new-text is to begin to replace the old-text.

Address is a string of one to six hexadecimal digits. Leading zeros may be omitted.

old-text:
Specifies the text that is to be replaced.

MSHP checks the text in the phase or module at the specified address whether it is identical with the old text; replacement by new text takes place only if the text is identical.

Old-text *must* be specified when modifying a phase; it *may* be specified when modifying a module.

If the phase is expanded and the old text is in the expanded area, specify a pair of hexadecimal 0's for each byte in 'old-text'.

Old-text can be in one of the following formats:

- An even-numbered string of 2 to 32 hexadecimal digits, where one pair of hexadecimal digits describes one byte in the phase.

  The same applies to modules; however, the specification must be a multiple of 4 digits.

- A string of one to sixteen characters, enclosed in quotes, where each character represents one byte in the phase.

  The same applies to modules; however, the specification must be an even number of bytes.

- A repetition factor, which is a decimal value that indicates how often the associated string of hexadecimal digits occurs in the resulting text string. This string must not exceed 32 hexadecimal digits.

  The specified repetition factor must be 2 or higher; it precedes, without intervening blanks, the associated string and must be enclosed in slashes. For example:

  /16/FF (means sixteen FF's)

new-text:
Specifies the text that is to replace the text at the specified address (see above).

New text can be in any of the formats described under old-text (see above).

If old-text is specified, new-text must have the same length (in bytes).

If new-text is specified without old-text, the colon must be specified at the beginning of the new-text line.

Example

ALTER 2034 47F0F000 : 47F0F800

COMPATIBLE

Operation	Operands
COMPAtible	WITH=(product,...)

The COMPATIBLE statement is used to indicate to MSHP at installation time those products that are compatible with the shipped product(s).

Compatible products are usually based on the same base products, contain the same components as the shipped products. For example, VSAM Space Management and VSAM Backup/Restore, both require VSE/VSAM as a base product. Compatible products may run concurrently with each other, and may also be stored in the same sublibrary.

Description of Operands

WITH = (product,...):
Specifies the name(s) of the compatible product(s).

Example

COMPATIBLE WITH = 301H07

COMPRISES

Operation	Operands
COMPRises	component [PHAses=(member-name,...)] [MODules=(member-name,...)] [MACros=(member-name,...) [TYpe=member-type]]

The COMPRISES statement is used to specify the component(s) comprised in the shipped product and the sublibrary members that make up the component(s). The information is entered in the system history file. A separate COMPRISES statement must be issued for each component contained in the shipped product.

Restriction: One COMPRISES statement may not refer to more than a total of 100 phases, modules, and macros.

Description of Operands

component:
Specifies the component comprised in the shipped product.

PHAses = (member-name,...):
Specifies the phases of the named component.

MODules = (member-name,...):
Specifies the modules of the component.

MACros = (member-name,...):
Specifies the macros of the component.

TYpe = member-type:
Indicates the type of the affected macro. The member-type can be one character only.

If the operand is omitted, type E is assumed.

Example

```
COMPRISES 5666-301-02    -
      PHASES = PHASA*    -
      MODULES = MODA*    -
      MACROS = MACA*
COMPRISES 5666-301-03    -
      PHASES = PHASB     -
      MODULES = MODB*
```

DATA

Operation	Operands
DATA	no operands

The DATA statement in conjunction with /$ delimits input that is to be passed by MSHP to the linkage editor or the librarian.

Restrictions:

- A DATA statement (with its corresponding terminating delimiter /$) may be followed only by another DATA statement, not by any other detail control statement.
- The end-of-data indicator (/$) is valid only when input is entered via SYSIPT. Substitute this delimiter by hitting END/ENTER if input is entered from the console.
- Input for the linkage editor must not contain 'named INCLUDE' statements; however, this is not checked by MSHP.

Linking from a link-book (where link-book is an object module that contains LNKEDT control statements) must be requested with the MSHP statement INVOLVES.

The sequence of delimiters and input is as follows:

DATA The initiating delimiter

Input Linkage editor or librarian statements

/$ The terminating delimiter, which must be in columns one and two in an input line.

where: Input refers to data on SYSIPT after // EXEC MSHP has been read from SYSRDR.

MSHP checks the first line after the DATA statement. If this is a linkage editor control statement, all input beginning with the statement and up to, but excluding, the next /$ or /* line, is passed unaltered and unchecked to the linkage editor. If it is a librarian control statement, MSHP passes the input in the same way to the librarian program.

The linkage editor control statements checked for are:

ACTION
ENTRY
INCLUDE
PHASE

The librarian control statement checked for is:

CATALOG

MSHP internally converts any old MAINT CATALR and CATALS statements into the CATALOG statement.

Example

DATA

DEFINE HISTORY

Operation	Operands
DEFine	History [AUXiliary|SYStem] EXTent=start-track:tracks [SPlit=split-track] [UNIT=SYSxxx] [IDentifier='file-identifier']

The DEFINE HISTORY statement is used to create extent definitions for a history file in the user label area of the partition in which MSHP is executed.

Restrictions:

- If you use IBM-supplied standard labels or if your own standard label set contains DLBL and EXTENT statements for the system history file (filename IJSYSHF), do not use DEFINE HISTORY SYSTEM in any MSHP job accessing the system history file.
- A DEFINE statement, if used, must immediately follow the applicable function control statement; it may not be placed at the end of several functions or at the end of the job stream.

Description of Operands

AUXiliary:
Specifies that an auxiliary history (work) file is to be defined. This is the default, if neither AUXiliary nor SYStem is specified.

The auxiliary history file is maintained under the filename IJSYS02 on the default logical unit SYS002. MSHP normally uses this file as a history work file. As such, the permanent DLBL and EXTENT definition that most systems contain for the IJSYS02 work file is sufficient. The DEFINE HISTORY AUXILIARY statement allows you to explicitly define a temporary auxiliary history file (in the user label area) on the logical unit indicated in the UNIT operand.

SYStem:
Specifies that the system history file is to be defined.

The system history file is part of the IBM-distributed system and is maintained under the filename IJSYSHF. To access the file, MSHP uses the IBM-set default logical unit SYSREC. However, you can use any programmer logical unit to refer to the file if you place it on a volume other than that of SYSREC.

The history file should be permanently defined. If it is to be on the SYSREC volume, supply the following statements:

```
// DLBL IJSYSHF,'A5666301.SYSTEM.HISTORY.FILE',99/365
// EXTENT SYSREC,,1,0,start-address,number-of-tracks/blocks
```

With the DEFINE HISTORY SYSTEM statement you can define a temporary system history file on the logical unit specified in the UNIT operand. This definition is, of course, valid only for the duration of the applicable MSHP job.

Since the system history file normally contains all the status information of the system, you should always keep a backup copy of it.

EXTent = start-track:tracks:
Specifies the extent information for the history file.

Start-track specifies the sequential number of the track (relative to zero) where the extent is to begin. For FBA devices, start-track indicates the block number at which the extent is to start.

Tracks specifies the number of tracks (or FBA blocks) to be allocated to the history file.

For the number of tracks or blocks required on the various types of disk volume, see *VSE/Advanced Functions Planning and Installation.*

SPlit = split-track:
Specifies, for CKD devices, which track is the last one in each cylinder to be allocated to the history file. (The first cylinder occupied by the file is the one in which the "start-track" lies, and the last cylinder is determined by the number of tracks specified.)

Split-track is a two-digit decimal integer equal to the number of tracks per cylinder minus one.

UNIT = SYSxxx:
Specifies the logical unit (other than SYSREC) on which the history file is to reside.

Defaults: If not specified, MSHP takes the following defaults:

- For a system history file: SYSREC
- For an auxiliary history file: SYS002

IDentifier = 'file-identifier':
Specifies the history file identification that is to be entered in the VTOC.

The file-identifier is a string, enclosed in quotes, of one to 44 alphameric characters.

Defaults: If the operand is not specified, MSHP takes the following defaults:

- For an auxiliary history file:

  'A5666301.AUXILIARY.HISTORY.FILE'

- For the system history file:

  'A5666301.SYSTEM.HISTORY.FILE'

Example

DEFINE HIST EXTENT = 19:96

DELETE

Operation	Operands
DELete	[from-line[+rel]] : to-line[+rel]

The DELETE statement indicates the lines to be deleted from a macro (definition) when applying a local/APAR fix.

Description of Operands

from-line:
Specifies the line-number (in columns 73 through 78 in the macro) where deletion begins. The from-line is the first line to be deleted.

Default: If omitted, from-line is assumed to be equal to the to-line value. This means that only the line designated by to-line is deleted.

+rel:
Identifies the position of the statement relative to the from-line number.

rel is an integer of one or two digits. It applies to E- or F-type macros only.

to-line:
Identifies the last line of the lines to be deleted. The value of to-line must be equal to or greater than the value given in from-line.

+rel:
Identifies the position of the statement relative to the to-line number.

rel is an integer of one or two digits. It applies to E- or F-type macros only.

Example

DELETE 000380:000400

EXCLUDE

Operation	Operands
EXCLude	{PRoduct=(product,...)| COmponent=(component[-level],...)| PTf=(ptf-number,...)}

The EXCLUDE statement is used to exclude specific products, components or PTFs from a service application (with the INSTALL SERVICE statement). This implicitly includes service for all other products, components, or PTFs shipped on the tape.

Description of Operands

PRoduct = (product,...):
Specifies the product(s) that are not to be serviced.

COmponent = (component[-level],...):
Specifies the component(s) that are not to be serviced. If level is not specified, MSHP excludes all levels of the component.

PTf = (ptf-number,...):
Lists the PTFs that are not to be installed.

Examples

EXCLUDE PRODUCT = XE8G98

EXCLUDE COMPONENT = 5666-301-01-H07

EXECUTE

Operation	Operands
EXECute	(control-program,...)[Xref|NOXref]

The EXECUTE statement is used to indicate which system programs (assembler, librarian or linkage editor) are to be called in which order to process the data submitted with the TAILOR statement. The data to be processed must immediately follow the EXECUTE statement and be terminated by /$.

Description of Operands

(control-program,...):
MSHP calls the specified system program(s) in the submitted order to process the data which immediately follows the EXECUTE statement (and is terminated by /$).

If two programs are specified, the output of the first program is taken as input to the second without any modification.

Any mismatch between the program and the data (for example, an object deck as input for ASSEMBLY) is not checked by MSHP, but results in an error situation diagnosed by the called program.

The following programs or program combinations can be specified:

```
EXEC ASSEMBLY,LNKEDT

EXEC ASSEMBLY,LIBR

EXEC LNKEDT

EXEC LIBR

EXEC ASSEMBLY
EXEC LNKEDT

EXEC ASSEMBLY
EXEC LIBR
```

MSHP internally converts any reference to the old MAINT program into a reference to the new LIBR program.

Xref:
Specifies that the cross-reference list of included macros as given by the ASSEMBLY program is to be recorded in the history file.

NOXref:
Specifies that the cross-reference list of included macros is not to be recorded in the history file.

Example

```
EXECUTE (ASSEMBLY,LNKEDT)
```

GENERATE

Operation	Operands
GENerate	{PHAse=member-name| MODule=member-name| MACro=member-name [TYpe=member-type]}

The GENERATE statement is used as a detail control statement to the SELECT statement to regenerate (retailor) individual phases, modules, or macros from the generation file.

Description of Operands

PHAse = member-name:
Indicates to MSHP the name of the phase that is to be regenerated.

MODule = member-name:
Indicates to MSHP the name of the module that is to be regenerated.

MACro = member-name:
Indicates to MSHP the name of the macro that is to be regenerated.

TYpe = member-type:
Indicates the type of the affected macro. The member-type can be one character only.

If the operand is omitted, type E is assumed.

Example

GENERATE MACRO = BMS030 SUBLIB = A

INCLUDE

Operation	Operands
INCLude	{PRoduct=(product,...)| COmponent=(component[-level],...)| PTf=(ptf-number,...)}

The INCLUDE statement is used to indicate to MSHP that only the named products, components, or PTFs are to be included in a service application (with INSTALL SERVICE). This implicitly excludes service for all other products, components, or PTFs shipped on the service tape.

Description of Operands

PRoduct=(product,...):
Specifies the product(s) to which service is to be applied.

COmponent=(component[-level],...):
Specifies the component(s) to which service is to be applied.

If level is omitted, all levels of the component are serviced.

PTf=(ptf-number,...):
Lists the PTFs that are to be applied.

Examples

INCLUDE PRODUCT=301H07

INCLUDE COMPONENT=5666-301-01-H07

INFLUENCES

Operation	Operands
INFluences	{PHAse=(member-name,...) | MODule=(member-name,...) | MACro=(member-name,...) | [TYpe=member-type]}

The INFLUENCES statement identifies which generated phases, modules, or macros of the serviced component are affected by a PTF or local/APAR fix and have to be regenerated.

Restriction: One INFLUENCES statement may not refer to more than a total of 100 phases, modules, and/or macros.

Description of Operands

PHAse = (member-name,...):
Names the phases to be regenerated.

MODule = (member-name,...):
Names the modules to be regenerated.

MACro = (member-name,...):
Names the macros to be regenerated.

TYpe = member-type:
Indicates the type of the affected macro. The member-type can be one character only.

If the operand is omitted, type E is assumed.

Example

INFLUENCES PHASE = DFHFCP*

INSERT

Operation	Operands
INsert	after-line[+rel]

The INSERT statement identifies where, in a source book (macro), additions are to be made when archiving a local/APAR fix or when initiating a local or APAR fix by means of the CORRECT statement. The statement further serves as the initiating delimiter A /$ (on SYSIPT) or a blank line (at the console) is the terminating delimiter. for the input line to be inserted.

Description of Operands

after-line:
Specifies the line number in the macro (in columns 73 through 78) after which the source input (following the INSERT statement up to the next /$) is to be inserted.

after-line is an integer of one to six digits. If fewer than six digits are coded, leading zeros are supplied.

+rel:
Specifies the position of the source input relative to the after-line number.

rel is an integer of one or two digits.

Restriction: rel applies to E- or F-type macros only.

Example

INSERT 7100

INVOLVES

Operation	Operands
INVolves	LINK=(link-book,...)

The INVOLVES statement explicitly requests link-editing to be performed when installing an archived product, or when applying PTFs from a service tape.

As a detail control statement to APPLY, INCORPORATE, and CORRECT, it indicates that, as the final step of the particular function, a link-edit run must be performed.

Description of Operands

LINK = (link-book,...):
Link-book specifies the name of a module that is to be included in the link-edit step.

Link-book is a string of one to eight characters, the first one of which must *not* be an asterisk.

If you specify several link-books, the linkage editor includes the named modules in the same sequence as they occur in the list. You can specify up to 100 link-books. The linkage editor is called for each link-book specified.

Example

INVOLVES LINK = IJWIND

OR

Operation	Operands
OR	no operand

The OR statement initiates a set of alternative REQUIRES statements that are to be checked in case the preceding set of requirements is not met.

Two or more REQUIRES statements following each other immediately are considered to be in an 'AND' relation. This means that the REQUIRES check is successful only if the prerequisites, corequisites, and negative prerequisites of the whole set of REQUIRES statements are met.

Example

```
REQ PRE=XE8G98
OR
REQ PRE=XE8E24
```

PTF

Operation	Operands
PTF	=(ptf-number,...)

The PTF statement is used as a detail control statement to the LIST SERVICETAPE COVER statement to print selected cover letters.

Description of Operands

=ptf-number:
Identifies the PTF whose cover letter is to be printed.

Example

PTF=(UD34634,UD38476)

REPLACE

Operation	Operands
REPlace	[from-line[+rel]] : to-line[+rel]

The REPLACE statement is used when applying (CORRECT) or archiving (ARCHIVE) a local or APAR fix to define where replacement of lines in a source macro must begin and end.

The replacing data must follow immediately the REPLACE statement and is to be terminated by an input line containing /$ in columns 1 and 2 (or by a blank line when entered from SYSLOG).

Description of Operands

from-line:
Specifies, by the line-number in columns 73 through 78 in the macro, the first line to be deleted and to be replaced by the first (if any) input line. (Input refers to data that follows the REPLACE statement.)

Default: If from-line is not specified, it is assumed to be equal to to-line. In that case, only this line (the one designated by to-line) is replaced in the macro. It may, however, be replaced with more than one line of replacing data.

+rel:
Specifies the position of the line relative to the from-line number.

rel is an integer of one or two digits.

Restriction: rel applies to E- or F-type macros only.

to-line:
Specifies that, beginning with from-line, all lines in the macro are to be deleted up to and including the line indicated by to-line. to-line is the line-number contained in columns 73 through 78 of the macro to be modified.

+rel:
Specifies the position of the line relative to the to-line number.

rel is an integer of one or two digits.

Restriction: rel applies to E- or F-type macros only.

Example

REPLACE 212400 : 212420

REQUIRES

Operation	Operands
REQuires	[component] [PRE=req-list] [CO=req-list] [NOT=req-list]

The REQUIRES statement is used to specify the requirements (such as prerequisite PTFs) that must be met to successfully install a shipment package or apply service in PTF or local/APAR fix format. The specified requirements are entered in the history file that accompanies the programming package.

Restrictions:

1. The number of requirements per PTF, local/APAR fix, component, or product specified in one or more REQUIRES statements must not exceed 88.

2. At least one of the operands PRE=, CO=, or NOT= must be present.

You may connect several requirements (with an 'AND' relation) by specifying several REQuires statements in succession. This means that the REQUIRES check is successful only if the requisites of *all* the REQUIRES statements are met.

You can also delimit such a set of REQUIRES statements from a preceding set by means of the OR statement. If the preceding set of requirements (at least one) fails, MSHP tests the set of requirements initiated by OR. If that test is successful, all the requirements are considered to be met.

Description of Operands

component:
If the requirements specified in the req-lists are PTFs or local/APAR fixes, then component specifies the component to which the PTF or local/APAR fix belongs.

Default: If component is omitted, then the PTFs or local/APAR fix(es) specified as requirements are assumed to belong to the component to which the "requiring" PTF or local/APAR fix is applied.

Restrictions: Component must not be specified if the requirement in a req-list is neither a PTF nor a local/APAR fix. Component must always be specified if REQUIRES is used in conjunction with the ARCHIVE statement.

PRE:
Indicates that the requirements specified in the req-list have to be installed *prior to* the requested service application or installation function.

A prerequisite condition is also considered as being met if a prerequisite PTF has been superseded by another, installed PTF.

CO:
Indicates that the requirements specified in the req-list have to be applied *together with* the requested service application or installation function.

If REQUIRES is used as a detail control statement to CORRECT, CO= indicates that the requesting local/APAR fix will be applied even though the requirements specified in the req-list are not met; however, MSHP issues a warning message.

NOT:
Indicates that the requirements specified in the req-list must *not* be installed prior to the requesting service application or installation function.

req-list:
A 'req' is one of the following:

- PTF number or local/APAR fix number
- component[-level]
- product (or old feature number)

Restriction: In a requirements list, all items must be of the same type: PTF numbers, APAR numbers, components, and products may not be mixed.

Examples

REQUIRES 5666-301-01 PRE=DY48376
REQUIRES PRE=301H07
REQUIRES CO=DY23540
REQUIRES NOT=5666-301-01-H07

RESOLVES

Operation	Operands
RESolves	['comment'][APARs=(apar-number,...)]

The RESOLVES statement associates a comment with a product, a PTF, a local/APAR fix, or a generated member; it is also used to indicate which APARs are fixed by a PTF.

Restriction: Only one comment per associated product (or fix or member) can be recorded in the history file. If more than one RESOLVES 'comment' statement is specified, the last one will be recorded.

Description of Operands

'comment':
Specifies that a comment relating to a PTF, a local/APAR fix, a product, or a generated member is to be inserted in the history file.

comment is a string of characters enclosed in quotes. The maximum length of the string is 35 characters if the comment is associated with a local/APAR fix; it is 57 for any other comment.

APARs=(apar-number,...):
Specifies the APAR number(s) corrected by a given PTF.

Restriction: This operand must be specified in a RESOLVES statement that relates to a PTF (RESOLVES being used in conjunction with ARCHIVE PTF and APPLY component:ptf-number).

Example

RESOLVES 'INIT DISK ERROR' APAR=DY45000

RESTART

Operation	Operands
RESTart	restart-line[+rel]

The RESTART statement is used for the correction of edited macros (with the CORRECT statement). It indicates that a new sequence number series starts after the specified statement.

Description of Operands

restart-line:
Specifies the sequence number of the statement after which the new sequence number series starts.

+rel:
Specifies the position of the desired statement relative to 'restart-line'.

Example

RESTART 000850

SCAN

Operation	Operands
SCan	[offset] [ARGument={'char-string'|hex-string}]

The SCAN statement is used when correcting a phase (after AFFECTS PHASES=...) to search for a specified string in a phase and to display 16 bytes of the phase.

Description of Operands

offset:
Specifies the displacement (relative to the beginning of the phase) where, in the phase, the search for the specified ARGUMENT string is to be started. If the ARGUMENT=string operand is omitted, MSHP displays 16 bytes of the phase, starting at 'offset'. 'offset' is a number of up to six hexadecimal digits; leading zeros may be omitted.

ARGument={'char-string'|hex-string}:
Specifies the string that is to be searched in the phase. It can be in one of the following formats:

- A string of 1 to 16 characters, enclosed in quotes, where each character represents one byte in the phase.
- An even string of 2 to 32 hexadecimal digits, where each pair of hexadecimal digits describes one byte in the phase.

The following table shows the results of specifying the two operands 'offset' and 'ARGUMENT' in various combinations.

	Offset	ARG	Result
First scan after AFFECTS PHASES	-	-	Invalid; error message.
	-	x	Scanning for specified string from offset 0.
	x	-	Display of 16 bytes from specified offset.
	x	x	Scanning for indicated string from specified offset.
Subsequent scan	-	-	Scanning from current offset for old argument string, which must be known from preceding scan request.
	-	x	Scanning for specified string from current offset.
	x	-	Display of 16 bytes from specified offset.
	x	x	Scanning for indicated string from specified offset.

Example

SCAN 0040 ARG=47500000

SUPERSEDES

Operation	Operands
SUPersedes	(ptf-number,...)

The SUPERSEDES statement identifies which PTFs are superseded by a given PTF when that PTF is being built.

MSHP requires the list of superseded PTFs to be complete. For example: If PTF2 supersedes PTF1, and subsequently a PTF3 is issued that supersedes PTF2, then PTF3 must be specified as also superseding PTF1.

Description of Operands

(ptf-number,...):
Specifies the PTF(s) that are superseded.

The superseded ptf-numbers are recorded in the history file entry for the superseding PTF.

The superseded PTF itself, if on the system, is marked in the history file entry as superseded.

Restriction: The maximum number of PTFs that can be specified as superseded in *one* SUPERSEDES statement is 100. The *total* number that can be specified is 255.

Example

SUPERSEDES (UD38765,UD37645)

VERIFY

Operation	Operands
VERify	verify-line[+rel]

The VERIFY statement applies to source members of types E and F only. It designates where a verification is to be made for a local or APAR fix correction. The VERIFY statement must be followed by a single line of text. MSHP checks whether this text is present in the statement indicated by 'verify-line'.

Description of Operands

verify-line:
Specifies the sequence number of the source statement to be verified.

verify-line is an integer of one to six decimal digits. If fewer than six digits are coded, leading zeros are supplied.

rel:
Specifies the position of the desired statement in relation to the statement number indicated for verify-line.

rel is an integer of one or two digits.

Example

VERIFY 007100+25

IBM VSE/Advanced Functions
Maintain System History
Program Reference
Order No. SC33-6199-02

READER'S COMMENT FORM

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

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.*

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address: _____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

SC33-6199-02

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | **Please Do Not Staple** | Fold and Tape

No Postage necessary if mailed in the United States

BUSINESS REPLY MAIL

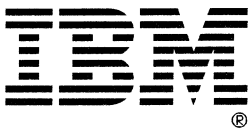
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

Postage will be paid by addressee:

International Business Machines Corp.
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold and Tape | **Please Do Not Staple** | Fold and Tape

IBM®



IBM Virtual Storage Extended/
Advanced Functions
Order Number SC33-6199-02
(File No. S370/4300-34)
Printed in U.S.A.

SC33-6199-02