



ChangeMan SSM

Detection and Synchronization Guide

© Copyright 2001-2020 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contains Confidential Information. Except as specifically indicated otherwise, a valid license is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Product version: 8.4 (October 2013)

Publication date: October 2020 (rebranded only)

Contents

	Welcome to ChangeMan SSM	9
	Manual Organization	10
	Related Documents	11
	Using the Manuals	11
	Online Help	12
<i>Chapter 1</i>	Introduction	13
	Fingerprinting Technology	14
	Detection and Synchronization Component	14
	Functions for HFS Files	16
<i>Chapter 2</i>	Online Detection and Synchronization	17
	Logging into ChangeMan SSM	18
	Field Level Help	18
	Modifying User Options	19
	Default Options	20
	Detection and Synchronization for MVS	20
	Select a Function	21
	Option (0) - Masking and Filtering Options	22
	Specify Source or Text Mask Rules (Option 1)	23
	Specify Load Module and IMS Mask Rules (Option 2)	25
	View Mask Rules in Effect (Option 3)	26
	Update Excluded Data Set List for Synchronization (Option 4)	26
	Option (1) - Create Fingerprint Data Set	28
	Specify Environment	29
	Resolve Data Sets	30
	Fingerprint Data Set Pattern List	30
	Option (2) - Review Changes	31
	Specifying a Previously-Generated FGP Data Set	32
	Select Fingerprint Data Set Name Pattern	33
	Select Fingerprint Data Set Name	34
	Verifying the Fingerprint Data Set Name	34
	Interpret Results	35
	Option (3) - Package Changes	36
	Enter Data Set Names	36
	Create Change Basket Verification	38
	Option (4) - Package Data Sets	38
	Specify Data Sets	39
	Resolve the Data Sets	40
	Create Basket Verification	41
	Option (5) - View Basket	42
	Option (6) - Applying a Basket	43

Option (7) - View Fingerprints	45
Option (8) - Detect Changes	48
Option (9) - Package Changes	49
Package Changes	49
Create Basket Verification	51
Option (G) - Generate a Fingerprint Token	52
Fingerprint File or Member	53
Interpret Fingerprint Results.	54
Option (C) - Sync Checking and Comparing Environments	55
Specify Base Environment	55
Resolve Data Sets in Base	58
Specify the Target Environment	59
Resolve Data Sets in Target	60
Results of Synchronization Verification	61
View Comparison Report	62
Print Comparison Report	63
Option (P) - PDS Compare	64
Option (S) - Synchronizing Environments	65
Specify Base Environment	66
Specify Target Environment	67
Optional Change Basket Specification	68
Option (A) - Advanced Operations	70
Option (1) - Data Set Compare	71
Option (2) - Group Compare	73
Option (3) - Volume Compare.	79
Option (4) - Fingerprinting	81
Option (5) - Creating a Basket	83
Option (6) - Importing a Change Basket	86
Option (7) - Undo an Import.	89
Detection and Synchronization for HFS.	90
Option (1) - Fingerprint	91
Option (2) - Compare	92
Option (3) - Create Basket	93
Option (4) - Import Basket.	95
Option (5) - Undo Import.	96
Option (6) - View Basket	97
Option (7) - View Fingerprint	98
Chapter 3	
Batch Detection and Synchronization	99
Environment Comparison	100
Remote Environments and Firewalls	101
Scope and Size of Environment.	101
Compare Environments with Mask	102
Load Module Masking.	102
Application Control Block Masking (ACB)	103
Message Format Services Masking.	103
Group Masking	103
Search Strings and Masking	104

Area Masking	105
Language Masking	106
General Masking	106
Group Masking	107
Fingerprint Data Set Safeguards	107
Performance	108
Environment Synchronization	109
Change Basket	110
Transferring the Change Basket	110
Redundancy Management	111
Fingerprint Token Data Set	111
DCB Information	111

Chapter 4

Commands	113
Command Structure	114
Command Summary	114
Command Stacking	117
Reserved DDs	118
Comments	118
Using REXX Programs	119
Synchronization Commands	119
AUTORECALL Command	119
BASE Command	120
CLOCK Command	134
COMPARE Command	135
DASDRED Command	148
ENQUEUE Command	151
EXPORT Command	151
EXTENDED SYNC Command	153
EXTRACT Command	154
IDRLOAD Command	164
IMPORT Command	164
LIST Command	172
MASK Command	174
MODEL Command	183
NOECHO Command	187
ONLINE Command	188
ORPHANCHECK Command	188
REFTARG Command	189
RESTART Command	190
SETPRINT Command	191
SIMULATE Command	196
SMSMAP Command	196
TARGET Command	198
TRACE Command	198
VIEWFGP Command	198
VIEWBASKET Command	199
VOLMAP Command	200

HFS Synchronization Commands	201
Limitations	201
Security	201
Using Pattern Names	201
HFSBASE Command	202
HFSCOMPARE Command	204
HFSEXPOR Command	205
HFSEXTRACT Command	206
HFSIMPORT Command	207
HFSMODEL Command	208
HFSTARGET Command	209

Chapter 5

Utilities	211
Utilities Menu Options.	212
Transferring Files	213
Option (1) - Compress	213
Option (2) - FTP to a Remote Host.	214
Option (3) - Expand	216
Option (4) - Snapshot	217
Option (5) - Snapshot Comparison.	219
Option (6) - Convert VSAM Basket	220
Option (7) - Redundancy Detection	221
Option (D) - Demo.	224
Option (H) - HFS Demo	226

Chapter 6

Redundancy Detection and Management.	229
Tokenizing Volumes	230
Restart Tokenizing	231
Determine Redundancy Level	231
Report Redundant Data Sets and Members	233
Redundancy Report Examples	233
Data Set Redundancy Report	234
Member Redundancy Report.	235

Chapter 7

Snapshot Facility	237
Snapshot Process.	238
Compare Snapfiles - Mismatched Data Sets	239
Compare Snap Files - Last Referenced Date	241

Appendix A

Wildcards, Symbolics, and DCB Attributes	243
DCB Attributes	243
System Symbolics	244
Using Symbolics in Command Parameters	244
Using Symbolics in HPSPARMS Parameters.	244
Wildcards	245
Data Set, Object, and Cluster Name Wildcards	245
Member Name Wildcards	249
Volume Name Wildcards.	250

Appendix B

Transfer Change Baskets 251

Convert and Download Basket. 251

 Allocate an XMIT data set. 251

 Unload your Change Basket data set to a transmittable XMIT format. 251

 Define a New Transfer Type Called BIN80 251

 Download the 'your_userid.XMIT' file to PC. 252

Upload and Apply a Basket to a Target MVS System 252

 Copy Change Basket File to PC 252

 Allocate a XMIT data set on the Target MVS 252

 Define or Add a New Transfer Type: BIN80 252

 Upload the Change Basket PC file to MVS 253

 Issue the TSO RECEIVE command. 253

 Use the Change Basket Data Set as Input 253

Glossary 255

Index. 261

Welcome to ChangeMan SSM

This document describes ChangeMan[®] SSM (System Software Manager for z/OS), a product of Serena Software, Inc., a Micro Focus company.

ChangeMan SSM detects and tracks changes within groups of applications or system data sets, synchronizes local or remote operating environments, and restores environments to a clean and working state. ChangeMan SSM detects changes to data sets, PDS members, or full DASD volumes using a unique fingerprinting technology.

Before You Begin See the Readme for the latest updates and corrections for this manual. You can access the Readme from the SupportLine website at <https://supportline.microfocus.com/>.

- Objective** This manual provides information to help you:
- Understand the ChangeMan SSM Detection and Synchronization component.
 - Use the Online Detection and Synchronization facility.
 - Use the Batch Detection and Synchronization facility.
 - Use the Online Utility functions.
 - Use the Redundancy Detection functions.
 - Use the Snapshot volume comparison functions.

Audience This manual is intended for anyone responsible for detecting and managing change between different environments, data sets, and other components.

Change Bars Change bars in the left margin identify changes for SSM 8.4.

Manual Organization

The following table describes the chapters in the *ChangeMan SSM Detection and Synchronization Guide*.

Chapter	Description
1	Introduction. Explains the Detection and Synchronization component of ChangeMan SSM.
2	Online Change Detection and Synchronization. Explains how to use the Change Detection online panels. A detailed description of each parameter displayed in the panels is provided.
3	Batch Change Detection and Synchronization. Explains how to use the Change Detection batch commands. Command structure, stacking, and parameter descriptions are provided.
4	Commands. Describes the Detection and Synchronization commands, parameters, and sub-parameters.
5	Utilities. Explains how to use the Utilities provided with the Detection and Synchronization component.
6	Redundancy Detection and Management. Explains how to use Redundancy Detection.
7	Snapshot Facility. Explains how to use the Snapshot Facility.
Appendix A	Describes the following: <ul style="list-style-type: none"> ■ DCB attributes of files used in synchronization. ■ System symbolics used in select parameters in the batch synchronization facility. ■ Wildcards used in online and batch synchronization facilities.
Appendix B	How to package and apply a change basket via e-mail.
Glossary	Defines terms used within ChangeMan SSM.
Index	Index of subjects.

Related Documents

The following manuals are available in Adobe Acrobat PDF format on the SupportLine website at <https://supportline.microfocus.com/>.

Title	Description
ChangeMan SSM Change Tracking User's Guide	Instructions for using the Change Tracking Component.
ChangeMan SSM Detection and Synchronization User's Guide	Describes the Detection and Synchronization component of ChangeMan SSM.
ChangeMan SSM Data Extraction User's Guide	Instructions for using the Data Extraction Component.
ChangeMan SSM Messages Guide	A listing of ChangeMan SSM error and information messages and their meaning.
ChangeMan SSM Getting Started Guide	Provides: <ul style="list-style-type: none"> ■ An overview of the ChangeMan SSM components. ■ Instructions for setting up your environment to get started quickly. ■ Instructions for performing the most commonly used functions.
ChangeMan SSM Installation Guide	Provides instructions for installing the three components: <ul style="list-style-type: none"> ■ Change Tracking ■ Detection and Synchronization ■ Data Extraction.

Using the Manuals

The manuals use Adobe Portable Document Format (PDF). To view PDF files, use Adobe® Reader®, which is freely available from www.adobe.com.



TIP Be sure to download the *full version* of Reader. The more basic version does not include the search feature.

PDF Document Features

This section highlights some of the main Reader features. For more detailed information, see the Adobe Reader online help system.

The PDF manuals include the following features:

- **Bookmarks.** All of the manuals contain predefined bookmarks that make it easy for you to quickly jump to a specific topic. By default, the bookmarks appear to the left of each online manual.
- **Links.** Cross-reference links within a manual enable you to jump to other sections within the manual and to other manuals with a single mouse click. These links appear in blue.

- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.
- **Advanced search.** Starting with version 6, Adobe Reader includes an advanced search feature that enables you to search across multiple PDF files in a specified directory. (This is in addition to using any search index created by Adobe Catalog—see step 3 below.)

Multiple Document Search

Adobe Reader Version 6 and higher supports multiple document search. To search across multiple PDF documents concurrently, perform the following steps:

- 1 In Adobe Reader, select Edit > Search (or press CTRL+F).
- 2 In the text box, enter the keywords for which you want to search.
- 3 Select the **All PDF Documents in** option, then browse to select the folder in which you want to search. (If you have a document open that has an Adobe Catalog index attached, you can leave the **In the index named...** option selected to search across all the manuals in the index.)
- 4 Optionally, select one or more of the additional search options, such as **Whole words only** or **Case-Sensitive**.
- 5 Click the **Search** button.

For more information, see Adobe Reader's online help.

Online Help

ChangeMan SSM includes an online help system that offers the following:

Attribute	Description
Guidance Information	Press F1 in the command field to display an overview of the current panel.
Index	A list of key words from which you can access detailed information on each subject. Press F1 from the main menu of each of the components to access the menu.
Field Sensitive Help	Field sensitive help is available for every field on the ISPF panels. Place the cursor on the field and press F1 for help.

Chapter 1

Introduction

ChangeMan SSM is a comprehensive suite of change management solutions. Together, these solutions assist all organizational personnel in detecting, tracking, reporting, implementing, distributing, and recovering from changes to the MVS environment. ChangeMan SSM provides solutions for both small data centers as well as large, complex environments with multiple LPARs, CPUs, and sites.

ChangeMan SSM also assists with data center consolidations and mergers/acquisitions.

ChangeMan SSM consists of three components:

- Change Tracking
- Detection and Synchronization
- Data Extraction

These components provide you with toolsets to detect changes and synchronize environments at the data set, PDS member, and VSAM record level.

Fingerprinting Technology	14
Detection and Synchronization Component	14

Fingerprinting Technology

All ChangeMan SSM components use the Fingerprinting technology. Like DNA mapping that uniquely identifies a human, Fingerprint technology uniquely identifies a file by creating a unique 8 byte *token* representing the contents (*the content Fingerprint*) of any PDS member or data set. When the contents of a data set or member change, so does its token. As long as the contents of the data set or member do not change, the generated token will not change. This means that a data set can be copied to another volume, distributed to a remote site, or reblocked without incurring a change to its token.

ChangeMan SSM uses Fingerprint technology to detect differences in the contents of one or more members, an entire data set, a group of data sets, a volume, or group of volumes regardless of the naming conventions. For partitioned data sets, one token is generated for each member, and one for the entire PDS.

For VSAM KSDS clusters, ChangeMan SSM generates one token per record, and one composite token for the entire data component. These tokens are stored externally in a Fingerprint data set. The VSAM Fingerprint representations are evaluated to determine the changed records. By comparing the current VSAM Fingerprint data set with a previous one, the added, updated, and deleted records are identified. To synchronize VSAM clusters, ChangeMan SSM creates a VSAM Change Basket containing only the added, deleted, and changed records. The Change Basket, created in a Base environment, is transferred to local or remote environments for Import processing to synchronize the Target cluster.

Flat files and flattened databases can also be Fingerprinted at the record level. Flat files can be Fingerprinted as either KEYED or TEXT format. A Fingerprint token is produced for each record of the flat file. The record token is a six-byte representation of the content of a record. If the flat file has a user specified key, then the token is the six-byte representation plus the key.

PDS member-level directory tokens can optionally be used as an additional determination of change (when a change occurs in the PDS directory during the comparison and synchronization processes). Refer to the EXTENDED SYNC command in the *ChangeMan SSM Detection and Synchronization Guide*.

Detection and Synchronization Component

This component determines whether two environments, data sets, groups of data sets, or volumes (local or remote), differ. The following types of data sets are supported:

- Partitioned (PDS and PDSE)
- Physical Sequential (PS)
- Extended Sequential (PS)
- Direct Access (DA)

You can perform the following tasks with the Detection and Synchronization component:

- Compare local or remote software environments.
- Detect changes in local or remote software environments.
- Verify synchronization between local and remote software environments.

- Report byte-level differences in large environments, passing on only the changed PDS members to ChangeMan SSM's side-by-side compare facility, COMPAREX, or IBM's SuperC.
- Verify that current source and production load modules match.
- Ensure that software has been properly distributed and installed.
- Maintain all LPARs from a central location.
- Package software changes in a single physical sequential file (Change Basket).
- Save Resources by transmitting only the changes to other sites using FTP, Connect:Direct, or your preferred software.
- Apply Change Baskets using the IMPORT command.
- Verify the proper installation of the change package.
- Facilitate data center consolidation and migration.
- Synchronize multiple LPARs, operating environments, and software packages in local or remote CPUs.
- Eliminate redundant data sets and members (by content) in either a group of concatenated data sets, a pool of volumes, or the entire data center.
- Avoid full volume and data set transmissions from one location to another by transferring only the changed components.
- Identify mismatched/missing data sets, DCB differences, directory differences, and SPACE differences between environments.

ChangeMan SSM creates fingerprint data sets that consist of tokens (called base and target fingerprints) representing the contents of each environment. Evaluating these tokens reveals any differences between the data sets in each base and target environment. This process also identifies missing data sets, missing members, and newly-allocated data sets. The Detection and Synchronization component can then synchronize the two environments (base and target) to make them identical.

To synchronize the data sets, the Detection and Synchronization component creates a change basket (a single PS data set) that contains all of the data components required to make the two environments identical. The change basket can be transferred (using NDM, FTP, or other communication software), and imported into one or more target environments.

Optionally, a reverse change basket can be generated during the import process. This enables a backup of the environment components affected by the APPLY process, thereby automating the back-out process (if necessary).

By using the fingerprint tokens, the Detection and Synchronization component can compare any two environments, regardless of size or naming conventions.

To detect changes over a period of time, a single previously-generated fingerprint data set may be referenced to provide the environment. This fingerprint data set represents the "before" change environment, while the current data sets on DASD are re-fingerprinted to represent the current environment. This is performed by the REFTARG command.

The Detection and Synchronization component provides extensive masking capabilities that enable you to look for the actual code differences in source/load modules, rather than search for superficial differences (such as an embedded date or time). In conversion projects, large groups of data sets can be scanned to locate a user-specified string.

Masking is ignored during synchronization operations to ensure that the synchronized environments are identical.

Functions for HFS Files

For HFS files, you can compare local or remote software environments, detect changes, and verify synchronization. You can perform the following tasks for HFS files:

- Fingerprint a group of HFS files.
- Compare two HFS fingerprint data sets and report differences.
- Capture changes into an HFS change basket.
- Apply an HFS change basket to synchronize environments.
- Use path modeling to compare files in different locations, as well as update a location with a different path.

Chapter 2

Online Detection and Synchronization

The *ChangeMan SSM* Detection and Synchronization component provides online access to the change detection and synchronization functions. This chapter describes each of those functions, and includes step-by-step instructions for using them.

You invoke the functions by selecting options from panels.

Some of the functions generate JCL which, when executed in batch, complete the selected function. JCL is generated when the scope of the function is not feasible as an online task. Other functions execute in the foreground, and receive results immediately.

The Detection and Synchronization component is available to any user with a valid TSO userid; you do not need ChangeMan SSM authorization to access the Detection and Synchronization functions. Granted authority is required only for access to the Change Tracking component.

Logging into ChangeMan SSM	18
Field Level Help	18
Modifying User Options	19
Default Options	20
Detection and Synchronization for MVS	20
Select a Function	21
Option (0) - Masking and Filtering Options	22
Option (1) - Create Fingerprint Data Set	28
Option (2) - Review Changes	31
Option (3) - Package Changes	36
Option (4) - Package Data Sets	38
Option (5) - View Basket	42
Option (6) - Applying a Basket	43
Option (7) - View Fingerprints	45
Option (8) - Detect Changes	48
Option (9) - Package Changes	49
Option (G) - Generate a Fingerprint Token	52
Option (C) - Sync Checking and Comparing Environments	55
Option (P) - PDS Compare	64
Option (S) - Synchronizing Environments	65
Option (A) - Advanced Operations	70
Detection and Synchronization for HFS	90

Logging into ChangeMan SSM

When logging into *ChangeMan SSM*, the ChangeMan SSM Main Menu is the first menu to display.

```
HPS@PRI1 ---- ChangeMan SSM Main Menu (Ver 8.4.0)   Time: 13:49:14 -----
Option ==>

Select one of the highlighted functions:

0 Options                - Review or Modify Parameters

A Administration         - Change Tracking Administrator Functions
C Change Tracking        - History, Problem Determination, Recovery
Z HFS Change Tracking    - HFS History, Problem Determination

S Detection and Synchronization - Detect Changes and Synchronize MVS data
H HFS - Detect and Synchronize - Detect Changes and Synchronize HFS data
D Data Extraction        - Extract Changed Data in KSDS, other VSAM,
                        PS or DA files and Flattened Databases
U Utilities              - Utilities and Self-guided Product Demos

X Exit

                        ChangeMan SSM
                        Copyright (c) 1999-2013 - Serena Software, Inc.
                        All Rights Reserved.

Press END to Exit
```

The options described in this chapter are:

- Option 0 - Review default parameters and modify user options.
- Option S - Perform Detection and Synchronization functions for MVS data.
- Option H - Perform Detection and Synchronization functions for HFS data.

Field Level Help

Contextual help is provided for every field in the ChangeMan SSM ISPF Interface. To access the help panels, position the cursor in the desired field, then press PF1. Pressing PF1 at the command line provides general help for the panel.

Panel Navigation

To exit ChangeMan SSM at any time, press PF4.

To return to the previous panel from any of the subordinate panels, press End, or type X and press Enter.

Modifying User Options

You must specify your user options before generating JCL through the ISPF panels. If these are not specified, the first time you attempt to generate and save JCL through an ISPF panel, it will fail.

To set up or modify your user options, choose Option 0 from the ChangeMan SSM Main Menu. The Defaults and User Options panel displays.

```
HPSUOPTN ----- ChangeMan SSM Defaults and User Options (8.4.0.00) -----
Command ==>

The Administrator can modify the defaults by editing the member HPSPARMS
in the HPSPLIB library: XYZPRD.SSM834U.PARMLIB

Defaults in effect:
SSM Site Master           ==> XYZPRD.SSM83U.SITE.MASTER
SSM Object Master        ==> XYZPRD.SSM83U.OBJECT.MASTER
SSM Delta Master         ==> XYZPRD.SSM83U.DELTA.MASTER
Print Utility            ==> VPSPRINT
Allowed Number of Nodes for temporary SMS data sets: ==> 4
Allocation Parameters for temporary SMS data sets:   ==> TRK,(120,060)
User Defaults to be set:
Report Disposition Panel ==> Y           Y-Display panel, N-Delete report
Comparison Utility       ==> BUILT-IN   BUILT-IN/COMPAREX/SUPERC
Your JCL Library         ==> SSM834.JCL
JOB Card Statements:
==> //USER001 JOB (Account),'ChangeMan SSM',
==> //          CLASS=A,NOTIFY=USER001,MSGCLASS=X
==> //*
==> /*
```

The options you can modify are on the lower part of the panel under User's Options to be Set. They are described in the following table:

Option	Description
Report Disposition Panel	Specify if the intermediate panel (HPSUREP) is to be displayed. The default is Y (display the panel). If N is specified, the temporary outlist data sets allocated during the session are deleted automatically.
Comparison Utility	Specify your comparison utility: BUILT-IN, COMPAREX, or SUPERC.
Your JCL Library	Specify your personal JCL library to hold the batch JCL generated by the ISPF panels. This data set must be pre-allocated.
JOBCARD	The job cards specified here will be used as the JOBCARD in the generated JCL.

Default Options

The parameters displayed under Defaults in Effect cannot be modified on this panel. An administrator can change these defaults by editing the HPSPARMS member. This procedure is documented in the *ChangeMan SSM Installation Guide*.

Option	Description
SSM Site Master	Name of the main control file.
SSM Object Master	Name of the main Object file.
SSM Delta Master	Name of the database where backup versions of members in delta-tracked data sets are stored.
Print Utility	Name of the print utility used by the Comparex interface to print the Differences Report.
Allowed Number of Nodes	Number of nodes allowed for internally-allocated data sets.
Allocation Parameters	Allocation size for the temporary report data sets.
Preferred Comparison Utility	Name of the comparison utility that ChangeMan SSM will interface with to produce the Differences report. The available choices are: Comparex, ChangeMan SSM, and SuperC.

Detection and Synchronization for MVS

Choose option **S** (Detection and Synchronization) from the ChangeMan SSM Main Menu to display the Detection and Synchronization Menu.

```

HPSMENUT ----- Detection and Synchronization Menu -----
Option ==>

0  Masking and Filtering - Detection masking and Synchronization filtering

1  Fingerprint           - Fingerprint an Environment
2  Review Changes       - Detect changes to a Fingerprinted Environment
3  Package Changes *    - Changes to an Environment; Current vs. FGP
4  Package Data Sets *  - Export full data sets into a Basket
5  View Basket          - Display the list of data sets and their actions
6  Apply Basket *       - Apply changes to the Target Environment

7  View Fingerprint     - Display data set names, attributes, and members
8  Detect Changes       - Compare the Fingerprints of two Environments
9  Package Changes *    - Changes to an Environment; FGP vs. FGP

G  Generate Token       - Fingerprint Token for a single file or PDS member
C  Check Synchrony     - Compare selected members in Environment(s)
P  PDS Compare         - Produce a MEMTOMEM compare report for two PDS's
S  Synchronize *       - Two local Environments, or just save a Basket

A  Advanced Operations - Batch jobs suitable for large Environments
X  Exit                - Press END to Exit

```

Select a Function

Use this procedure to select a function from the Detection and Synchronization Menu.

- 1 In the Option field, type one of the following characters for the function you want to perform.

Character	Function
0	<p>Displays the Masking and Filtering menu where you can choose sub-options to perform the following functions:</p> <ul style="list-style-type: none"> ■ Define the MASK rules for your current ISPF session. These rules are retained in your profile until changed. Mask rules can be defined for fingerprinting or change detection in source or LOAD modules. MASK rules are ignored for synchronization functions. ■ Update your excluded data set list for synchronization. <p>See "Option (0) - Masking and Filtering Options" on page 22 for additional detail.</p>
1	Fingerprint a data set, group of data sets, or a software environment.
2	Review Changes. Detect changes to a fingerprinted environment. One fingerprint data set represents the environment in the past. Changes to the current data sets referenced from the fingerprint file are detected.
3	Package Changes. Changes to an environment are exported into a change basket, by comparing one fingerprint data set to the current environment.
4	Package Data Sets. Full data sets are exported into a basket. Data sets are resolved from a list of data set name patterns using the catalog
5	View Basket. Displays the list of data sets, PDS members, and the actions to be performed when applying the basket
6	Apply Basket. Apply changes contained in a basket to the target environment.
7	View Fingerprint. Displays the data set names, attributes, and optionally, the members represented by the fingerprint data set.
8	Detect Changes. Compares fingerprints of two environments, and displays the results.
9	Package Changes. Changes to an environment are exported into a change basket, by comparing two fingerprint data sets.
G	Generate a fingerprint token. Generates the token for an entire PDS, physical sequential file, or a PDS member. This allows a quick verification that a single file or member is in sync with another. For example, if a token is generated for a PDS before and after transferring to a remote site, a matching token will guarantee the PDS was transferred without corruption.

Character	Function
C	Check Synchrony. Verifies the synchrony of two local data sets, or data set groups. Performs byte-level comparison of selected members from one or two environments. When two environments are specified, displayed members can be restricted to those in-sync or out-of-sync. The out-of-sync members are identified and can be compared at a byte-level using the Comparex, ChangeMan SSM, or SuperC interface.
P	PDS Compare. Reports state of synchronization between two PDSs.
S	Synchronize. Generates the JCL needed to synchronize two groups of local data sets in a one-step process, or save the Basket without applying to the Target environment.
A	Advanced Operations. This sub-menu has panels which allow more options for the Detection and Synchronization functions. In addition, all of the functions available from this menu produce batch JCL, which is suitable for large environments and long running jobs.
X	Exit. Exits ChangeMan SSM.

- 2 Press Enter to proceed to the next panel, or press End (PF3) to return to the main menu.

Option (0) - Masking and Filtering Options

When using the fingerprint technology, selected areas of an environment can be masked.

Information such as compile date and time that is generated by the source-embedded `&sysdate` or `&sysime` macros is placed in the load module, thereby affecting the tokens that represent the load module contents. As a result, such load modules may be logically identical, but physically different.

The undesired data elements can be masked to neutralize their presence in the token generation process. All masking rules are defined using this panel.

The masks remain in effect for all advanced operations until you change them or turn them off.

Refer to ["MASK Command" on page 174](#) for additional information.

```

HPSUMASK ----- Masking and Filtering Menu -----
Option ==>

1 - Source or Text Masking      (Generic Source Masking)
2 - Load Module Masking        (Generic Date, Time and IMS Masking)
3 - Show Masks                  (Current Masks in Effect for SSM Functions)

4 - Update Excluded Data Set List for Synchronization

Press END to Exit

```

The following table describes the Masking and Filtering menu items:

Option	Description
1 - Source or Text Masking	Define the mask rules for source or text files. Masks user-defined areas, character strings, area boundaries, and bytes with binary zeros before or after a found string before generating the FGP tokens.
2 - Load Module Masking	Define the masking rules for load modules, such as embedded compile date and time, IMS generated ACB, and MFS control blocks.
3 - Show Masks	Display the current mask rules for the detection functions. The mask commands are applicable to both base and target environments. Any online fingerprinting or batch JCL generated during the session will be affected by the mask rules.
4 - Update Excluded Data Set List for Synchronization	Create and maintain a table of patterns to automatically exclude matching data set names from the list initially returned for Base and Target environments. This pattern table is preserved across ISPF sessions.

Use this procedure to select a function.

- 1 Type the character for the desired function.
- 2 Press Enter to proceed to the next panel, or press End to return to the previous panel.

Specify Source or Text Mask Rules (Option 1)

If a mask command is generated by this panel, user-defined strings or specified areas are replaced with binary zeros before generating the tokens.

For more information on how to access this panel, ["Option \(0\) - Masking and Filtering Options" on page 22](#).

```

HPSUMSK1 -----Source and Text Masking-----
Command ==>

Specify the Masking Rules for all operations that involve Fingerprinting.

Language Masking      ==> NONE          (None, JCL, COBOL, ASSEM)
Squeeze White Spaces  ==> Y            (Y: Multiple Blanks into One)
MIXED or UPPER Case   ==> MIXED        (MIXED: asis, UPPER: convert)

AREA to be Masked:    COL    LEN    (COL may be a signed value)
                    ==>  _____

STRING Search and Mask:
  Search for String   ==> _____ (Upper/Lower asis)

  Area to Search      LOWBOUND HIGHBOUND    (Optional)
                    ==>  _____

  Bytes Before/After  BEF    AFT    (Optional - Nr of Bytes to MASK)
                    ==>  _____

Press END to save changes and exit

```

Items on the Source and Text Masking panel are described in the following table:

Parameter	Description
Language Masking	Applies a set of pre-defined masking rules to the source or text data sets in the designated base or target environments. The options are COBOL, ASSEMBLER, and JCL.
Squeeze White Spaces	Specify if blanks are to be compressed. Y - squeeze multiple consecutive blanks into a single blank before the record is passed to the fingerprinting process. N - do not compress.
MIXED or UPPER Case	Specify if characters are to be converted to uppercase before generating the token. MIXED - do not convert. UPPER - convert to uppercase.
AREA to be Masked	Specifies the starting column and the length of the area to be masked. COL - starting column of the area to be masked. A negative value indicates a position relative to the end of the record. LEN - length of the area to be masked. If area masking is desired, both COL and LEN must be specified.
STRING Search and Mask	Specifies the string to search for.
Area to Search	Optional. Restricts the search to the area between LOWBOUND and HIGHBOUND.
Bytes Before/After	Optional. Specifies the number of bytes to be masked before and after the found string. BEF - number of bytes that are to be masked before the string. AFT - number of bytes that are to be masked after the string.

Use this procedure to specify masking rules.

- 1** Enter values for the Language Masking, Squeeze White Spaces, and MIXED or UPPER case parameters.
- 2** Optionally, enter values for the remaining parameters.
- 3** Press Enter to save your masking specifications, or press End to return to the Masking and Filtering menu.

Specify Load Module and IMS Mask Rules (Option 2)

These masking rules are applicable to load libraries in the both base and target environments.

```

HPSUMSK2 ----- Load Module and IMS Masking-----
Command ==>

Specify the Masking Rules for all operations that involve Fingerprinting.

Load Module Masking:
  CSECT processing           ==> N      (Yes or No)  MASK=CSECT
  Compile Date               ==> N      (Yes or No)  MASK=DATE
  Compile Time               ==> N      (Yes or No)  MASK=TIME

IMS Library Masking:
  Application Control Blocks (ACB) ==> N      (Yes or No)  MASK=ACB
  Message Format Services (MFS)   ==> N      (Yes or No)  MASK=MFS

Press END to save changes and exit

```

The following table describes fields on the Load Module and IMS Masking panel:

Parameter	Description
CSECT processing	Specifies that each CSECT of the load module is to be fingerprinted separately. The tokens generated are then used to produce a composite token for the entire load module. The order of CSECTS does not affect the generated tokens for the load module.
Compile Date	Causes ChangeMan SSM to search for the compile date, and if found, mask the date. As a result, two identical programs compiled on two different dates will generate identical tokens.
Compile Time	Causes ChangeMan SSM to search for the compile time, and if found, mask the time. As a result, two identical programs compiled at two different times will generate identical tokens.
Application Control Blocks (ACB)	Automatically applies predetermined masking rules for ACB modules. As a result, ACB modules created at different times will generate identical tokens if the contents are logically identical.
Message Format Services (MFS)	Automatically applies predetermined masking rules for MFS files. As a result, MFS files created at different times will generate identical tokens if the contents are logically identical.

Use this procedure to define load module and IMS mask rules.

- 1 Enter values for each of the fields in the panel.
- 2 Press Enter to save your masking specifications, or press End to return to the Masking and Filtering menu.

View Mask Rules in Effect (Option 3)

This panel displays the overall masking rules in effect for the session.

The following sample Session Masks in Effect panel shows that load module masking (CSECT, DATE, and Time) is in effect.

You can modify these rules with Option 1 or Option 2 on the Masking and Filtering menu.

```

HPSUMSK3 ----- Session Masks in Effect-----
Command ==>

The following are the MASK(s) in effect for all ISPF SSM Functions.
To change these rules, use options 1 or 2 from the MASK menu.

Language Masking: ==> MASK=(SQUEEZE)
Area Masking:      ==> MASK=(NONE)
String Masking:   ==> MASK=(NONE)
Load/IMS Masking: ==> MASK=(NONE)

Press END to exit

```

The following table describes the parameters in the Session Masks in Effect panel:

Parameter	Description
Language Masking	Displays the MASK command (if language masking is used).
Area Masking	Displays the MASK command (if area masking is used).
String Masking	Displays the MASK command (if string masking is used).
Load/IMS Masking	Displays the MASK command (if Load/IMS masking is used). MASK=(CSECT,DATE,TIME) applies to load modules. MASK=ACB or MASK=MFS applies to the IMS. Application Control Blocks or Message Format Services.

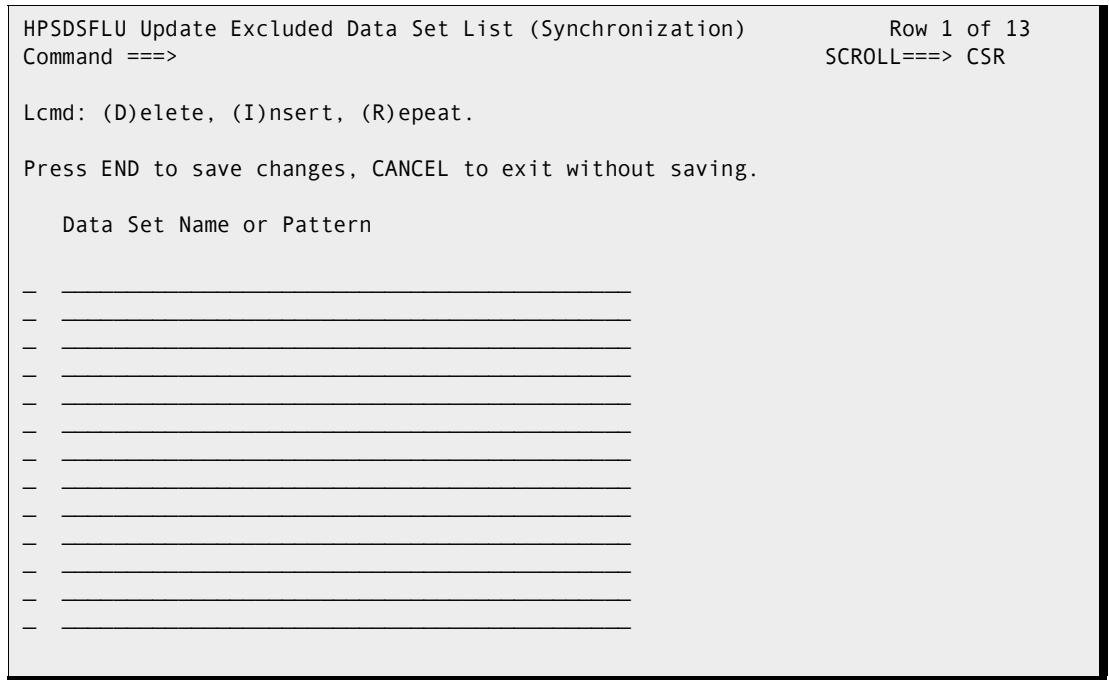
When you finish reviewing the information, press End to return to the Masking and Filtering menu.

Update Excluded Data Set List for Synchronization (Option 4)

Use this option to create and maintain a persistent list of explicit data set names or data set name patterns which you do not want to consider for synchronization. This list is preserved across ISPF sessions.

When using the Fingerprint and Check Synchrony and Synchronize functions, the data set names displayed on the Resolved Data Sets panel will not include those which match an entry in this exclude list.

Within Advanced Operations (Option A on the Detection and Synchronization menu), for jobs created which select data sets by pattern from one or more DASD volumes, an EXCLDSN statement will be generated for each entry in this list.



How to Use This Panel

Use the following procedures to add, delete, and update data set names and patterns.

A data set name pattern consisting of only a "/" or an "*" is not allowed. To see all of the available patterns supported by this function, refer to [Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243](#) of this guide.

When you are finished, press End/PF3 to save your changes, or type CANCEL on the command line and press Enter to exit the panel without saving.

To Add or Update an Entry

- 1 Enter an explicit data set name or pattern to be excluded. You may type on a blank line or over an existing entry. Following are some examples of data set name patterns:

Pattern	Result
OLINK/	Exclude if the last node starts with LINK.
AAA.BBB./	Exclude if the first two nodes are AAA.BBB.
AAA.*.BBB./	Exclude if the first and third nodes are AAA and BBB respectively.
!PROC!	Exclude if any node contains the string PROC.

- 2 Press Enter to validate your input.

To Delete an Entry

- 1 Type a D in the line command field next to the entry to be deleted.

- 2 Press enter to update the list.

To Repeat an Entry

- 1 Choose an entry that you want to repeat and type an R in the line command field.
- 2 Press enter to repeat the line.

To Insert Additional Lines

- 1 Type an I in any line command field.
- 2 Press enter to insert a blank line.

Option (1) - Create Fingerprint Data Set

This function creates a representation of a software environment. To compare or synchronize two versions of software, the target environment:

- Creates a fingerprint data set that represents the group of data sets in that environment.
- Transfers the fingerprint file to the base environment. The fingerprint is now available for use in change detection or synchronization processes.

A similar approach detects changes in a software package. This is done as follows:

- Create a fingerprint.
- Change the code using your existing methodology.
- Review the changes since creating the fingerprint.
- Create a new fingerprint.
- Compare the old and new fingerprints; report the differences.

Specify Environment

Use this procedure to specify (explicitly or as a pattern) the data sets that comprise the software environment.

HPSURMT1	Create Fingerprint Data Set	Row 1 of 13
Command =====>		SCROLL====> CSR
Enter one or multiple Explicit DSN(s) and/or DSN Pattern(s) on the lines below.		
Specify a NEW or Existing Fingerprint DSN:		
Fingerprint DSN (PS)	====> USER123.FINGER.PRINT	
Catalog Search	====> Y N or Y: Resolve data sets through catalog	
Explicit or Pattern DSNs similar to ISPF 3.4		

- 1 Type values for the following options:

Parameter	Description
Fingerprint DSN (PS)	Type the name of the fingerprint data set that is to contain the base environment fingerprint tokens. If it is an explicit name, enclose it in quotes (or the current TSO userid will be prefixed). A naming convention, such as including a second node of <i>FGP</i> , is recommended for the fingerprint data sets. If the <i>FGP</i> does not exist, it is allocated. Existing fingerprint data sets must have the following attributes: DCB=(DSORG=PS, RECFM=FB, RECSIZE=512, BLKSIZE=4096)
Catalog Search	Y - the data sets in the group are cataloged (a catalog search is performed to resolve the pattern). N - the data sets are not cataloged and must be specified explicitly.
Explicit or Pattern DSNs	Specify the explicit data set names or data set patterns that comprise this environment. There are no limits to the number of data sets or patterns that may be entered. Use the Lcmd commands to manipulate beyond the limits of the initial panel. Both explicit and pattern data sets may be used concurrently. Multiple data set patterns are allowed. See " Catalog Search " on page 248 for information on specifying pattern names.

- 2 Press Enter to proceed to the next panel, or press End to return to the previous menu.

Resolve Data Sets

Use the Resolved Data Sets (BASE) panel to refine the list of data sets to be fingerprinted. Exclude any unwanted data sets that may have been included.

```

HPSUSYNR -----Resolved Data Sets (BASE)----- Row 1 of 4
Command =====>                                SCROLL====> CSR

Enter X to Exclude any data set. Overtyp e Volser(s) for Uncataloged DSN(s).

Lcmd:  X

Press ENTER To Continue or END to Exit

  Resolved Data Sets                                Volser
_  USER14.TEST.PDS.LOADLIB                          HPS001
_  USER14.TEST.PDS.PANELS                           WORK01
_  USER14.TEST.PDS.SOURCE                           HPS001
_  USER14.TEST.PS.DATA                              HPS002
***** Bottom of data *****

```

If there are two sets of identical data set names (one set cataloged and the other uncataloged), the cataloged versions can be resolved first. The *volser* of the cataloged data sets subsequently can be changed to point to the uncataloged data set.

Volser is the DASD volume serial number where the data set is resident. This field can be changed to point to uncataloged data sets residing on another volume.

When you finish, press Enter to proceed with the fingerprinting process. The screen is locked until fingerprinting is complete.

To further refine the resolved list of data sets:

- 1 Type an X in front of any of the data sets you want to exclude from the list, then press Enter.
If you exclude a data set from the resolved list, the list is re-positioned at that point. Press PF7 to go to the top of the list.
- 2 Press Enter again to proceed with the fingerprinting process, or press End to return to the previous panel.

Fingerprint Data Set Pattern List

The Fingerprint Data Set Pattern List panel displays. You can enter the following information:

- Description of Environment – a simple description of the environment that the fingerprint data sets represent.
- High Level Qualifier of FGP file – the high level qualifier that is common to all fingerprint data set names for an environment.

The fingerprint data set pattern list can be brought up in the View Fingerprint, Review Changes, and Package Changes panels by specifying * for the fingerprint data set name pattern.

The populated fingerprint data set can be transferred to a remote site (for remote synchronization tasks), or saved for future change verifications during a change cycle.

```

HPSUFGPL ----- SSM Fingerprint Data Set Pattern List -----
Command ==>
HPS262I Fingerprint Data Set is created successfully.

    Optionally enter a description and a high level qualifier to identify
    the Fingerprint file. This allows future selection of FGP files for a
    given application or environment. Overtyp e entries to MODIFY or REMOVE.

    Fingerprint File ==> USER14.FGP.TEST.Y2000.M01D10

Press ENTER to continue or END to exit

----- FGP Patterns -----
    Description of Environment          High-Level Qualifier of FGP File
_   Local Development Environments_____ USER14.P390_____
_   Remote Development Environments_____ USER14.P391_____
_   My test FGPs_____                 USER14.FGP.TEST_____
_   _____
_   _____

```

If you modified the FGP Patterns table, press Enter to store the change, then press End to return to the Detection and Synchronization menu.

Option (2) - Review Changes

This function verifies the changes in an environment. The “before change” environment is represented by its fingerprint data set. Current data sets for which there is a representation in the FGP file are internally re-fingerprinted. The two sets of fingerprints are then evaluated for change.

To verify the synchrony of remote sites, transfer the fingerprint data set created at the target site to the base site. Only the group of data sets of concern will be evaluated for change.

Specifying a Previously-Generated FGP Data Set

```

HPSUREV1                                Review Changes
Command ==>

Specify a previously generated Fingerprint data set, to identify all changes
made to the environment it represents, since the Fingerprint was generated.

Fingerprint DSN or pattern ==> *
  Extended Sync Check    ==> Y          - Y/N  (Include directory info)

Reporting Options:
  Exceptions only        ==> Y          - Y/N
  -or-
  Sync                  ==>           - Y/N
  OutSync               ==>           - Y/N
  Mismatch              ==>           - Y/N

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
      ending with *. If * is specified alone, then the Fingerprint DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit

```

Use this procedure to specify the current environment.

- 1 Enter values for the following options:

Parameter	Description
Fingerprint DSN or Pattern	Type the name of the previously-generated fingerprint data set (or one that was received from the remote site). This represents the target environment. The entire data set name, without quotes, must be specified for an explicit data set name. GDGs may be entered using either the explicit or relative generation number, such as: HLQ.FGP.G0010V00 HLQ.FGP(-1) Data set name patterns are also supported. Any DSN ending with an asterisk is resolved by the catalog, and a list of matching names displays. If only * is specified, then the fingerprint data set pattern list displays.
Extended Sync Check	Y - directory information is considered when comparing PDS members. N - directory information is ignored when comparing PDS members. Directory information includes ISPF statistics.
Exceptions Only	Y - only exceptional conditions, such as members and data sets that are out-of-sync or mismatched, are reported. N - reporting will be dictated by the Sync, Outsync, and Mismatch parameters.

Parameter	Description
Sync	Y - report matched and synchronized items (items with equal content in both environments). N - suppress reporting of in-sync items.
OutSync	Y - report matched but non-synchronized items, (items with unequal content across environments). N - suppress reporting of OutSync items.
Mismatch	Y - report mismatched items (items that do not exist in both environments). N - suppress reporting of Mismatched items.

- 2 Press Enter to proceed to the next panel, or press End to return to the Detection and Synchronization menu.

Select Fingerprint Data Set Name Pattern

Use this procedure to select the fingerprint data set name pattern.

```

HPSUFGPL -- ChangeMan SSM Fingerprint Data Set Pattern List (Ver 8.4.0) ---
Command ==>

Select a Fingerprint DSN pattern for the application or environment.
To modify entries from this table, overtype the lines. To remove
entries, overtype with blanks.

Press ENTER to continue or END to exit

----- FGP Patterns -----

Description of environment                High Level Qualifier of FGP file
- Local Development Environments_____ USER14.P390_____
- Remote Development Environments_____ USER14.P391_____
S My test FGPs_____                 USER14.FGP.TEST_____
- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____

```

- 1 Type S to elect the line in the fingerprint data set pattern list whose description and high level qualifier match the desired fingerprinted environment. An character in the Line Command field selects that line.

If the high level qualifier is not present, enter values for the following options:

Parameter	Description
Description of environment	Type a simple description of the environment that the fingerprint data sets represent.
High Level Qualifier	This is the high level qualifier that is common to all fingerprint data set names for fingerprints representing an environment.

Overtyping the line you want to modify, then press Enter to save the change. Then select the line.

- 2 Press Enter to proceed to the next panel, or press End to return to the Detection and Synchronization menu.

Select Fingerprint Data Set Name

Type S on the line Command line to select the desired fingerprint data set.

```

HPSUFGPS          Select a FINGERPRINT Data Set          Row 1 of 3
Command =====>          SCROLL====> CSR

Enter 's' to Select the desired FINGERPRINT from the list below.

_  USER14.FGP.TEST.Y2000.JAN01
_  USER14.FGP.TEST.Y2000.JAN05
S  USER14.FGP.TEST.Y2000.JAN10
    
```

Verifying the Fingerprint Data Set Name

The fingerprint file specified represents the target (old) environment, from which to review changes. The data sets for the base (current) environment are referenced from the target, and are re-fingerprinted in the foreground. The base environment, therefore, is the current environment on DASD. If the fingerprint data set represents a large environment, it is best to create batch submittable JCL through Option 2 of the Advanced Operations menu.

```

HPSUREV2          Review Changes
Command ===>

          Press ENTER to review the changes to the environment.

Selected Fingerprint file ==>USER14.FGP.TEST.Y2000.JAN10

Note: The environment represented by the Fingerprint data set will be
      re-Fingerprinted. This operation may take several minutes, during
      which time your screen will be locked. For large environments, use
      option 2 under Advanced Operations to create JCL for batch submission

Press ENTER to proceed.
    
```

Verify that the name shown in the Selected Fingerprint file field is correct, then press Enter to re-fingerprint the environment and display changes, or press End to return to the previous panel.

Interpret Results

After verifying the fingerprint data set name, a report, similar to the following, displays.

```

Command Column 10      20      30      40      50      60
Count-seg
.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....
  1  "BASE=RESET
  2  "TARG=RESET
  3  "EXTENDED=YES
  4  "TARG=LOAD=DSN=USER14.FGP.TEST.Y2000.JAN10
  5  "REFTARG
  6  "SETPRINT=(EXCEPTIONSONLY=Y)
  7  "COMPARE=DSNBYSN=SHORT

```

HPS0142I HPSTRANS Fingerprint data set loaded successfully.

HPS0942I HPSTRANS Function REFTARG completed:

REPORT OF IN-SYNC DATA SETS

MemCnt	Base-vol	Base-dsname	Targ-vol	Targ-dsname
78	HPS001	USER14.TEST.LOADLIB	HPS001	USER14.TEST.LOADLIB
51	HPS001	USER14.TEST.SOURCE	HPS001	USER14.TEST.SOURCE

129 In-Sync Members in 2 In-Sync Data Sets

USER14.TEST.PANELS			Status	USER14.TEST.PANELS		
WORK01	Member			Member	WORK01	
20_13:23	2000/01/10_15:23	HPS@PRI1	-OutSync -	HPS@PRI1	1999/12/	
18_15:42	2000/01/10_15:24	HPSUSYNR	-OutSync -	HPSUSYNR	1999/05/	

HPS0555W	HPSTRANS	Base	Target
		839	839 Member name matches
		837	837 Synchronized members
		2	2 Out of sync members
		0	0 Member name mismatches

OutSync PS	USER14.TEST.DATA	USER14.TEST.DATA

HPS0932I HPSTRANS **SHORT DSNBYSN SUMMARY REPORT:**

MATCHED DATA SETS:

2 In-Sync data sets:

BASE	TARGET	In-Sync Members
129	129	In-Sync Members

1 Out-of-Sync data sets:

BASE	TARGET	Out-of-Sync Members
2	2	Out-of-Sync Members
837	837	In-Sync Members
839	839	Matched Members
0	0	MisMatched Members
839	839	Members

```

UNMATCHED DATA SETS:
  0 Unmatched  BASE data sets with 0 members
  0 Unmatched  TARG data sets with 0 members
  0 TOTAL Unmatched data sets with 0 members

```

In the example, members HPS@PRI1 and HPSUSYNR, and physical sequential data set USER14.TEST.DATA have been changed (indicated by OutSync) since the fingerprint data set was created. Only a partial report is shown, including the compare and report summary portions. The other two PDS had no changes.

Option (3) - Package Changes

This function creates the JCL to compare a fingerprint data set (target) with a live environment (base) to identify any changed members and newly-created data sets.

When the extraction completes, the changes are exported to a change basket data set. There is a super directory control area in the beginning of the change basket data set; this area indicates what has been unloaded from the base, and what will be imported to the target environment.

```

HPSUPKG1                               Package Changes
Command ==>

Specify a Fingerprint DSN or a Pattern to package the changes made to the
environment since the Fingerprint was taken. This panel will generate batch
JCL for review/submission.

Fingerprint DSN (Like 3.4) ==> USER14.TEST.Y2000.JAN10

Change Basket DSN ==> BASKET.TEST.Y2000.JAN10
Basket Comment ==> CHANGES TO THE TEST ENVIRONMENT

(New only) DASD Unit ==> SYSDA (SYSDA, SYSALLDA...)
Space Units ==> TRK (TRK, CYL, BLK)
Primary ==> 5 (1-99999)
Secondary ==> 5 (0-99999)

Saved Member Name ==> PACKAGE (Saved member name in user JCL library)

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
ending with *. If * is specified alone, then the Fingerprint DSN
patterns selection table will be displayed.

Press ENTER to continue or END to exit

```

Enter Data Set Names

Specify the fingerprint data set name for the target environment.

If you specify a data set name ending with *, the pattern is resolved, and you can select a fingerprint data set from a list.

If you specify * alone, the Fingerprint Data Set Pattern List panel displays.

To package the changes:

- 1 Enter values for the Fingerprint DSN, Basket DSN, and Member Name fields. these parameters are required.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to proceed, or press End to return to the Synchronization Menu.

The following table describes the parameters displayed in the Package Changes panel.

Parameter	Description
Fingerprint Data Set	Type the name of a fingerprint data set representing the target environment. This is the original (old) environment. The current data sets will be re-fingerprinted, and the changes will be extracted and exported into a basket. Data set name patterns are also supported. Any DSN ending with a * will be resolved by the catalog, and a list of matching names display. If only * is specified, then the fingerprint data set pattern list displays (see "Select Fingerprint Data Set Name Pattern" on page 33).
Change Basket DSN	Type the name of a change basket data set to hold the super directory and the actual changed members or files and newly created data sets. This data set is used in later steps as the input to the Apply Basket function. Enclose it in quotes for an explicit name or the current TSO userid is prefixed. If this is a new data set, input the allocation specifications.
Basket Comment	This field allows up to 40 alphanumeric characters to describe the change basket. Special characters (punctuation and symbols) are not allowed in the description.
DASD Unit	Type an esoteric name for the allocation of the new data set. Only specify this for a new basket allocation.
Space Units	Specify TRK, CYL, or BLK for the allocation type of the new data set. CYL allocation is recommended for large baskets.
Primary	A value in the range 1-99999 can be specified for primary allocation size of the data set.
Secondary	A value in the range 0-99999 can be specified for secondary allocation size of the data set. Allow secondary allocation for large baskets.
Saved Member Name	The default member name can be changed.

Create Change Basket Verification

Verify that the specifications below are correct, then press Enter to generate the JCL. After the job completes, verify that the "Basket Created Successfully" message prints. View the created basket using Option 5 (View Basket).

```
HPSUPKG2                Create Change Basket
Command ==>>>

                                Press ENTER to generate a job to package changes into a Basket.

                                Review the JCL, and then SUBMIT for execution.

Fingerprint DSN ==> USER14.FGP.TEST.Y2000.JAN10
Basket DSN      ==> USER14.BASKET.TEST.Y2000.JAN10
Comment        ==> CHANGES TO THE TEST ENVIRONMENT

After submitting the job, and verifying that the Basket was successfully
created, the Basket DSN pattern may be entered into the Basket DSN pattern
table. To do this, select Option-5 (View Basket) from the Synchronization
menu and enter * for the Basket DSN.

Press ENTER to proceed.
```

Press Enter to proceed, or press End to return to the Package Changes panel.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Option (4) - Package Data Sets

This function takes entire data sets, and places them into a basket. Environments can be transferred to another site, or duplicated with a different high level qualifier using this basket.

Specify Data Sets

The data sets may be specified explicitly, or as a data set name pattern.

```

HPSUCRE1                                Create Basket                                Row 1 of 13
Command =====>                        SCROLL====> CSR

Enter one or more explicit or pattern DSN(s) on each line

Specify the Basket DSN which will contain the resolved data sets:
New or Existing Basket DSN ==>
  Basket Comment      ==> YOUR COMMENT HERE
(New only) DASD Unit  ==>          (SYSDA,SYSALLDA...)
                   Space Units ==>          (TRK,CYL,BLK)
                   Primary      ==>          (1-99999)
                   Secondary     ==>          (0-99999)

Saved Member Name ==> CREATFUL (Saved member name in JCL library)

Press ENTER to continue or END to exit

  Explicit or Pattern DSNs similar to ISPF 3.4
- _____
- _____
- _____
- _____
- _____
- _____

```

Use this procedure to specify the data sets.

- 1** Enter values for the Basket DSN and Member Name fields. These parameters are required.
- 2** Enter one or more explicit pattern DSNs for the data sets you want to package into the basket. At least one DSN is required.
- 3** If the basket does not exist, enter values for the basket allocation parameters.
- 4** Press Enter to resolve the DSNs, or press End to return to the Detection and Synchronization Menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the panel parameters

Parameter	Description
Basket DSN	Type the name of a change basket data set to hold the super directory and the actual offloaded data sets. This data set is used in later steps as the input to the import function. Enclose it in quotes for an explicit name or the current TSO userid is prefixed. If this is a new data set, input the allocation specifications.
Basket Comment	This field allows up to 40 alphanumeric characters to describe the change basket. Special characters (punctuation and symbols) are not allowed in the description.

Parameter	Description
(New DSN) Unit	Enter esoteric name for the allocation of the new data set.
Space Units	Specify TRK, CYL, or BLK for the allocation type of the new data set. CYL allocation is recommended for large tasks.
Primary	A value in the range 1-99999 can be specified for primary allocation size of the data set.
Secondary	A value in the range 0-99999 can be specified for secondary allocation size of the data set. Allow secondary allocation for large tasks.
Saved Member Name	The default member name can be changed.
Explicit or Pattern DSNs	Type a line command. The line commands are: I – Insert a blank line following the current line. R – Replicate; repeats the current line. D – Delete the current line. Specify the explicit data set names or data set patterns that comprise this environment. There are no limits to the number of data sets or patterns that may be entered for an environment. Both explicit and pattern data sets may be used concurrently. Multiple data set patterns are allowed. See " Catalog Search " on page 248 for information on specifying pattern names.

Resolve the Data Sets

This panel lets you refine the list of data sets to be packaged. Exclude any unwanted data sets from this table. Through this facility, if there are two sets of identical data set names (one set cataloged and the other uncataloged), the cataloged versions can be resolved first. Later, on this panel, the *volser* of the uncataloged data sets can be changed to point to the uncataloged data sets.

Upon finalizing the list, press Enter to proceed with the create basket process.

```

HPSUSYNR -----Resolved Data Sets (BASE)-----ROW 1 OF 7
Command =====>                                SCROLL====> CSR
Enter X to Exclude any data set. Overtyp e Volser(s) for Uncataloged DSN(s) .
Lcmd, X
Press ENTER to continue or END to Exit
  Resolved Data Sets                                Volser
  _ SER13.TEST.CNTL                                SER001
  _ SER13.TEST.LINKLIB                              SER004
  _ SER13.TEST.MAC                                  SER002
  _ SER13.TEST.MSGS                                SER001
  _ SER13.TEST.PANELS                              SER003
  _ SER13.TEST.SKELS                               SER001
  _ SER13.TEST.SRC                                 SER005
***** BOTTOM OF DATA*****
    
```


Use this procedure to resolve data sets in the base environment.

- 1 Type X in front of the data sets you want excluded from the list.
- 2 Press Enter to resolve the Data Set List and proceed to the next panel, or press End to return to the previous panel.

If you exclude a data set from the resolved list, the list is re-positioned from that point. Use the PF7 key to return to the top of the list.

The following table describes the parameters displayed in the Resolved Data Sets panel.

Parameter	Description
Resolved Data Sets	Line Command: X -- Exclude items from further processing. It can be used concurrently on multiple lines.
Volser	The DASD volume serial number where the data set is resident. This field can be changed to point to uncataloged data sets residing on another volume.

Create Basket Verification

This panel is a reminder to review the generated JCL before submission, and that the created basket may be viewed using Option 5 (View Basket).

```

HPSUNOT2                Create Basket
Command ==>

      Press ENTER to generate JCL to create a Basket of the environment.

      Review the JCL, and then submit for execution.

After submitting the job, and verifying that the Basket was successfully
created, the Basket DSN pattern may be entered into the Basket DSN pattern
table. To do this, select Option-5 (View Basket) from the Synchronization
menu and enter * for the Basket DSN.

Press ENTER to proceed.

```

Press Enter to generate the JCL, or press End to return to the previous panel.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Option (5) - View Basket

View Basket displays a list of the contents of a basket. Use this to verify the contents of a basket before IMPORT/APPLY processing.

```
HPSUBSKV ----- ChangeMan SSM View Basket (Ver 8.4.0) -----  
Command ==>  
  
Specify an explicit Basket DSN or Basket DSN pattern ending with *.  
  
Basket DSN or pattern ==> USER14.BASKET.TEST.Y2000.JAN10  
  
NOTE: The Basket DSN can be specified explicitly, or as a DSN pattern  
       ending with *. If * is specified alone, then the Basket DSN  
       patterns selection table will be displayed.  
  
Press ENTER to continue or END to exit
```

Use this procedure to display the contents of the basket.

- 1 Type a basket name in the Basket DSN field. This parameter is required.
If wanted, a basket DSN name pattern ending with * may be specified.
If * is specified alone, then the Basket Data Set Name Patterns panel displays for selection of a name pattern.
In both these cases, an intermediate panel displays with a list of resolved data set names to select a basket DSN from
- 2 Press End to process the View Basket command in the foreground, or press End to return to the Synchronization Menu.

The results of the View Basket operation display in an ISPF panel.

This report is deleted automatically if the Report Panel option in the User Options panel is set to N.

If Report Panel is set to Y, then the choice to Delete, Print, and Keep the report are provided.

For information on how to access this panel, see "Select a Function" on page 21, Option 6.

```

HPSUAPL                               Apply Basket
Command =====>

The Basket contains all changed members and/or entire new data sets.

Basket DSN or pattern    ==> USER14.BASKET.TEST.Y2000.M01D10
Apply or Simulate       ==> APPLY                (APPLY/SIMULATE)

Optional Data Set RENAME:
Old High Level Qualifier ==> *                (HLQ of data sets in Basket)
New High Level Qualifier ==> *                (HLQ for Target data sets )

Optional VOLSER redirect:
VOLSER for new data sets ==> * (* indicates system managed allocation)

Saved Member Name       ==> APPLY (Saved member name in user JCL library)

Press ENTER to create the JCL or press  END to exit.
    
```

The following table provides a description of the parameters displayed in the Apply Basket panel.

Parameter	Description
Basket DSN or Pattern	Enter one of the following: <ul style="list-style-type: none"> The explicit name of a basket data set. A basket data set name pattern. After pressing Enter, a list of matching names displays, where you may choose a data set name. A single asterisk. After pressing Enter, the <i>Basket Data Set Pattern List</i> panel displays, where you may enter a data set name or pattern, or choose one that was previously entered.
Apply or Simulate (Import RUN Mode)	Enter SIMULATE or APPLY. SIMULATE – process the Super Directory and simulate the outcome of an APPLY. This process generates a report that can be used as a change ticket. APPLY-- copy the changed members being exported from the base environment. IEBCOPY or its functional equivalent (such as PDSFAST) is used to copy the PDS members to the target libraries.
Old High Level Qualifier	* is the default. This is an optional parameter that is used when the base high level qualifier is different than the target high level qualifier. Specify the base high level qualifier here.
New High Level Qualifier	* is the default. This is an optional parameter that is used when the base high level qualifier is different than the target high level qualifier. Specify the target high level qualifier here.

Parameter	Description
VOLSER for New Data Sets in the Target	* - allows the catalog to be used to allocate the data sets in the change basket to the target environment for existing data sets. A <i>volser</i> specification confines the allocation of new data sets in the change basket to the specified volume.
Saved Member Name	Default member name; can be changed.

Use this procedure to apply a change basket.

- 1 Specify values for *Basket DSN or Pattern*, *Apply or Simulate*, and *Saved Member Name*.



NOTE See the *Basket DSN or Pattern* field description in the above table for additional information.

- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL, or press End to return to the Detection and Synchronization menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Option (7) - View Fingerprints

View Fingerprints lists the contents of a fingerprint data set. Use this to verify the contents of a fingerprint before a comparison or a package operation. The member names for each PDS represented by the fingerprint data set can optionally be displayed.

```

HPSUFGPV ----- View Fingerprint Data Set -----
Command ==>

Specify an explicit Fingerprint DSN or Fingerprint DSN pattern ending with *.

Fingerprint DSN (like 3.4) ==> USER14.FGP.TEST.Y2000.M01D10
      Member Reporting ==> Y                (Y/N - Display member list)

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
      ending with *. If * is specified alone, then the Fingerprint DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit

```

The following table describes the parameters displayed in the View Fingerprints panel.

Parameter	Description
Fingerprint DSN	Name of a fingerprint data set to be viewed. This may be entered as an explicit data set name without quotes, or as a data set name pattern ending with *. If a pattern is specified, a list of data sets is presented. If * is specified, the Fingerprint Data Set Pattern List displays.
Member Reporting	Specify if names are to be reported. N – do not display the names of the members with tokens stored in the fingerprint data set. This is the default. Y – report all member names.

Use this procedure to view a fingerprint data set.

- 1 Type values for Fingerprint DSN and Member Reporting fields. These parameters are required.
- 2 Press Enter to generate the report or press End to return to the Detection and Synchronization menu.

The results of the view fingerprint operation display in an ISPF panel. This report is automatically deleted if the Report panel option in the User Options panel is set to N. If the Report Panel parameter is set to Y, then the choice to Delete, Print, and Keep the report is provided. For a description of this option, ["Modifying User Options" on page 19](#).

The member lists from USER14.TEST.PANELS and USER14.TEST.SOURCE were omitted so that the report would fit on a single printed page.

```

Command      Column  10          20          30          40          50          60          7
Count-seg    ....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....!.....
  1         "BASE=RESET
  2         "TARG=RESET
  3         "BASE=LOAD=DSN=USER14.FGP.TEST.Y2000.M01D10
  4         "VIEWFGP=BASE=MEMBER=YES

HPS0575I  HPSTRANS  Fingerprints loaded for Dsn: USER14.TEST.PDS.LOADLIB
HPS0575I  HPSTRANS  Fingerprints loaded for Dsn: USER14.TEST.PDS.PANELS
HPS0575I  HPSTRANS  Fingerprints loaded for Dsn: USER14.TEST.PDS.SOURCE
HPS0575I  HPSTRANS  Fingerprints loaded for Dsn: USER14.TEST.PS.DATA
Masking in Effect: NONE

HPS0142I  HPSTRANS  Fingerprint data set loaded successfully.
                    4 data sets included.
                    0 data sets excluded.
                    Dsname = USER14.FGP.TEST.Y2000.M01D10
                    DDname = TREEDD

HPS0906I  HPSTRANS  Fingerprint data set loaded to the Base:

                    C O N T E N T S   O F   B A S E
Time-of-Fingerprint  NrMem  Fmt  Lrecl  Blksiz  Volser  Dsname
2000/01/10 16:18:46   48  U    0   6,000  HPS001  USER14.TEST.LOADLIB
      HPSABOVE HPSGROUP HPSKEYED HPSREBLK HPSSNAPV HPSUDSCB HPSVSAM HPSZISVC
      HPSBASKV HPSICLOS HPSMAINT HPSRENAM HPSSNAPX HPSUEDIT HPSVSAMX HPSZMLS
      HPSBELOW HPSIDCAM HPSMEMS  HPSREPR  HPSSNAPZ HPSUFR4K HPSVSYNC HPSZREV
      HPSCOMPR HPSIDCVS HPSNCOMP HPSRTO   HPSSTART HPSUNBLK HPSWTRTO HPSZROUT
      HPSDCOMP HPSINIT  HPSNOKEY HPSRTWTO HPSTASK1 HPSUSVCT HPSZAUTH HPSZSVC
      HPSDSFTC HPSINIT0 HPSNOTFY HPSRWAIT HPSTASK2 HPSUTO4K HPSZBRGT HPSZSYNC

2000/01/10 16:18:50   839  FB    80   6,000  WORK01  USER14.TEST.PANELS
      < MEMBER LIST OMITTED >
2000/01/10 16:18:51    51  FB    80   6,000  HPS001  USER14.TEST.SOURCE
      < MEMBER LIST OMITTED >
2000/01/10 16:18:51    PS   FB    80   6,000  HPS002  USER14.TEST.DATA

HPS0912I  HPSTRANS  16:18:52 Base:
                    3  Nr of PDS data sets in group.
                    1  Nr of non-PDS data sets in group.
                    4  Total data sets in group.

                    968  Nr of PDS members in group.
                    3  Nr of identical member sets.
                    3  Nr of redundant members in the identical me

                    1,621,384  Total PDS hashed bytes.
                    41,000  Total non-PDS hashed bytes.

                    0  Orphaned aliases.
                    0  Non orphaned aliases.
    
```

Option (8) - Detect Changes

This function compares two fingerprint data sets (base and target) to determine the changed members, newly found data sets, and scratched data sets. This lets you review changes in environments, as well as verify that remote environments are synchronized.

If environments are to be synchronized, run Detect Changes as a pre-step to review what will be placed into the basket.

HPSUREV3	Detect Changes	
Command ==>		
Specify the BASE and TARGET Fingerprints to view changes between the two Fingerprinted environments.		
Explicit or pattern Fingerprint data sets:		
BASE (NEW)	==>	*
TARGET (OLD)	==>	USER14.*
Extended Sync Check	==>	Y - Y/N (Include directory info)
Reporting Options:		
Exceptions only	==>	Y - Y/N
-or-		
Sync	==>	- Y/N
OutSync	==>	- Y/N
Mismatch	==>	- Y/N
NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern ending with *. If * is specified alone, then the Fingerprint DSN patterns selection table will be displayed.		

The following table describes the parameters displayed in the Detect Changes panel.

Parameter	Description
Base (Fingerprint Data Set)	Type the name of a fingerprint data set representing the Base environment. This is the changed (new) environment from which the changes will be extracted in a synchronization operation.
Target (Fingerprint Data Set)	Type the name of a fingerprint data set representing the target environment (this represents the 'before change' environment).
Extended Sync Check	Specifies if directory information is to be considered when comparing the members. Directory information includes ISPF statistics. Y - consider the directory information (stored in the fingerprint file) when comparing PDS members. N - ignore directory information.
Exceptions Only	Specifies the type of conditions to report. Y - report exceptional conditions, such as out-of-sync and mismatched data sets and members. N - reporting will be dictated by the Sync, Outsync, and Mismatch parameters. A Y specification is mutually exclusive with other reporting options.

Parameter	Description
Sync	Specifies if synchronized items (those items with equal content in both environments) are to be reported. Y - report synchronized items. N - do not report in-sync items.
OutSync	Specifies if non-synchronized items (those items with unequal content across environments) are to be reported. Y - report out-of-sync items. N - do not report OutSync items.
Mismatch	Specifies if mismatched items (items that do not exist in both environments) are to be reported. Y - report mismatches. N - do not report mismatches.

Use this procedure to compare the data sets.

- 1 Enter values for the Base, Target, and Extended Sync Check fields. These parameters are required.
- 2 Select the reporting option (at least one of the optins must be set to Y).
Exception Only overrides the reporting options with its own settings, and is mutually exclusive with Sync, OutSync, and Mismatch.
- 3 Press Enter to submit the Detect Changes command in the foreground, or press End to return to the Detection and Synchronization Menu.

Option (9) - Package Changes

This function creates the JCL to compare two fingerprint data sets (base and target) and extract the changed members, newly found data sets, and scratched data sets. Once the extract completes, the changes are exported to a change basket data set.

In the beginning of the change basket data set, there is a super directory control area that indicates what has been unloaded from the base, and what will be exported to the target environment

Package Changes

The changes are determined based on fingerprint representations of the base and target environments. The extracted changes are then written to the change basket.

Use a change basket naming convention and comments that clearly represent the purpose of the basket.

```

HPSUPKG3                Create Change Basket
Command ==>

Specify the BASE and TARGET Fingerprints to package changes between the two
Fingerprinted environments.

Explicit or pattern Fingerprint data sets (like 3.4):
  BASE   (NEW)           ==> USER14.FGP.TEST.Y2000.JAN10
  TARGET (OLD)           ==> USER14.FGP.TEST.Y2000.JAN04

Specify Change Basket DSN ==> BASKET.TEST.Y2000.JAN10
      Basket Comment ==> Changes from JAN 4 to JAN 10 2000

(New only) DASD Unit     ==>                               (SYSDA,SYSALLDA...)
      Space Units        ==>                               (TRK,CYL,BLK)
      Primary            ==>                               (1-99999)
      Secondary          ==>                               (0-99999)

      Saved Member Name  ==> CREATE

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
      ending with *. If * is specified alone, then the Fingerprint DSN
      patterns selection table will be displayed.
    
```

The following table describes the parameters in the Create Change Basket panel.

Parameter	Description
Base (Fingerprint Data Set)	Specifies the name of a fingerprint data set that represents the base environment. This is the changed (new) environment from which the changes will be extracted. Data set name patterns are supported. Any DSN ending with a * is resolved by the catalog, and a list of matching names displays. If only * is specified, then the fingerprint data set pattern list displays. (See "Select Fingerprint Data Set Name Pattern" on page 33.)
Target (Fingerprint Data Set)	Specifies the name of a fingerprint data set that represents the target environment (this represents the 'before change' environment). Data set name patterns are supported. Any DSN ending with a * is resolved by the catalog, and a list of matching names displays. If only * is specified, then the fingerprint data set pattern list displays. (See "Select Fingerprint Data Set Name Pattern" on page 33.)
Basket DSN	Specifies the name of a change basket data set to hold the super directory, the actual changed members or files, and newly-created data sets. This data set is used in later steps as the input to the import function. Enclose the name in quotes for an explicit name; otherwise, the current TSO userid is prefixed. If this is a new data set, type the allocation specifications.

Parameter	Description
Basket Comment	Allows up to 40 alphanumeric characters to describe the change basket. Special characters (punctuation and symbols) are not allowed in the description.
(New DSN) Unit	Specifies an esoteric name for the allocation of the new data set.
Space Units	Specifies TRK, CYL, or BLK for the allocation type of the new data set. CYL allocation is recommended for large tasks.
Primary	Specifies the primary allocation size (value 1 - 500) of the data set.
Secondary	Specifies the secondary allocation size (value 0 - 500) of the data set. Allow secondary allocation for large tasks.
Saved Member Name	Specifies the member name if the default is not used.

Use this procedure to create the change baskets.

- 1** Enter values for the Base Fingerprint DSN, Target Fingerprint DSN, Basket DSN, and Member Name fields. These parameters are required.
- 2** Optionally, enter values for the remaining parameters.
- 3** Press Enter to generate the JCL or press End to return to the Synchronization menu.
The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Create Basket Verification

Verify that the specifications are correct, then press Enter to generate the JCL.

You can view the created basket using Option 5 (View Basket).

```
HPSUPKG4                Create Change Basket
Command ==>>>

                Press ENTER to generate JCL to package changes in a Basket.

                Review the JCL, and then submit for execution.

BASE FGP ==> USER14.FGP.TEST.Y2000.JAN10
TARG FGP ==> USER14.FGP.TEST.Y2000.JAN04

BASKET  ==> USER14.BASKET.TEST.Y2000.JAN10
COMMENT ==> Changes from JAN 4 to JAN 10 2000

After submitting the job, and verifying that the Basket was successfully
created, the Basket DSN pattern may be entered into the Basket DSN pattern
table. To do this, select Option-5 (View Basket) from the Synchronization
menu and enter * for the Basket DSN.

Press ENTER to proceed.
```

Press Enter to proceed, or press End to return to the Package Changes panel.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Option (G) - Generate a Fingerprint Token

This function generates and displays the fingerprint token of an entire physical sequential data set or PDS, and optionally a PDS member.

If the synchronization of two data sets or PDS members (local/remote) is in question, this function can be used to determine the status. If the tokens are identical the files have identical contents.

Fingerprint File or Member

The data set or member to be fingerprinted is specified on the Generate Token panel.

```

HPSUFGP0                                Generate Token
Command =====>

ISPF LIBRARY:
  Project  ===> USER14
  Group   ===> CMNSSM
  Type    ===> PANELS
  Member  ===> HPSUFGP0          - Blank or explicit member

OTHER PARTITIONED OR PHYSICAL SEQUENTIAL DATA SET:
  Data Set Name  ===>
  Volume Serial  ===>          - If not cataloged

Press ENTER to continue or END to exit

```

The following table describes the parameters in the Generate Token screen:

Parameter	Description
Project	Specifies the standard ISPF data set project.
Group	Specifies the standard ISPF data set group.
Type	Specifies the standard ISPF data set type.
Member	Specifies the PDS member to be fingerprinted.
Data Set Name	Specifies the name of the data set to be fingerprinted. Enclosing the data set name in quotes indicates an explicit data set. Otherwise, the TSO User ID is prefixed.
Volume Serial	Specifies the volume serial; only required for uncataloged data sets. VSAM data sets are not fingerprinted through this panel.

Use this procedure to fingerprint a data set or PDS member.

- 1 Enter the values for Project, Group, Type, and Member fields for the ISPF LIBRARY (Member need not be provided if the entire data set is to be fingerprinted.)
- OR -
Enter values for the Data Set Name and Volume Serial (if not cataloged) fields for the Other Partitioned Or Physical Sequential Data Set.
Information for either the ISPF Library or the Other Partitioned Or Physical Sequential Data Set must be provided, but not both.
- 2 Press Enter to generate the fingerprint token and continue to the next topic, ["Interpret Fingerprint Results"](#), where you can review the generated tokens; or press End to return to the previous menu.

Interpret Fingerprint Results

Upon return, the desired fingerprints are displayed on the lower part of the panel.

If two partitioned data sets have identical fingerprint tokens, they are identical in content.

The panel is still active; additional requests can be entered by specifying another data set or member.

```

HPSUFGP1                                Generate Token
Command =====>

ISPF LIBRARY:
  Project  ===> USER14
  Group   ===> CMNSSM
  Type    ===> PANELS
  Member  ===> HPSUFGP0          - Blank or explicit member

OTHER PARTITIONED OR PHYSICAL SEQUENTIAL DATA SET:
  Data Set Name  =====>
  Volume Serial  =====>

      Fingerprint for the whole data set  =====> 37D588DD-00103CC1
      Fingerprint for requested member    =====> 2E64CE6A-000010F9

Press ENTER to generate another token, or press END to exit

```

The following table describes the fields in the Take Fingerprint panel after fingerprinting a member or data set.

Output Fields	Description
Fingerprint for the whole data set	Displays the fingerprint for the entire data set. This value can be used to determine if two or more data sets are identical, without actually comparing them. This process is useful when the data sets reside in two remote locations, and their content equality or inequality needs to be determined. This is not an editable field.
Fingerprint for requested member	Displays the fingerprint for the specified member. If a member was not specified, then the "Member not specified" message displays. This is not an editable field.

To generate another fingerprint token, enter another data set or member name, then press Enter, or press End to return to the previous menu.

Option (C) - Sync Checking and Comparing Environments

This function verifies two local data sets (or groups of data sets) for synchrony.

Using the fingerprint technology, ChangeMan SSM can eliminate all data sets or members that have identical tokens, and show only those members that have different content. This feature lets you focus on the members that are different. When two versions of a software package go through a quality assurance process, the changed members are isolated on a single panel.

Define the environments using the HPSDSYN1 (base) and HPSDSYN2 (target) panels. Each of these panels is followed by a panel to resolve or exclude the data set names entered on the prior panel.

The synchronization results are then reported on a subsequent panel.

Each pair of members or files (PS and DA) reported to be out-of-sync can be selected to view the differences, and a Differences Report can be printed.

Refer to the DETECT*, CPXFAST*, or SUPERC* JCL in the SAMPLES library for performing similar functions in batch mode.

The panels displayed in the following examples are for the BUILT-IN comparison utility. You specify the comparison utility on the ChangeMan SSM Defaults and User Options panel (Option 0 on the ChangeMan SSM Main Menu). The choices are BUILT-IN, COMPAREX, and SUPERC.

Specify Base Environment

The data sets that comprise the base environment are specified on this panel, either explicitly or through a pattern.

```

HPSDSYN1                BASE PDS (1 of 2)                Row 1 of 13
Command ==>>>                SCROLL ==>>> CSR

Members reside within ONE environment ==> N    Y: BASE only    N:BASE and TARG
Explicit Member/Pattern ending with * ==> *

Compare like-named cataloged vs. non-cataloged DSNs/patterns ==> N    Y/N
Specify VOLSER ==>

Reporting Options TYPE(S) ==> ADD,UPD,DEL    Specify a list: ADD,UPD,DEL or ALL
Reporting Options RECORD ==> Y                N: Report the first segment only
                                                Y: Report the entire record

Cataloged PDS (or Pattern) ==> Y    -(Y or N) Y: Resolve data sets from catalog
Sync ==> N    -(Y or N) N: Do not report equal members
OutSync ==> Y    -(Y or N) Y: Filter only the differences
Sort by Token ==> N    -(Y or N) N: List sorted by Member Name

Press ENTER to continue or END to exit

    Explicit or Pattern DSNs similar to ISPF 3.4
    _  user298.lpara.* _____
    _  _____
    _  _____
    _  _____
  
```

The following table describes parameters on the BASE PDS panel.

Parameter	Description
Members reside within one environment	Specify if members are in one environment. Y - there is only the base environment. Members can be compared with other members in the same environment, but no Sync or OutSync comparison is performed. N - there are two environments (base and target). In Sync members, OutSync members, or both may be displayed.
Explicit Member or Pattern ending with *	Specify an explicit member name, or a member name pattern to limit the sync check to the members matching this criteria. If * is specified, then all members are considered.
Compare like-named cataloged vs. non-cataloged Specify VOLSER	To facilitate comparing cataloged and uncataloged datasets having the same name, specify if DSNs/patterns are to be copied from this panel to the target panel. Y - copy the DSNs/patterns from this panel to the target panel. N - do not copy the DSNs/patterns from this panel to the target panel. Specify a 6-character alphanumeric volume name to be propagated to the target dataset panel VOLSER field.
Reporting Options TYPE(S)	Specify Differences Report type. Add - Indicate records added in the base member. Del - records deleted from the base. Upd - Indicate updated records in the members. All - Show all records in the members. More than one parameter can be specified. For example: 'Add, Upd, Del' is a valid specification.
Reporting Options RECORD	Specify how the record is to be displayed. Y - display the entire record. N - display only a partial record. As much of the record that fits on a single line will be displayed.
Cataloged PDS	Indicate if the data sets are cataloged. Y - the data sets in the base group are cataloged (a catalog search is performed to resolve the specified patterns). N - the data sets are not cataloged, and must exist by their explicit name. This requires a <i>volser</i> to be specified on the subsequent panel.
Sync	Specify if Sync items are to be reported. Y - reporting of synchronized items (items with equal content in both environments). N - do not report synchronized items.

Parameter	Description
Outsync	Specify if non-synchronized items are to be reported. Y - report non-synchronized items (items with unequal content across environments). N - do not report non-synchronized items. Positive (Y) responses to Sync and Outsync result in reporting both categories; negative (N) responses to both result in statistics only.
Sort by Token	Specify if the list is to be sorted by Fingerprint token or member name. Y - display the Member List sorted by Fingerprint token. The hash token will be shown for all members in the list. The purpose of this option is to enable identifying differently-named members with the same content. N - display the Member List sorted by Member Name. Pairs of members will appear together, and if they are In-Sync, the hash token will be shown for the first member only.
Line Command	Type a line command in the area to the left of each line of the Explicit or Pattern DSNs information area. The line commands are: I - Inserts a blank line following the current line R - Repeats the current line D - Deletes the current line
Explicit or Pattern DSNs	Specify the data set names or patterns that will comprise this environment. There are no limits to the number of data sets or patterns that may be entered for an environment. Use the Lcmd commands to manipulate beyond the limits of the initial panel. Both explicit and pattern data sets may be used concurrently. Multiple pattern data set names are allowed. See " Catalog Search " on page 248 for information on specifying pattern names.

Use this procedure to specify the base environment.

- 1** Enter values for the Catalog Search, Sync, OutSync, and Explicit or Pattern DSNs fields. These parameters are required.
- 2** Optionally, enter values for the remaining parameters.
- 3** Press Enter to proceed to the Resolved Data Sets panel, or press End (PF3) to return to the previous menu.

Resolve Data Sets in Base

If you specified a Catalog Search on the BASE PDS panel (Cataloged PDS = Y), then the following Resolved Data Sets (BASE) panel displays with the associated Volser.

```

HPSDSYNR -----Resolved Data Sets (BASE)----- Row 1 of 4
Command =====>                                SCROLL====> CSR

Enter X to Exclude any data set.  Overtyp e Volser(s) for Uncataloged DSN(s).

Press ENTER to continue or END to exit

  Data Sets Involved in the Compare Process          Volser

_  USER298.LPARA.INSTALL                            SRQAE3
_  USER298.LPARA.LOAD                               SRQAD7
_  USER298.LPARA.MESSAGES                           SRQAD6
_  USER298.LPARA.PANELS                             SRQAE4
***** Bottom of data *****
    
```

If you specified N for Cataloged PDS and entered the explicit data set names, the Resolved Data Sets panel displays without the Volser.

The following table provides a description of each of the parameters displayed in the Resolved Data Sets (BASE) panel.

Parameter	Description
Lcmd	Type an X in the line command area to the left of each item that is to be excluded from further processing.
Resolved Data Sets	Items selected for the environment and resolved by the software display.
Volser	The DASD volume serial number where the data set is resident. This field can be changed to point to the uncataloged data sets residing on another volser.

Use this procedure to resolve the data sets in the base environment.

- 1** Type an X (Exclude) in front of any of the data sets that are to be excluded from the base environment list.
- 2** If the Volser is not displayed, enter it for each data set to be included.
- 3** Press Enter to proceed to the next panel, or press End to return to the BASE PDS panel.

If you exclude a data set from the resolved list, the list is re-positioned from that point. You may need to use the PF7 key to go back to the top of the list.

To compare a group of cataloged data sets with their corresponding uncataloged ones, type Y to resolve the names from the catalog, then overtyping the VOLSER field on the subsequent panel.

- 2 Press Enter to proceed to the next panel, or press End to return to the Resolved Data Sets (BASE) panel.

Resolve Data Sets in Target

If you specified a Catalog Search on the TARG PDS panel (Cataloged PDS = Y), then the following Resolved Data Sets (TARGET) panel displays with the associated Volser.

```

HPSDSYNR -----Resolved Data Sets (TARGET)----- Row 1 of 4
Command =====>                                SCROLL====> CSR

Enter X to Exclude any data set.  Overtyping Volser(s) for Uncataloged DSN(s).

Press ENTER to continue or END to exit

   Data Sets Involved in the Compare Process           Volser
_  USER298.LPARB.INSTALL                             SRQAE3
_  USER298.LPARB.LOAD                                SRQAE3
_  USER298.LPARB.MESSAGES                            SRQAE3
_  USER298.LPARB.PANELS                              SRQABD
***** Bottom of data *****
    
```

If you specified N for Cataloged PDS and entered the explicit data set names, the Resolved Data Sets panel displays without the Volser.

The following table provides a description of each of the parameters displayed in the Resolved Data Sets (TARGET) panel.

Parameter	Description
Lcmd	Type an X in the line command area to the left of each item that is to be excluded from further processing.
Resolved Data Sets	Items selected for the environment and resolved by the software display.
Volser	The DASD volume serial number where the data set is resident. This field can be changed to point to the uncataloged data sets residing on another volser.

Use this procedure to resolve the data sets in the base environment.

- 1 Type an X (Exclude) in front of any of the data sets that are to be excluded from the base environment list.
- 2 If the Volser is not displayed, enter it for each data set to be included.
- 3 Press Enter to proceed to the next panel, or press End to return to the TARG PDS panel.

If you exclude a data set from the resolved list, the list is re-positioned from that point. You may need to use the PF7 key to go back to the top of the list.

Results of Synchronization Verification

After fingerprinting in memory, the *Member List to Compare* screen displays. In the following illustration, out-of-sync items (members found in both groups, but having different contents) are identified.

```

HPSDSYN3                               Member List To Compare                               Row 1 of 10
Command ==>>>                               SCROLL ==>>> CSR

(S)elect one pair of members to be compared

Press ENTER to continue or END to exit

  Group Type Member      Fingerprint      Volser Data Set Name
_ B      $ADMSET  C003B356-00002968  SRSM70  USER298.LPARA.INSTALL
_ T      $ADMSET  3C9570C3-00002968  SRSM7A  USER298.LPARB.INSTALL
_ B      $AUTH    A18EB6C5-000003EA  SRSM70  USER298.LPARA.INSTALL
_ T      $AUTH    CE72A1B9-000003EA  SRSM7A  USER298.LPARB.INSTALL
_ B      $BACKUP  13DC5575-000007A4  SRSM70  USER298.LPARA.INSTALL
_ T      $BACKUP  0754211B-000007A4  SRSM7A  USER298.LPARB.INSTALL
_ B      $INIT    E272ACAD-00000997  SRSM70  USER298.LPARA.INSTALL
_ T      $INIT    408EF2F0-00000997  SRSM7A  USER298.LPARB.INSTALL
_ B      $MAINT   4FCFEC61-00001FE2  SRSM70  USER298.LPARA.INSTALL
_ T      $MAINT   5EA9991B-00001FE2  SRSM7A  USER298.LPARB.INSTALL
***** Bottom of data *****

-----
HPS268I Completed. Sync= 2382, OutSync= 10, (S)elect pairs to COMPARE.
-----

```

NOTE If there were no out-of-sync items, the BASE PDS panel displays with a message that shows the number of members that were in sync.

In the above illustration, the statistics show that ten members are out-of-sync (different contents), and 2382 members are in sync (same content). You can select a pair of members for further analysis by ChangeMan SSM, Comparex, or SuperC (depending on which comparison utility you specified on the ChangeMan SSM Defaults and User Options panel).

The following table describes the fields in the Member list to compare panel.

Field	Description
Group	Environment names from the previous panels.
Type	Load modules are designated by 'LOAD' in this area.
Member	Load or source member name.
Fingerprint	Fingerprint token.
Volser	Volume where the data set resides. This also allows support for uncataloged data sets.
Data Set Name	Data set name where the members exist.

Use this procedure to generate and view a ChangeMan SSM, Comparex, or SuperC Difference Report.

- 1 Type S by each member to select a pair to compare.

- 2 ChangeMan SSM, Comparex, or SuperC will be invoked internally to report the actual differences.

View Comparison Report

If you are using the ChangeMan SSM Built-In comparison utility, the following side-by-side comparison report is produced.

```

ISRBR0BA  USER298.SYN10319.T1431202.OUTLIST          Line 00000000 Col 001 080
Command ===>                                         Scroll ===> PAGE
***** Top of Data *****
Command      Column 10          20          30          40          50          60          7
Count-seg    ...!...!...!...!...!...!...!...!...!...!...!...!...!...
  1-1  "DATACOMP=(BASEDSN=USER298.LPARA.INSTALL,
    -2  "          BASEMEM=$ADMSET,BASEVOL=SRSM70,
    -3  "          TARGDSN=USER298.LPARB.INSTALL,
    -4  "          TARGMEM=$ADMSET,TARGVOL=SRSM7A,
    -5  "          REPORT=(SBS=Y,ADD,UPD,DEL,COMPLETE=Y))

Vol=(SRSM70)  Base=USER298.LPARA.INSTALL($ADMSET)
dsorg=P0  recfm=FB  lrecl=80  blksize=6,000

Vol=(SRSM7A)  Targ=USER298.LPARB.INSTALL($ADMSET)
dsorg=P0  recfm=FB  lrecl=80  blksize=6,000

Base  Targ    0    0    1    1    2    2    2    3          0    0    1    1    2    2    2    3
Nr    Nr    ...4...8...2...6...0...4...8...2    ...4...8...2...6...0...4...8...2
  1    1 //jobname JOB                    <> //USER298 JOB
                                     ||
                                     ||

HPS1617I  HPSDCOMP  14:31:20 Comparison Summary:
                                     0 Inserted =>
                                     0 Deleted <=
                                     1 Updated <> changed seg of updated
                                     || unchanged seg of updat
                                     0 Moved ~
                                     268 Unchanged ||

```

The differences may also be viewed using Comparex or IBM SuperC. If you installed the ChangeMan SSM interface to Comparex, refer to the *Comparex User's Guide* for more information.

To print the report, press PF3 to display the Report Disposition Panel, described in the following topic.

Print Comparison Report

This optional panel allows you to print the comparison report.

```

HPSUREP ----- Report Disposition Panel -----
Command ==>

Report Dataset Name: USER298.SYN10319.T1431202.OUTLIST
Printing Mechanism: VPSPRINT
Report Disposition ==> D          (P=Print) (K=Keep) (D=Delete)
                                   (PD=Print and delete)
                                   (JD=Submit a job to print and delete)
                                   (JK=Submit a job to print and keep)

Local Printer Id
(Or Print Class)   ==>          (For Print option)
VPSPRINT WRITER   ==>          (Applicable to VPSPRINT Only)

Job cards for submission if report disposition option above is JD or JK

==> //USER298 JOB 'SSM',
==> //          CLASS=A,NOTIFY=USER298,MSGCLASS=X
==> //*
==> //*

Press ENTER to continue; Enter END Command to exit.

```

NOTE You can bypass this panel by typing N for the Report Disposition Panel parameter in the Defaults and User Options panel (displayed by selecting Option O).

The following table describes the parameters displayed on the Report Disposition Panel:

Parameter	Description
Report Dataset Name	The name of the temporary data set dynamically allocated by ChangeMan SSM. This is a three- or four-node data set name. Depending on the installation parameters specified (number of nodes allowed in the allocation rules for temporary SMS type data sets), the outlist data set would have one of the following forms: User id.Thhmmssss.OUTLIST User id.SYnyydd.Thhmmssss.OUTLIST
Printing Mechanism	Name of the print utility (specified during installation). The options are: PRINTOFF PRINTDS VPSPRINT
Report Disposition	Specify the disposition. The options are: P - Print K - Keep D - Delete PD - Print and delete JD - Submit to print and delete JK- Submit to print and keep.

Parameter	Description
Local Printer Id	Specify the Print Class (if the Print option has been specified as the Report Disposition).
VPSPRINT WRITER	Specify the writer module name (if VPSPRINT has been specified as the print mechanism).
Job card	Job card information. This information (specified when Option O is first invoked) can be changed on this panel. Applicable only if JD or JK is specified.

Press Enter to process the request, or press End to return to the Synchronization Menu.

Option (P) - PDS Compare

This function verifies the synchrony of two partitioned data sets, physical sequential, or direct access files.

```

HPSUPDS2 ----- Compare Two Partitioned Data Sets -----
Command ==>

Base Data Set:                                Target Data Set:
  Project ==> USER14                          Project ==> USER12
  Library ==> DEV                              Library ==> DEV
  Type    ==> SRL                              Type     ==> SRL

Other Partitioned Data Set:
  Base DSN  ==>
  Target DSN ==>

Specify REPORTING options:
  Filter    ==> *      (specify member pattern or * for all members)
  Extended  ==> N      (Y/N - Include Directory Information in comparison)
  Sync      ==> N      (Y/N - Show in-sync members)
  OutSync   ==> N      (Y/N - Show out-of-sync members)
  Mismatch  ==> N      (Y/N - Show mismatched members)

Press ENTER to Compare the two data sets or END to exit.
    
```

The following table describes the parameters displayed in the panel.

Parameter	Description
Base Data Set: Project/Library/ Type	Specify the base data set name in the ISPF library.
Target Data Set: Project/Library/ Type	Specify the target data set name in the ISPF library.
Other Partitioned Data Set	Specify data set name. Enclosing the data set name in quotes indicates an explicit data set; otherwise, the TSO <i>userid</i> is prefixed.

Parameter	Description
Filter	Type * for all members in the library, or a member pattern ending with *. A pattern limits the comparison report to those members that match the member pattern.
Extended	Specify if directory information is to be included in the comparison. Y - include directory information in the comparison. N - do not consider directory information in the comparison.
Sync	Specify if synchronized items are to be reported. Y - report synchronized items (those items with equal content in both environments). N - suppress reporting synchronized items.
Outsync	Specify if non-synchronized items (items with unequal content across environments) are to be reported. Y - report non-synchronized items. N - suppress reporting of non-synchronized items Y responses to Sync and Outsync result in reporting both categories; N responses to both result in statistics only.
Mismatch	Specify if non-identical names are to be reported. Y - report non-identical names. N - suppress reporting non-identical names.

Use this procedure to compare two partitioned or sequential data sets.

- 1 Enter values for the Base Data Set, Target Data Set, Sync, Outsync, Mismatch, and Member Name fields.
- 2 Press Enter to compare the files, or press End to return to the Detection and Synchronization menu.

Option (S) - Synchronizing Environments

The next two panels generate JCL to synchronize two similar environments in a one-step process.

You specify a group of data sets as the base, and another environment as the target, after which both environments are internally fingerprinted. Any differences are identified and copied to an intermediate change basket, which is immediately imported to the target environment.

Before applying the changes to a target data set, pre-compress it in the following sequence:

- First, do the deletes.
- Next, delete the members that are to be updated.
- Finally, compress the PDS.

Parameter	Description
Reverse DSN	Type a reverse basket name if a reverse change basket needs to be created before updating the Target environment. This data set may optionally be used to undo the effects of the IMPORT operation.
Explicit or Pattern DSNs	Type a line command. The line commands are: I - Insert a blank line following the current line. R - Repeat the current line. D - Delete the current line. Specify the explicit data set names or data set patterns that comprise this environment. There are no limits to the number of data sets or patterns that may be entered for an environment. Both explicit and pattern data sets may be used concurrently. Multiple data set patterns are allowed. See " Catalog Search " on page 248 for information on specifying pattern names.

Use this procedure to specify the base environment.

- 1 Enter values for the Catalog Search and Explicit or Pattern DSNs fields.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL, or press End to return to the Synchronization menu.
The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Specify Target Environment

This panel establishes the target environment.

```

HPSUSNC2          Synchronize: TARGET Environment (2 of 2)          ROW 1 OF 1
Command =====>          SCROLL====> CSR

Enter one or more explicit or pattern DSN(s) on each line

Catalog Search to produce a list ==> Y (Y/N): Y - Resolves data sets online
Catalog Processing Commands      ==> Y (Y/N): N - Generates volume commands

Press ENTER to continue or END to exit

    Explicit or Pattern DSNs similar to ISPF 3.4
    _  USER17.LPARA_____
    -  _____

```

The following table describes the parameters in the Synchronize: TARGET Environment (2 of 2) panel.

Parameter	Description
Catalog Search	Specify if a cataloged search is to be used. Y - the data sets in the base environment are cataloged, and a catalog search will be performed to resolve the pattern. N - the data sets are not cataloged, and, therefore the specified data set names will be used in the JCL as supplied.
Catalog Processing Commands	Specify the type of commands to use to locate the data sets. Y - data sets in the base environment must be located using the catalog processing commands (BASE=GENERIC). N - volume processing commands (BASE=INCLVOL) must be used to locate the data sets.
Explicit or Pattern DSNs	Type a line command. The line commands are: I - Insert a blank line following the current line. R - Repeat the current line. D - Delete the current line. Specify the explicit data set names or data set patterns that comprise this environment. There are no limits to the number of data sets or patterns that may be entered for an environment. Both explicit and pattern data sets may be used concurrently. Multiple data set patterns are allowed. See " Catalog Search " on page 248 for information on specifying pattern names.

Use this procedure to specify the target environment.

- 1 Enter values for the Explicit or Pattern DSNs fields.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL, or press End to return to the Synchronize: Base Environment (1 of 2) panel.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Optional Change Basket Specification

This panel allows an optional change basket specification.

If a change basket DSN is specified, the change basket will be saved under that name. Additionally, the change basket will not be applied to the target environment. This allows

the basket to be reviewed before it is applied, as well as allowing the basket to be applied to multiple targets.

```

Environment Synchronization
Command ==>

Press ENTER to generate JCL to synchronize the two environments,
      - or -
Optionally, specify a Change Basket DSN to save the changes without
applying them to the TARGET environment.

      Note 1: YOU MUST VERIFY the BASE and TARGET commands to ensure
              proper association of data set names.

      Note 2: Direction of synchronization is from BASE to TARGET.

Optional Basket data set to save the exported members or entire data sets:
Change Basket DSN    ==> BASKET.DEV.Y2000.M01D11
      Comment        ==> Sync Bobs DEV with Sams
(New DSN) Unit       ==> SYSDA                      (SYSALLDA,SYSDA,...)
      Space units    ==> CYL                        (TRK,CYL,BLK)
      Primary        ==> 10                        (1-99999)
      Secondary      ==> 10                        (0-99999)

Press ENTER to proceed.

```

The following table describes the parameters on the Environment Synchronization panel.

Parameter	Description
Change Basket DSN	Type the name of a change basket data set that will hold the super directory, the actual changed members or files, and the newly-created data sets. This data set will be the input to the import function. Enclose the name in quotes for an explicit name; otherwise, the current TSO userid is prefixed. If this is a new data set, type the allocation specifications. Specifying this parameter prevents the changes from being applied. To complete the synchronization, the change basket must be applied manually (Option 6).
Comment	Type a description of the change basket (up to 40 characters). Special characters (punctuation and symbols) are not allowed.
(New DSN) Unit	Enter an esoteric name for the allocation of the new data set.
Space Units	Specify the allocation type. The options are: TRK CYL (recommended when large changes are expected). BLK Use the RLSE parameter in your JCL.

Parameter	Description
Primary	Specify primary allocation size of the data set; value 1-99999.
Secondary	Specify secondary allocation size; value 0 -99999. Allow secondary allocation for large tasks.

Use this procedure to specify the target environment.

- 1 Type values for the Change Basket DSN and Comment fields.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL, or press End to return to the Detection and Synchronization panel.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Option (A) - Advanced Operations

The advanced operations provide additional commands, parameters, and subparameters for the detection and synchronization functions. The advanced operations also generate batch JCL, which is appropriate for large environments.

```

HPSMENUMD ChangeMan SSM Detection and Synchronization - Advanced Ops (Ver 8.4.0)
Option ==>

1 Data Set Compare      - Compare Two Local Partitioned Data Sets
2 Group Compare         - Local or Remote Environments
3 Volume Compare        - Compare of two Local Volumes

4 Fingerprinting        - Large Environments
5 Create Basket         - Export Changes from the Base; FGP vs. FGP

6 Import Basket         - Import Changes to the Target
7 Undo Import          - Recover file(s) from Reverse Basket

X Exit

Press END to Exit
    
```

The following table describes the menu items:

Option	Description
1 - Data Set Compare	Generates the JCL to compare two local partitioned or sequential data sets, using different modes of comparisons.
2 - Group Compare	Generates the JCL to compare two PDSs, groups, or DASD volumes using different modes of comparison. The process can compare either two live environments, or their fingerprint representations (FGP data sets).

Option	Description
3 - Volume Compare	Generates the JCL to compare two DASD volumes using different modes of comparison. The process compares two live environments and can be restricted to certain data set types.
4 - Fingerprinting	Generates the JCL to create a fingerprint data set (FGP) for an entire DASD volume, a DASD volume subset, or any group of data sets.
5 - Create Basket	Generates JCL that references two fingerprint data sets (base and target). Comparing the base and target fingerprint data sets, it identifies the data sets and members that are different in content (EXTRACT), and then copies them to a change basket data set (EXPORT). This process packages new data sets, and identifies data sets that were deleted.
6 - Import Basket	Generates JCL to apply a change basket to a target environment. The outcome of the process can be verified using the simulate option before executing the apply option. The simulate report may be used as a change ticket.
7 - Undo Import	Generates JCL to undo an import process, utilizing the reverse change basket data set.
X - Exit	Exits the Advanced Operations menu.

Option (1) - Data Set Compare

This function generates JCL to verify the synchrony of two partitioned or physical sequential data sets, or of two direct access files.

```

HPSUPDS ----- Compare Two Partitioned Data Sets -----
Command ==>

Base Data Set:                                Target Data Set:
  Project ==> ser12                            Project ==> ser12
  Library ==> test                             Library ==> prod
  Type    ==> panels                           Type    ==> panels

Other Partitioned Data Set:
  Base DSN  ==>
  Target DSN ==>

Specify REPORTING options:
  Filter    ==> *      (specify member pattern or * for all members)
  Extended  ==> Y      (Y/N - Include Directory Information in comparison)
  Sync      ==> Y      (Y/N - Show in-sync members)
  OutSync   ==> Y      (Y/N - Show out-of-sync members)
  Mismatch  ==> Y      (Y/N - Show mismatched members)

  Saved Member Name ==> MEMTOMEM (Saved member name in the user JCL library)

```

The following table describes the parameters in the Compare Two Partitioned Data Sets panel:

Parameter	Description
Base Data Set: Project/Library/ Type	Specify the base data set name in the ISPF library.
Target Data Set: Project/Library/ Type	Specify the target data set name in the ISPF library.
Other Partitioned Data Set	Specify data set name. Enclosing the data set name in quotes indicates an explicit data set; otherwise, the TSO <i>userid</i> is prefixed.
Filter	Type * for all members in the library, or a member pattern ending with *. A pattern limits the comparison report to those members that match the member pattern.
Extended	Specify if directory information is to be included in the comparison. Y - include directory information in the comparison. N - do not consider directory information in the comparison.
Sync	Specify if synchronized items are to be reported. Y - report synchronized items (those items with equal content in both environments). N - suppress reporting synchronized items.
Outsync	Specify if non-synchronized items (items with unequal content across environments) are to be reported. Y - report non-synchronized items. N - suppress reporting of non-synchronized items Y responses to Sync and Outsync result in reporting both categories; N responses to both result in statistics only.
Mismatch	Specify if non-identical names are to be reported. Y - report non-identical names. N - suppress reporting non-identical names.
Saved Member Name	The default member name may be changed.

Use this procedure to compare two PDSs:

- 1 Enter values for the Base Data Set, Target Data Set, Sync, Outsync, Mismatch, and Member Name fields. These parameters are required.
- 2 Press Enter to generate the JCL, or press End to return to the Advanced Operations menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

Option (2) - Group Compare

This function generates the JCL to verify the synchronization between two groups of data sets, two volumes, or pools of volumes. The base and target can be represented by a local data sets or fingerprint data sets created locally or at a remote site.

For information on how to access this panel, see ["Select a Function" on page 21](#).

Specifying the Base Environment

Specify the base environment by either a fingerprint representation, a cataloged data set pattern or an uncataloged data set pattern.

```

HPSUSET1          Data Set Group Compare (BASE Environment 1 of 2)
Command =====>
Fingerprint Data Set      =====> 'somnode.LPAR.FGP'
  Select subset by DSN Pattern =====> (OPTIONAL)
  Select subset by DSN Type =====> PO,PS,DA,SRC,LOAD (OPTIONAL)
  --OR--
Cataloged Data Set Pattern =====>
  Select subset by DSN Type =====> PO,PS,DA,SRC,LOAD (OPTIONAL)
  --OR--
Uncataloged Data Set Pattern =====>
  Specify Volser Pattern =====>
  Select subset by DSN Type =====> PO,PS,DA,SRC,LOAD (OPTIONAL)

Press ENTER to continue or press End to exit.

```

To specify the Base environment:

- 1 Enter a value for the fingerprint data set (PS) field,
-or-
enter a value for the catalog data set pattern field,
-or-
enter a value for the uncatalog data set pattern field.

You must specify one of these three parameters.

- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to proceed to the next panel or press End to return to the Advanced Operations menu.

The following table describes the parameters that display in the Data Set Group Compare (BASE Environment 1 of 2) panel.

Parameter	Description
Fingerprint Data Set (PS)	Enter the name of the pre-created base environment fingerprint data set. Enclose it in quotes for an explicit name or the current TSO userid will be prefixed.
Optional DSN Pattern	A subset of data sets from the specified fingerprint data set can be selected by providing the ChangeMan SSM data set pattern capabilities. For example, SER12.DEV/ indicates the selection of all data sets where the first node is SER12 and the second node starts with the characters DEV. (See Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for additional information on DSN patterning.)
Optional DSN Type	A subset of data sets from the specified fingerprint data set can be selected by providing ChangeMan SSM's data set type selection capabilities. Example: LOAD indicates the selection of all load libraries from the fingerprint data set. More than one type specification is allowed. Separation between types is made by specifying a ",".
Cataloged Data Set Pattern	Enter the pattern of cataloged data sets to be fingerprinted. A catalog search will be performed to resolve the data sets. Further filtering can be done by specifying the data set type.
Optional DSN Type	A subset of data sets from the specified fingerprint data set can be selected by providing the ChangeMan SSM data set type selection capabilities. For example, LOAD indicates the selection of all load libraries from the fingerprint data set. More than one type specification is allowed. Separation between types is made by specifying a ",".
Uncataloged Data Set Pattern	Specify a data set pattern to be fingerprinted. The volume(s) designated on the following line will be scanned for such pattern of data sets. For example, SYS1/ indicates the selection of all data sets where the first node is SYS1 and the following node(s) can be anything. The pattern 2LINKLIB indicates the selection of all data sets where the second node is LINKLIB and the other nodes can be anything. (See Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for additional information on DSN patterning.)

Parameter	Description
Volser Pattern	Specify the explicit or pattern volser(s) to establish the base environment. A list of volsers or patterns can be specified by using "," as the separator. For example, SER001,SER002 indicates the two explicit volumes to be involved. A pool of volumes can be designated by providing "*" as the last character. For example, SER* indicates all volumes starting with the character string SER. (See Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for additional information on volume patterning.)
Optional DSN Type	A subset of data sets from the specified fingerprint data set can be selected by providing the ChangeMan SSM data set type selection capabilities. For example, LOAD indicates the selection of all load libraries from the fingerprint data set. More than one type specification is allowed. Separation between types is made by specifying a ",".

Specifying the Target Environment

Specify the target environment by either a fingerprint representation, a cataloged data set pattern or an uncataloged data set pattern.

```

HPSUSET2      Data Set Group Compare (TARGET Environment 2 of 2)
Command =====>

Fingerprint Data Set      ==> P390.USER12
  Optional DSN Pattern    ==>
  Optional DSN Type      ==> (PO,PS,DA,SRC,LOAD)

  --OR--
Cataloged Data Set Pattern ==>
  Optional DSN Type      ==> (PO,PS,DA,SRC,LOAD)

  --OR--
Uncataloged Data Set Pattern ==>
  Volser Pattern        ==>
  Optional DSN Type      ==> (PO,PS,DA,SRC,LOAD)

Optional Base HLQ ==> USER14 (Transformation Model)
  Targ HLQ ==> USER12 (Transformation Model)

Report Type ==> CUSTOM (Custom/Default)
  Member Name ==> GRPTOGRP (Saved member name in user JCL library)

Press ENTER to create the JCL or press End to exit.

```

To specify the Target environment:

- 1 Enter a value for the Fingerprint Data Set (PS) field,
-or-
enter a value for the Catalog Data Set Pattern field,
-or-
enter a value for the Uncatalog Data Set Pattern field.

You must specify one of these parameters.

- 2 Enter values for the Optional Base HLQ, Optional Targ HLQ and Report Type, Sync, and Member Name fields. These parameters are required.
- 3 Optionally, enter values for the remaining parameters.
- 4 Press Enter to generate the JCL or press End to return to the Advanced Operations Menu. If Custom is specified for Report Type, then the Reporting Options panel will be displayed.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the parameters displayed in the Data Set Group Compare (Target Environment 2 of 2) panel.

Parameter	Description
Report Type	'DEFAULT' indicates that the standard, DSNbyDSN=(SHORT=DISPLAYTOKENS=N) style report is desired. 'CUSTOM' indicates that additional reporting options should be displayed for selection.
Base HLQ	The High Level Qualifier for the BASE environment. This allows two environments with different HLQs to be compared.
Targ HLQ	The High Level Qualifier for the Target environment. This allows two environments with different HLQs to be compared.
Member Name	The default member name can be changed.

The following fields are described later in this section.

- Fingerprint Data Set
- Cataloged Data Set Pattern
- Uncataloged Data Set Pattern

Specifying the Report Parameters (Optional)

Specifying CUSTOM on the previous panel for Report Type requires that additional reporting options be specified. The data set and member options available depend on the Report Type selected.

```

HPSUSET3                               Reporting Options
Command =====>

Specify ONE report type and press enter. Then specify highlighted options.
Report Type
  Member Level  ==> Y (Y/N) - Compares members within matched data sets
  Data Set Level ==> N (Y/N) - Report only Data Set level information
  Composite     ==> N (Y/N) - Pairs members by TOKEN, identifies redundant
                               members, and sync for hard to model DSN's

Data Set Options
  DsnSync       ==> N (Y/N) - Include In-Sync Data Sets
  DsnOutSync    ==> Y (Y/N) - Include Out-of-Sync Data Sets
  DsnMismatch   ==> N (Y/N) - Include Mismatched Data Sets
  Extended Sync ==> N (Y/N) - Include Directory Information in compare
  Report Order  ==> DSNAME      (DSNAME/TOKEN) - Order by Name or Token

Member Options                               For Matched Data Sets Sync and OutSync)
  Sync          ==> N (Y/N) - Report In-Sync Members
  Outsync       ==> N (Y/N) - Report Out-of-Sync Members
  Mismatch      ==> N (Y/N) - Report Mismatched Members
  DisplayTokens ==> N (Y/N) - Show Fingerprint Tokens

Press ENTER to create the JCL or press End to exit.

```

To specify the custom parameters:

- 1 Enter a value of Y for one of the Report Type fields, then press Enter.
- 2 Specify the parameters for the highlighted data set and member options.
- 3 Press Enter to generate the JCL or press End to return to the Target Environment (2 of 2) menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the parameters displayed in the *Reporting Options* panel.

Parameter	Description
Report Type:	Specify one of Member Level (DSNbyDSN), Data Set Level (DSNtoDSN), or COMPOSITE mode of comparisons, as described below.
Member Level	DSNbyDSN mode indicates that, between the two environments, all corresponding data set names should be considered in the verification process. Data set A, for example, is matched with data set A. If the two data set tokens are identical, both data sets are exactly identical. A summary report for each pair of data sets is produced. If the two <i>data set tokens</i> (data sets A on both volumes in this example) are different, a MEMtoMEM mode will be triggered to identify the members which are out-of-sync.

Parameter	Description
Data Set Level	DSNtoDSN mode indicates that, between the two environments, all corresponding data set names, or TOKENS should be considered in the verification process. Data set A, for example, is matched with data set A. If the two data set tokens are identical, both data sets are exactly identical. A summary report for each pair of data sets, and mismatched data sets is provided.
COMPOSITE	COMPOSITE mode indicates that data set member <i>tokens</i> on both environments are the sole criterion for synchronization verification. Once the base and target Fingerprints tokens are retrieved from the existing Fingerprint Data Sets, or are just created, they are sorted internally in memory. Tokens in the base are checked against the tokens in the target. Once two identical tokens are found, the members that own those tokens have the same contents, and therefore are considered <i>in-sync</i> . Identically named members with different tokens are <i>out-of-sync</i> . Appropriate reports can be requested by combining <i>match</i> and <i>mismatch</i> with <i>sync</i> and <i>outsync</i> .
DsnSync	Enter Y or N. Specifying Y for DsnSync indicates the reporting of synchronized data sets, those data sets with equal content in both environments. N suppresses reporting of synchronized data sets.
DsnOutSync	Enter Y or N. Specifying Y for DsnOutSync indicates the reporting of non-synchronized data sets, those items with unequal content across environments. N suppresses reporting of non-synchronized data sets.
DsnMismatch	Enter Y or N. Specifying Y for DsnMismatch indicates the reporting of non-matching data set names. N suppresses reporting of mismatched data sets.
Extended Sync	Enter Y or N. Specifying Y for Extended Sync indicates that directory information is considered during the comparison. N suppresses directory information during the compare.
Report Order	Type DSNAME or TOKEN. For the Data Set Level report, the data sets can be listed in DSNAME order, or TOKEN order. Token order lists identical data sets together, even if they have inconsistent naming conventions.
Sync	Enter Y or N. Specifying Y for Sync indicates the reporting of synchronized items, those items with equal content in both environments. N suppresses reporting of items.

Parameter	Description
OutSync	Enter Y or N. Specifying Y for Outsync indicates the reporting of non-synchronized items, those items with unequal content across environments. N suppresses reporting of items.
Mismatch	Enter Y or N. Specifying Y for Mismatch indicates the reporting of non-identical member names. N suppresses reporting of mismatched members.
DisplayTokens	Enter Y or N. Specifying Y for DisplayTokens indicates the reporting of member tokens. N suppresses reporting of member tokens, and instead reports Sync, OutSync, and Mismatch.

Option (3) - Volume Compare

Volume Compare generates the JCL to verify the synchronization of two entire DASD volumes, or a subset of their data sets.

```

HPSUVOL                               Volume Compare
Command ==>
Specify the Explicit BASE and TARGET Volser: to be Compared. Type is optional, and
      if specified, it restricts the selection of data sets on the volume. Separate
      more than one Type using commas.

BASE Volser:                           TARGET Volser:
VOL ==> RES001                          VOL ==> ARES01 (explicit)
Type ==>                                  Type ==>      (PO,PS,DA,LOAD,TEXT, SRC)

Specify COMPARISON method and REPORTING options:
DSNbyDSN ==> N      DSNtoDSN ==> Y
Sync ==> N          OutSync ==> Y
Mismatch ==> Y
Member ==> VOLTOVOL (Saved member name in user JCL library)
Press ENTER to generate the JCL or press End to exit.

```

To compare two DASD volumes:

- 1 Type values for the Base Vol, Target Vol, Sync, Outsync, Mismatch, and Member Name fields. These parameters are required.
- 2 Type a Y value for the DSNbyDSN field or the DSNtoDSN field. One or the other must be provided, but not both. (mutually exclusive)
- 3 Optionally, enter values for the remaining parameters.
- 4 Press Enter to generate the JCL or press End to return to the Synchronization Menu.
The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the parameters displayed in the DASD Volume Compare panel.

Parameter	Description
Base VOL	Specify an explicit volser to establish the base environment.
Base Type	A subset of data sets from the specified volume can be selected by using the ChangeMan SSM data set type selection capabilities. For example, LOAD indicates the selection of all load libraries on the volumes. If no value is specified, all non-VSAM data sets will be selected for the fingerprinting process.
Target VOL	Specify an explicit volser to establish the target environment.
Target Type	A subset of data sets from the specified volume can be selected by using the ChangeMan SSM data set type selection capabilities. For example, LOAD indicates the selection of all load libraries on the volumes. If no value is specified, all non-VSAM data sets will be selected for the fingerprinting process.
DSNbyDSN	DSNbyDSN mode indicates that between the two environments, all corresponding data set names should be considered in the verification process. Data set A in the base, for example, is matched with data set A in the target. If the two <i>data set tokens</i> are identical, both data sets are identical. A summary report for these data sets is produced. If the two <i>data set tokens</i> (data sets A on both volumes in this example) are found to be different, a MEMtoMEM mode will automatically be triggered to identify the members that are out-of-sync.
DSNtoDSN	DSNtoDSN mode indicates that <i>data set tokens</i> on both environments should be verified to determine which data sets, in their entirety, are in-sync or out-of-sync. Name correspondence will not be performed. This mode is used only when naming conventions on both volumes are different, and, name transformations (MODEL), can not easily be defined. If data set associations can be established using the MODEL command, the DSNbyDSN along with the MODEL provides a much better comparison report.
Sync and Outsync	Enter Y or N. Specifying Y for Sync indicates the reporting of members that have <i>identical contents</i> , in both environments, regardless of their names. N suppresses reporting of items in that category. Specifying Y for Outsync indicates the reporting of members that have differences in their <i>contents</i> . N suppresses reporting of items in this category. Positive responses to both results in reporting both categories, negative responses to both results in statistics only.

Parameter	Description
Match and Mismatch	Enter Y or N. Specifying Y for Match indicates the reporting of <i>identical member names</i> , in both environments, regardless of their contents. N suppresses reporting of items in that category. Specifying Y for Mismatch indicates the reporting of <i>non-identical names</i> . N suppresses reporting of items in this category. Positive responses to both results in reporting both categories, negative responses to both results in statistics only.
Member	The default member name can be changed.

Option (4) - Fingerprinting

This function generates the JCL to create a Fingerprint Data Set to represent the content for a group of data sets, an entire DASD volume, or a pool of volumes.

For information on how to access this panel, see ["Select a Function" on page 21](#).

```

HPSUFGP                Create Fingerprint Data Set
Command =====>
Cataloged Data Set Pattern      =====> somnode.CMNSSM.V730
  (Optional) Select a subset by Type =====>                (PO,PS,DA,SRC,LOAD)
  --OR--
Uncataloged DSN Pattern or '/' for all =====>
  Explicit Volser or Pattern      =====>
  (Optional) Select a subset by Type =====>                (PO,PS,DA,SRC,LOAD)
Specify the Fingerprint Data Set to save the results:
  Fingerprint DSN                =====> somnode.FGP.CMNSSM.V730
  GDG or New DSN allocation =====>                        (SYSALLDA,SYSDA,...)
    Space units                   =====>                (TRK,CYL,BLK)
    Primary                       =====>                (1-99999)
    Secondary                      =====>                (0-99999)
Member Name                      =====> FGPDSGRP      (Saved member name in the user JCL library)
Press ENTER to generate the JCL or press End to exit.

```

To create a Fingerprint Data Set:

- 1 Type a value for the Catalog Data Set Pattern field,
-OR-
enter values for the Uncatalog Data Set Pattern and Fingerprint DSN fields.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL or press End to return to the Advanced Operations Menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the parameters displayed in the Create a Fingerprint Data Set panel.

Parameter	Description
Cataloged Data Set Pattern	Enter the pattern of cataloged data sets to be fingerprinted. This facility works the same as ISPF Option 3.4. A catalog search is performed to resolve the data sets. Further filtering can be performed by specifying the data set type (below).
Optional DSN Type	A subset of data sets can be selected by providing the ChangeMan SSM data set type selection capabilities. For example, LOAD indicates the selection of all load libraries. More than one type specification is allowed. Separation between types is made by a ",".
Uncataloged Data Set Pattern	Specify a data set pattern to be fingerprinted. The volumes designated on the following line 'volser pattern' are scanned for such pattern of data sets. For example, SYS1/ indicates the selection of all data sets where the first node is SYS1 and the following nodes can be anything. The pattern 2LINKLIB indicates the selection of all data sets where the second node is LINKLIB and the other nodes can be anything. (For more information, refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243.)
Volser Pattern	Specify the explicit or pattern volsers to establish the Base environment. A list of volsers or patterns can be specified by using "," as the separator. For example, the specification of SER001,SER002 indicates these two explicit volumes are to be scanned. A pool of volumes can be designated by providing "*" as the last character. For example, SER* indicates all volumes starting with the string SER. (For additional information, refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243.) NOTE The \$\$EXCL skeleton will be imbedded in the generated JCL. This skeleton generates the EXCLDSN parameter which specifies additional data sets to be automatically excluded from volume processing.
Optional DSN Type	A subset of data sets can be selected by providing the ChangeMan SSM data set type selection capabilities. For example, LOAD indicates the selection of all load libraries. More than one type specification is allowed. Separation is made by a ",".
Fingerprint DSN	Enter the name of a fingerprint data set. Enclose it in quotes for an explicit name or the current TSO userid is prefixed. If this is a new data set specify the allocation parameters.
(New DSN) Unit	Enter the esoteric name for the allocation of the new data set.
Space Units	Specify TRK, CYL or BLK for the allocation type of the new data set. CYL is recommended when an entire DASD volume is to be fingerprinted.

Parameter	Description
Primary	A value in the range 1-99999 can be specified for primary allocation size of the data set.
Secondary Units	A value in the range 0-99999 can be specified for secondary allocation size of the data set.
Member	The default member name can be changed.

Option (5) - Creating a Basket

There are two methods by which data set changes can be extracted and exported into a change basket.

- Method 1 assumes that corresponding pairs of data set exist prior to beginning the Extract process.
- Method 2 does not assume the data sets are associated and lets you specify them online before beginning the extract process.

For information on how to access this panel, see ["Select a Function" on page 21](#).

Method 1

This function generates the JCL to compare two fingerprint data sets (Base and Target) to determine the changed members, newly found data sets, and scratched data sets. Once the extract completes, the changes are exported to a Change Basket data set. At the beginning of the Change Basket data set, there is a Super Directory control area that indicates what has been unloaded from the base and what will be imported to the target environment.

```

HPSUEXT1          Create Basket (Extract and Export)
Command =====>

BASE Fingerprint DSN  ==>
TARG Fingerprint DSN  ==>

Basket dataset to save the exported members or entire datasets:
  Basket DSN          ==> BASKET.DSN
  Comment             ==> YOUR COMMENT HERE
  (New DSN) Unit      ==>                               (SYSALLDA,SYSDA,...)
  Space units        ==>                               (TRK,CYL,BLK)
  Primary            ==>                               (1-99999)
  Secondary          ==>                               (0-99999)

Specify the EXTRACT method:
  DSNbyDSN          ==> Y          Method 1 (Extract changes from matching datasets )
  New DSN           ==> Y          (Extract new datasets )
  Scratch DSN       ==> Y          (Scratch target datasets not in base )

  NbyONE           ==> N          Method 2 (Association of datasets to be specified)

ENQ                ==> NO   Y/N (Utilize enqueue processing)
Member             ==> EXT$EXP (Saved member name in the user JCL library)

Press ENTER to create the JCL or press END to exit.

```

To Extract and Export differences using Method 1:

- 1 Enter values for the Base Fingerprint DSN, Target Fingerprint DSN, Basket DSN, and Member Name fields. These parameters are required.
- 2 Method 1: Type Y for DSNbyDSN and N for NbyOne.
- 3 Optionally, enter values for the remaining parameters.
- 4 Press Enter to generate the JCL or press End to return to the Synchronization Menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the parameters displayed in the Synchronize (Extract and Export) panel.

Parameter	Description
Base Fingerprint Data Set	Type the name of a fingerprint data set representing the Base environment. This is the changed (new) environment from which the changes will be extracted. Enclose it in quotes for an explicit name or the current TSO userid is prefixed.
Target Fingerprint Data Set	Type the name of a fingerprint data set representing the Target environment. (This represents the 'before change' environment.) Enclose it in quotes for an explicit name or the current TSO userid is prefixed.
Basket DSN	Type the name of a Change Basket data set to hold the Super Directory and the actual changed members or files and newly created data sets. This data set is used in later steps as the input to the import function. Enclose it in quotes for an explicit name or the current TSO userid is prefixed. If this is a new data set, input the allocation specifications.
Comment	Use this field to type a comment for the Change Basket. The comment can be up to 40 characters long. If the following characters are used, the comment must be enclosed in quotes: , * = ()
(New DSN) Unit	Type an esoteric name for the allocation of the new data set.
Space Units	Specify TRK, CYL, or BLK for the allocation type of the new data set. CYL allocation is recommended for large tasks.
Primary	Specify a value in the range 1-99999 for primary allocation size of the data set.
Secondary	Specify a value in the range 0-99999 for secondary allocation size of the data set. Allow secondary allocation for large tasks.

Parameter	Description
DSNbyDSN	Type Y or N. Specifying Y for this field indicates a DSNbyDSN extraction. This method of extract relies on the existence of corresponding data set names in both environments. For example, data set A in the base is paired with data set A in the target and then the out-of-sync modules are extracted from the base and copied to the Change Basket data set, for a later Import to the target. Deleted and renamed modules are reflected in the Super Directory portion of the Change Basket data set.
New DSN	Type Y or N. Specifying Y causes data sets that exist in the base, but not in the target, to be included in the change basket.
Scratch DSN	Type Y or N. Specifying Y includes scratch requests for data sets that exist in the target but not in the base. When the basket is applied, the data sets at the target site will be deleted.
NbyONE	Type Y or N. Specifying Y indicates a NbyONE extraction. This method of extract relies on the user to specify the data set associations. If this method is selected a subsequent panel displays for the user to determine the association. This method can be used for an N : 1 synchronization mode, in which N data sets in the base need to be synchronized with 1 data set in the target. An M : N synchronization can also be accomplished when the user defines a model for data set name associations.
ENQ	Type Y to enable enqueue processing. Type N to disable enqueue processing.
Member	The default member name can be changed.

Method 2

This panel is presented if the extract mode NbyONE was specified on the previous panel.

```

HPSUEXT2          Association of Non-paired Data Sets
Command =====>
Specify the Target Data Sets and the Source Data sets. Source data sets may be
multiple, while EACH target must be single (N : 1 relations).

TO   =====> SER12.PROD.SRC _____
FROM =====> SER12.TEST.SRC1 _____
=====> SER12.TEST.SRC2 _____
TO   =====> SER12.PROD.LINKLIB _____
FROM =====> SER12.TEST.LINKLIB _____
=====> _____
TO   =====> SER12.PROD.PANELS _____
FROM =====> SER12.TEST.PANELS _____
=====> _____
TO   =====> SER12.PROD.MSGS _____
FROM =====> SER12.TEST.MSGS _____
=====> _____
Press ENTER to generate the JCL or press End to exit.

```

To extract and export a data set using Method 2:

- 1 Enter values for the To and From fields. These parameters are required.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL or press End to return to the Synchronization Menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table describes the parameters displayed in the Association of Non-paired Data Sets panel.

Parameter	Description
TO	Specify the fully qualified target data set name. Do not enclose the name in quotes.
FROM	Specify the fully qualified base data set names from which the changed PDS members or files are extracted and copied to a Change Basket data set. Do not enclose the name in quotes.

NOTE This panel allows for a limited number of data sets to participate in the synchronization. The extract function, however, is not limited. Once the sample JCL is created, the user can edit the job to specify all data sets involved in the synchronization task.

Option (6) - Importing a Change Basket

This function generates the JCL to Import a Change Basket into a Target environment. The Change Basket data set is created by the Create Basket function described in Option 5.

For information on how to access this panel, see ["Select a Function" on page 21](#).

```

HPSUIMP                                Import the Basket
Command =====>

The Basket containing all changed members and/or entire new data sets.
Basket Data Set Name  =====> somnode.BASKET

Import RUN mode       =====> SIMULATE (SIMULATE/APPLY)
Use the Default settings =====> N (Y/N)
ENQ (Enqueue processing) =====> N (Y/N)
Member Name          =====> IMPORT (Saved member name in user JCL library)

Other Optional Parameters,                                     Default
Delete from Target, non-existing members in the Base         =====> Y (Y)
Scratch unmatched data sets in the target                     =====> Y (Y)
Catalog any newly-found data sets from the Base               =====> Y (Y)
Replace existing target data sets                             =====> N (N)
Force automatic conflict resolution                           =====> N (N)
Reverse Change Basket                                         =====> N (N)
Precompress target libraries before import                    =====> Y (Y)
Allocate: Specify '*' to use the catalog, or provide a Volser override:
    For old data sets in the target:                          =====> * (*)
    For new data sets in the target:                           =====> * (*)

Press ENTER to create the JCL or press End to exit.

```

To import a Change Basket:

- 1 Specify values for the mandatory fields in the panel.
- 2 Optionally, enter values for the remaining parameters.
- 3 Press Enter to generate the JCL or press End to return to the Synchronization Menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table provides a description of each of the parameters displayed in the Import a Change Basket panel.

Parameter	Description
Basket Data Set Name	Type the name of a Change Basket data set created from the Base environment. This data set contains a Super Directory, followed by the changed/added PDS members or files. Enclose it in quotes for an explicit name or the current TSO userid is prefixed.
Import RUN Mode	Type SIMULATE or APPLY. The first mode (SIMULATE), processes the Super Directory and simulates the outcome of an APPLY. This process generates a report that can be used as a change ticket. The second mode (APPLY) copies the changed members being exported from the Base environment. IEBCOPY or its functional equivalent (such as PDSFAST) is used to copy the PDS members to the target libraries.
Use the Default Settings	Type Y to set the IMPORT parameters to the defaults specified under <i>other optional parameters</i> . A specification of N in this field allows the user to change the defaults.
ENQ (enqueue processing)	Type Y to enable enqueue processing. Type N to disable enqueue processing.
Member Name	The default member name can be changed.
Delete from Target	N is the default. This parameter internally generates a DELETE parameter for the IMPORT function. If there are designated delete members from the target environment, by request, such members will be deleted. If this parameter is specified as N, such members will not be deleted from the target environment. Note: The DELETE parameter can be used even if you specified SIMULATE for the Import Run Mode.
Scratch from Target	Y is the default. This parameter directs ChangeMan SSM to scratch data sets during IMPORT processing that were identified as unmatched (found in the Base, but not in the Target) when the Change Basket was created. An N specification prevents the unmatched data sets from being scratched during IMPORT processing.
Catalog new data sets from Base	Y is the default. This parameter directs ChangeMan SSM to catalog, at the Target environment, the newly found data sets in the base. During the IMPORT process, newly found data sets in the change basket will be cataloged only if this parameter has been specified.

Parameter	Description
Replace existing target data sets	N is the default. If a data set in a change basket exists in the Target, an N prevents the existing data set from being overwritten. Y causes it to be replaced with the data set in the change basket.
Force Automatic Conflict Resolution	N is the default. This parameter internally generates a FORCE parameter for the IMPORT function. The conflicts determined by ChangeMan SSM will be forced. For example, if orphaned members are found in the Base environment, the process will not continue until these conflicts are resolved. Or, if the fingerprint token of a data set or a member has changed since the Export of the data, at the base site, these members will not be imported unless REJECT or FORCE are specified. Note: The FORCE parameter can be used even if you specified SIMULATE for the Import Run Mode.
Reverse Change Basket	Specify Y to create a Reverse Change Basket. The Reverse Change Basket name is the name of the Change Basket, with a final node of REVERSE. This option allows the changes made by applying the Change Basket to be reversed using Option 7. Specify N to not create the Reverse Basket.
Precompress target libraries before import	Y is the default. Y triggers a compress of the library before applying changes from the basket. An N does nothing before applying changes.
Allocate	For Old Data Sets and New Data Sets in the target: <ul style="list-style-type: none"> ■ Specify '*' to use the catalog to find data sets in the Target environment to which to apply changes from the Change Basket. ■ Specify a volser to apply changes from the Change Basket only to data sets residing on that volume.

Option (7) - Undo an Import

This panel creates the JCL to UNDO previously imported updates.

NOTE To use this facility, the IMPORT/APPLY process must have been performed with the specification that a Reverse Basket be generated.

For information on how to access this panel, "[Select a Function](#)" on page 21.

```
HPSUUNDO          UNDO the Import Operation
Command =====>
Reverse the update, or recover selected from the Reverse Change Basket.

Specify the Reverse Basket DSN  ===> somnode.BASKET.REVERSE

Optional Specifications:
  File to recover ('/' for all) ===> /
  Simulate Run or Apply         ===> APPLY      (APPLY OR SIMULATE)

Member Name      ===> REVERSE   (Saved member name in user JCL library)
Press ENTER to create the JCL or press End to exit.
```


To undo the updates:

- 1 Specify the Reverse Basket DSN.
- 2 Optionally, enter a specific file to recover
- 3 Optionally, specify Simulate
- 4 Press Enter to generate the JCL or press End to return to the Synchronization Menu.

The JCL can be edited, saved (SAVE), submitted (SUB) for execution, or cancelled (CAN).

The following table provides a description of each of the parameters displayed in the Import a Change Basket panel.

Parameter	Description
Reverse Basket Data Set Name	Specify a Reverse CHANGE BASKET to be applied to the DSN or Volser specified. The files will be restored to the state that they were at when the Reverse Change Basket was created.
File to Recover	I indicates all data set updates will be undone. A DSN pattern ending in I limits the Undo to those data sets.
Import RUN Mode	Type SIMULATE or APPLY. The first mode (SIMULATE), processes the Super Directory and simulates the outcome of an APPLY. This process generates a report that can be used as a change ticket. The second mode (APPLY) copies the changed members being exported from the Base environment. IEBCOPY or its functional equivalent (such as PDSFAST) is used to copy the PDS members to the target libraries.
Member	The default member name can be changed.

Detection and Synchronization for HFS

Choose Option H from the ChangeMan SSM Main Menu to display the main menu for HFS Detection and Synchronization. This option provides you with several functions for detecting changes and synchronizing data specifically for HFS files.

```

HPSMENUH ----- Detection and Synchronization - HFS Menu -----
Option ==>

1  Fingerprint          - Fingerprint an HFS environment
2  Compare              - Compare two HFS environments
3  Create Basket        - Extract changes to a Change Basket
4  Import Basket        - Apply changes from a Change Basket
5  Undo Import          - Recover file(s) from a Reverse Basket
6  View Basket          - Display HFS Basket contents
7  View Fingerprint     - Display HFS environment attributes

X  Exit

Press END to Exit

```

This table describes the six options listed on this panel.

Option	Description
Fingerprint	Generates the JCL to fingerprint an HFS file, or pattern of files.
Compare	Generates the JCL to compare two HFS environments represented by fingerprint data sets.
Create Basket	Generates the JCL to create an HFS change basket based on differences between two fingerprint data sets.
Import Basket	Generates the JCL to import an HFS change basket.
Undo Import	Generates the JCL to undo changes by using a reverse change basket.
View basket	Displays the contents of an HFS basket or a set of baskets.
View Fingerprint	Displays HFS environment attributes.
Exit	Returns to the previous panel.

Option (1) - Fingerprint

Choose Option 1 from the ChangeMan SSM Detection and Synchronization - HFS menu to display the Fingerprint HFS Files panel.

This option generates the JCL to fingerprint an HFS file, or pattern of files. This is the first step in synchronizing two software environments. The Fingerprint data set can be used to represent a remote environment, or an environment at a point in the past.

```

HPSHFGP1                Fingerprint HFS files                SCROLL==> CSR
Command ==>

Specify the fingerprint data set to save the results:
Data Set Name ==>

NEW Allocation ==>      (SYSALLDA,SYSDA,...)
  Space units ==>      (TRK,CYL,BLK)
  Primary ==>          (1-99999)
  Secondary ==>        (0-99999)

Member Name ==> HFSFGP  (Saved member name in the user JCL library)

  Explicit file or Pattern of HFS files to fingerprint:
  _____

Recursive ==> NO      (Yes/No) Include subdirectories?

Press ENTER to continue or END to exit
    
```

This table provides descriptions for all elements on the Fingerprint HFS files panel.

Field	Description
Data Set Name	Specify a physical sequential file to retain the fingerprint tokens. It may be either a NEW or pre-allocated data set.
NEW Allocation Parameters	Specify the space allocation for a NEW fingerprint data set.
Member Name	Specify the member name where the generated JCL will be saved.
Explicit file or pattern to fingerprint	Specify the name of the HFS file, or file pattern, to be fingerprinted. If the recursive option is used, files within sub-directories matching the pattern will be fingerprinted. The only pattern supported is a trailing asterisk, for example: <code>/home/user/*</code> .
Recursive	Specify YES to include subdirectories in the fingerprint data set. Specify NO to fingerprint only files in the specified directory.

Option (2) - Compare

Choose Option 2 from the ChangeMan SSM Detection and Synchronization - HFS menu to display the Detect HFS Changes panel.

This option generates the JCL to compare two HFS environments represented by fingerprint data sets, and reports in-sync, out-of-sync, and mismatched HFS files.

```

HPSHCMP1                                Detect HFS Changes
Command ==>>

Specify the BASE and TARGET Fingerprints to view changes between the two
Fingerprinted environments.

Explicit or pattern Fingerprint data sets:
  BASE (new) ==> USER002.**.FGP*
  TARGET (old) ==> *

Model From ==>
  To ==>

Member Name ==> HFSCMP

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
      similar to ISPF 3.4. If * is specified alone, the Fingerprint DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit
    
```

This table provides descriptions for all elements on the Detect HFS Changes panel.

Field	Description
BASE and TARGET Fingerprint Data Sets	Specify explicit or pattern fingerprint data set names for the BASE and TARGET environments. Do not put quotes around the names, but enter them as explicit fingerprint data set names (DSNs) or as DSN patterns similar to ISPF 3.4. See "Catalog Search" on page 248 for information on specifying pattern names.
Model	Specify the HFS model transform fields: From – specify a string to match in the base environment paths. This pattern is replaced by the string specified in the To field. To – specify a string to replace the one in the From field. This model allows file paths in the base environment to match those in the target environment.
Member Name	Specify the member name where the generated JCL will be saved.

Option (3) - Create Basket

Choose Option 3 from the ChangeMan SSM Detection and Synchronization - HFS menu to display the Create HFS Change Basket panel.

This option generates the JCL to create an HFS change basket based on differences between two fingerprint data sets.

```

HPSHCRE1                Create HFS Change Basket
Command ==>>>

Specify the BASE and TARGET Fingerprints to package changes between the two
Fingerprinted environments.

Explicit or pattern Fingerprint data sets:
  BASE (new) ==> IMACROB.**.FGP*
  TARGET (old) ==> *
Model From ==>
  To ==>

Change Basket DSN ==>
Basket Comment ==>
  NEW allocation ==>          (SYSALLDA,SYSDA,...)
  Space units ==>          (TRK,CYL,BLK) - Primary ==>          (1-99999)
                                   - Secondary ==>          (0-99999)
  Member Name ==> HFSCRE (Saved member name in the user JCL library)

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
      similar to ISPF 3.4. If * is specified alone, the Fingerprint DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit
    
```

This table provides descriptions for all elements on the Create HFS Change Basket panel.

Field	Description
BASE and TARGET Fingerprint Data Sets	Specify explicit or pattern fingerprint data set names for the BASE and TARGE environments. Do not put quotes around the names, but enter them as explicit fingerprint data set names (DSNs) or as DSN patterns similar to ISPF 3.4. See "Catalog Search" on page 248 for information on specifying pattern names.
Model	Specify the HFS model transform fields: From – specify a string to match in the base environment paths. This pattern is replaced by the string specified in the To field. To – specify a string to replace the one in the From field. This model allows file paths in the base environment to match those in the target environment.
Change Basket DSN	Specify the name of a new or existing Change Basket to be used for packaging changes. If the basket is new, specify the allocation parameters below. For an explicit data set name, place the name in quotes.
Basket Comment	Specify a comment for the change basket. This comment will be included within the change basket data set for future reference.

Field	Description
NEW Allocation Parameters	Specify the space allocation for a NEW Change Basket.
Member Name	Specify the member name where the generated JCL will be saved.

Option (4) - Import Basket

Choose Option 4 from the ChangeMan SSM Detection and Synchronization - HFS menu to display the Import HFS Change Basket panel.

This option generates the JCL to import and apply the changes in an HFS change basket.

```

HPSHIMP1          Import HFS Change Basket
Command ==>>>

Specify the explicit or pattern name of the change basket to import.

Change Basket DSN ==> *
Apply or Simulate ==> APPLY      (APPLY/SIMULATE)
Force             ==> NO        (YES/NO)
Model From ==>
To               ==>

Reverse Basket DSN ==>

NEW allocation ==>      (SYSALLDA,SYSDA,...)
Space units   ==>      (TRK,CYL,BLK) - Primary ==>      (1-99999)
                                     - Secondary ==>      (0-99999)
Member Name   ==> HFSIMP (Saved member name in the user JCL library)

NOTE: The Basket DSN can be specified explicitly, or as a DSN pattern
      similar to ISPF 3.4. If * is specified alone, the Basket DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit

```

This table provides descriptions for all elements on the Import HFS Change Basket panel.

Field	Description
Change Basket DSN	Enter the name of the Change Basket being imported. Do not enclose the DSN in quotes, but enter them as explicit fingerprint data set names (DSNs) or as DSN patterns similar to ISPF 3.4. See " Catalog Search " on page 248 for information on specifying pattern names.
Apply or Simulate	Specify SIMULATE to see what changes are contained in the basket, without modifying the local environment. Specify APPLY to synchronize the environments.
Force	Specify YES to force the changes from the change basket. Specify NO to skip changes when the target environment is not as expected.

Field	Description
Model	Specify the HFS model transform fields: From – specify a string to match in the base environment paths. This pattern is replaced by the string specified in the To field. To – specify a string to replace the one in the From field. This model allows file paths in the base environment to match those in the target environment.
Reverse Basket DSN	Specify the name of a new or existing Change Basket to be used for packaging changes. If the basket is new, specify the allocation parameters below. For an explicit data set name, place the name in quotes.
NEW Allocation Parameters	Specify the space allocation for a NEW Reverse Change Basket.
Member Name	Specify the member name where the generated JCL will be saved.

Option (5) - Undo Import

Choose Option 5 from the ChangeMan SSM Detection and Synchronization - HFS menu to display the Reverse HFS Change Basket panel.

This option generates the JCL to apply the changes in an HFS reverse change basket. A specific file or pattern may be specified to limit the import.

```

HPSHREV1                Reverse HFS Change Basket
Command ==>>>

Specify the explicit or pattern name of the reverse change basket.

Reverse Basket DSN ==> *
Apply or Simulate ==> APPLY      (APPLY/SIMULATE)

Include File            ==>

    Member Name        ==> HFSREV  (Saved member name in the user JCL library)

NOTE: The change basket DSN can be specified explicitly, or as a DSN pattern
      similar to ISPF 3.4. If * is specified alone, the Fingerprint DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit
    
```

This table provides descriptions for all elements on the Reverse HFS Change Basket panel.

Field	Description
Reverse Basket DSN	Enter the name of the Change Basket being imported. Do not enclose the DSN in quotes, but enter them as explicit fingerprint data set names (DSNs) or as DSN patterns similar to ISPF 3.4. See " Catalog Search " on page 248 for information on specifying pattern names.
Apply or Simulate	Specify SIMULATE to see what changes are contained in the basket, without modifying the local environment. Specify APPLY to synchronize the environments.
Include File	Specify the name or name pattern to be applied from the reverse change basket.
Member Name	Specify the member name where the generated JCL will be saved.

Option (6) - View Basket

Choose Option 6 from the ChangeMan SSM Detection and Synchronization - HFS menu to display the SSM View HFS Basket panel.

This option generates the JCL to create an HFS change basket based on differences between two fingerprint data sets.

```

HPSHBSKV ----- SSM View HFS Basket (Ver 8.4.0) -----
Command ==>>

Specify an explicit Basket DSN or Basket DSN pattern.

Basket DSN or pattern ==> *

NOTE: The Basket DSN can be specified explicitly, or as a DSN pattern
      similar to ISPF 3.4. If * is specified alone, the Basket DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit
    
```

This table provides descriptions for all elements on the SSM View HFS Basket panel.

Field	Description
Basket DSN or Pattern	Name of a basket to be viewed. This may be entered as an explicit basket data set name (DSN) without quotes, or as a DSN pattern similar to ISPF 3.4. See " Catalog Search " on page 248 for information on specifying pattern names. If a pattern is specified, a list of basket DSNs is presented. If * is specified, the Basket DSN patterns selection table displays.

Option (7) - View Fingerprint

Choose Option 7 from the ChangeMan SSM Detection and Synchronization - HFS menu to display HFS Fingerprint data sets (DSNs).

```

HPSHFGPV ----- View HFS Fingerprint Data Set -----
Command ==>

Specify an explicit Fingerprint DSN or Fingerprint DSN pattern.

Fingerprint DSN (like 3.4) ==> *

NOTE: The Fingerprint DSN can be specified explicitly, or as a DSN pattern
      similar to ISPF 3.4. If * is specified alone, the Fingerprint DSN
      patterns selection table will be displayed.

Press ENTER to continue or END to exit
    
```

This table provides descriptions for all elements on the View HFS Fingerprint Data Set panel.

Field	Description
Fingerprint DSN (like 3.4)	<p>Name of a fingerprint DSN to be viewed.</p> <p>This may be entered as an explicit fingerprint data set name (DSN) without quotes, or as a DSN pattern similar to ISPF 3.4.</p> <p>See "Catalog Search" on page 248 for information on specifying pattern names.</p> <p>If a pattern is specified, a list of fingerprint DSNs is presented.</p> <p>If * is specified, the Fingerprint DSN patterns selection table displays.</p>

Chapter 3

Batch Detection and Synchronization

The ChangeMan SSM Batch Detection and Synchronization Facility includes commands for use in JCL batch jobs to perform Detection and Synchronization functions on large scale environments. The resulting fingerprint data sets and reports can then be used to detect the changes in the software environments (a package, a large group of data sets, an entire DASD, or a DASD pool), and to synchronize software environments locally or remotely.

This chapter discusses each of the detection and synchronization commands including examples for proper use.

Environment Comparison	100
Compare Environments with Mask	102
Environment Synchronization	109
Redundancy Management	111
DCB Information	111

Environment Comparison

One or more local or remote environments can be compared using the fingerprinting technology of ChangeMan SSM. Following are the primary commands used in the environment comparison process. The commands are listed alphabetically.

- AUTORECALL
- BASE
- COMPARE
- MASK
- MODEL
- SETPRINT
- TARGET
- EXTENDED SYNC
- ORPHAN CHECK

The following commands are the support functions of the comparison process:

- LIST
- NOECHO
- ONLINE
- TRACE

The table below provides a description of each of the environment comparison commands.

Command	Description
AUTORECALL	Directs ChangeMan SSM to recall a migrated data set before continuing with a task. This function can be used with any system locate function issued by ChangeMan SSM.
BASE	Fingerprints a software environment, or establishes an environment representation by loading a previously generated fingerprint data set. This command is also used to save the fingerprint tokens in an external fingerprint data set.
COMPARE	Compares two software environments live or by using their fingerprint representations. This command is used to verify the synchrony between multiple local or remote environments.
EXTENDED SYNC	Includes the ChangeMan SSM directory token to determine the differences.
LIST	Reports the member or data set tokens in the base or target environments. This is used primarily to obtain a detailed member level report of the fingerprint data set. LIST can also be used to report the alias or orphaned members in an environment.

Command	Description
MASK	Allows you to mask certain user specified areas of the source members before generating the token. Also, if properly specified, embedded compile date/time may be masked.
MODEL	Establishes proper naming associations between two environments.
NOECHO	Suppresses the echoing of the supplied commands.
ONLINE	Requires the volumes specified in the list to be online.
ORPHANCHECK	Reports orphaned members during the comparison. An orphan is an alias member without the primary member.
SETPRINT	Controls the report contents.
TARGET	Fingerprints a software environment, or establishes an environment representation by loading a previously generated fingerprint data set. TARGET is also used to save the fingerprint tokens in an external fingerprint data set.
TRACE	Suppresses or activates the type T messages.

Remote Environments and Firewalls

Two software environments may be compared directly if both are available to the process.

If one or both are remote, the fingerprint representations must be made available to the compare process. The fingerprint compare does not introduce integrity issues with a software environment protected behind a firewall.

Fingerprint data sets can also be used to detect changes to a reference environment, over a period of time. The fingerprint data set is a physical sequential file, with the following DCB attributes:

```
LRECL=512, BLKSIZE=4096, DSORG=PS, RECFM=FB
```

Scope and Size of Environment

If you use the volume processing command (INCLVOL=pattern) to define an environment of data sets for fingerprinting, and if you use the * wildcard to indicate the entire data center DASD pool, an insufficient storage condition (ABEND 80A-10) may occur if there are more than 200 volumes in the pool.

Consider using either the catalog processing (GENERIC) facility, or limit the volume pattern. A large region size is required to process a very large pool.

Compare Environments with Mask

In the standard fingerprinting process, data is passed to the algorithm without masking its content. Embedded compile dates and times, for example, are included in the standard generation of fingerprint tokens. Typically, if two identical source modules are compiled and link-edited on two different dates, the resulting load modules are detected as being out-of-sync, even though they are logically identical. Because the macro instructions specifying the date and time (*&sysdate*, *&stime*, and COBOL *Date-Compiled*) can be placed in varying locations in the source, the resultant load modules can contain this information at varying locations. Furthermore, the various compilers may produce object code that varies the format of the generated date and time.

The masking facility identifies and excludes selected data strings in load modules, IMS-generated control blocks, or other text files when creating fingerprint tokens as specified by masking rules.

The facility applies the masking rule in each environment. For example, when fingerprinting load modules, the masking facility interrogates the load module at the CSECT level to determine the type of compiler used, and then invokes the proper compile date/time mask.

Masking facilities are provided for the following:

- IMS control blocks (ACB and MFS commands).
- Text strings (STRING command).
- Columns and Rows (AREA command).
- Language format rules (LANGUAGE command).
- Upper and lower case sensitivity (CASE command).
- Ignore blanks, spaces, and other white space (SQUEEZE command).

Masking is encoded internally in a generated fingerprint data set. Therefore, when loading two fingerprint data sets generated with masking, the masking must be identical.

If you are loading two fingerprints, only one of which was generated with masking, and if you want masking set for both, you must load the fingerprint with masking first.

Load Module Masking

If the MASK command is specified, ChangeMan SSM applies the specified masking rules to the environments designated by the BASE and TARGET commands.

The MASK command appears only once in the command stream; it is in effect for all subsequent BASE and TARGET commands. If masking is required, you must specify the MASK command at the beginning of the //hpsin dd* command stream.

For example:

```
MASK=(CSECT,DATE,TIME)
BASE=DSN=CMNSSM.V820.LOADLIB
TARG=DSN=CMNSSM.V810.LOADLIB
COMPARE=MEMTOMEM
BASE=RESET
TARG=RESET
```

```
BASE=DSN=CMNSSM.V820.LOADLIB  
TARG=DSN=CMNSSM.V810.LOADLIB  
COMPARE=MEMTOMEM
```

The MASK command indicates that CSECT parsing and automatic embedded DATE/TIME masking is in effect for both sets of base and target environments.

Application Control Block Masking (ACB)

If this MASK command is specified, ChangeMan SSM applies the IMS Application Control Block (ACB) masking rules to the environment designated by the BASE and TARGET commands.

For example:

```
MASK=ACB  
  
BASE=DSN=somnode.ACB.MONDAY  
TARG=DSN=somnode.ACB.TUESDAY  
COMPARE=MEMTOMEM
```

The MASK command indicates that the automatic ACB masking rule is to be applied to both the base and target environments.

Message Format Services Masking

If this MASK command is specified, ChangeMan SSM applies the IMS Message Format Services (MFS) masking rules to the environment designated by the BASE and TARGET commands.

For example:

```
MASK=MFS  
  
BASE=DSN=somnode.MFS.AUG  
BASE=SAVE=DSN=MFS.AUG.FGP
```

The MASK command in this example indicates that the automatic MFS masking rule is to be applied during the generation and saving of the fingerprint data set.

Subsequently, the fingerprint data set can be referenced to detect the out-of-sync, deleted, and added modules.

Group Masking

When the MASK command is specified, it directs ChangeMan SSM to apply the appropriate rule to the relevant data sets in a group.

If the environment to be fingerprinted is composed of ACB-type, MFS-type, or standard load libraries, the libraries are identified by ChangeMan SSM during processing, and the proper rule is applied to each type of load module.

If MASK=MFS or MASK=ACB is specified, MASK=CSECT is implied.

For example:

```
MASK=(MFS,ACB,CSECT,DATE,TIME)

BASE=GENERIC=DSN=somnode.CMNSSM.*
TARG=LOAD=DSN=AUG.FGP.MASKALL
COMPARE=DSNBYDSN=SHORT
```

This MASK command indicates that all masking rules are in effect. Rules are applied to the libraries in the group as required.

A fingerprint data set created in August (TARG) using all masking rules, is now used as a reference to detect the out-of-sync modules (BASE).

Search Strings and Masking

If a MASK command is specified, ChangeMan SSM masks user-defined strings in the source data sets. The masking rules apply to the environments designated by the BASE and TARGET commands.

There can be multiple MASK commands in the command stream. They are in effect for all the subsequent BASE or TARGET commands.

The MASK command is optional; if used, it must be specified at the beginning of the command stream, before the environment is defined by the BASE or TARGET commands.

In this example, the first MASK command indicates the string to search for, and if found, masks it with blanks.

The next MASK command squeezes multiple consecutive blanks into a single blank.

The SHOW parameter reports the masks that are in effect.

The mask rule applies to both the base and target environments.

```
MASK=STRING=(VALUE='string1',
              LOBOUND=4,
              HIBOUND=72)

MASK=(SQUEEZE,SHOW)

BASE=DSN=SYCNTRAC.V820.SRC
TARG=DSN=SYCNTRAC.V810.SRC
COMPARE=MEMTOMEM

BASE=DSN=SYCNTRAC.V820.MAC
TARG=DSN=SYCNTRAC.V810.MAC
COMPARE=MEMTOMEM
```

In this example, the first MASK command specifies the string to search for. Once the string is found, the string itself, along with four bytes before the string and three bytes following the string are masked with blanks.

The second MASK command looks for the string HPS in the last eight bytes. LOBOUND=1 and HIBOUND=-1 define the entire area where the string is to be searched for. This is the default setting.

The last MASK command squeezes multiple blanks into a single blank.

SOURCE or TEXT masking does not apply to the load libraries encountered in the specified BASE and TARGET patterns. These commands are automatically ignored for load libraries encountered in the base or target environments.

```
MASK=STRING=(VALUE=X'0111',
             BEFORE=4,
             AFTER=3)
```

```
MASK=STRING=(VALUE='HPS',
             LOBOUND=73,
             HIBOUND=80)
```

```
MASK=(SQUEEZE, SHOW)
```

```
BASE=GENERIC=DSN=somnode.SYCINTRAC.V820.*
TARG=GENERIC=DSN=somnode.SYCINTRAC.V810.*
```

```
COMPARE=MEMTOMEM=(BASEDSN=somnode.SYCINTRAC.V820.SRC,
                  TARGDSN=somnode.SYCINTRAC.V810.SRC)
```

```
COMPARE=MEMTOMEM=(BASEDSN=somnode.SYCINTRAC.V820.PANEL,
                  TARGDSN=somnode.SYCINTRAC.V810.PANEL)
```

Area Masking

Area Masking indicates masking of one or many areas of fixed or variable records.

The first MASK command specifies the area to be masked. Columns 73 through 80 are masked with blanks.

The second MASK command squeezes out the white spaces and displays the commands in effect.

```
MASK=AREA=(COL=73, LEN=8)
```

```
MASK=(SQUEEZE, SHOW)
```

```
MODEL=(NAME=MOD1,
       TRANSFORM=(2TEST=>PROD))
```

```
BASE=GENERIC=DSN=somnode.TEST.JCL.*
TARG=GENERIC=DSN=somnode.PROD.JCL.*
COMPARE=DSNBYDSN=(MODEL=MOD1)
```

The first two MASK commands specify the areas to be masked.

If specified, + indicates the location relative to the beginning of the record, and - indicates a location relative to the end of the record.

```
MASK=AREA=(COL=1, LEN=2)
MASK=AREA=(COL=-8, LEN=8)
MASK=(SQUEEZE, SHOW)
```

```
BASE=DSN=somnode.TEST.JCL.*
TARG=DSN=somnode.PROD.JCL.*
COMPARE=COMPOSITE
```

Language Masking

This category of masking applies a set of pre-defined masking rules to the source or text data sets found in the designated base or target environments.

The languages supported are COBOL, Assembler, and JCL, each with its own set of masking rules.

SQUEEZE masking is applied automatically for all language masking.

The pre-defined mask rules are:

COBOL:

Columns 1 - 6 and 73 - 80 are masked with blanks.
Comments are masked in entirety.
White spaces are squeezed.

Assembler:

Columns 73 - 80 are masked with blanks.
Comments are masked in entirety.
White spaces are squeezed.

JCL:

Columns 73 - 80 are masked with blanks.
SYSIN data is not involved in the masking process.
White spaces are squeezed.

The MASK command indicates that COBOL masking rules should be applied to all source data sets found in the base and target environments.

You must verify that all data sets involved in the process are COBOL.

```
MASK=LANG=COBOL

MODEL=(NAME=MOD1,
        TRANSFORM=(2TEST=>PROD))

BASE=DSN=somnode.TEST.COB*
TARG=DSN=somnode.PROD.COB*
COMPARE=DSNBYDSN=(MODEL=MOD1)
```

General Masking

A general masking rule may be enforced with any other masking rule in effect.

If SQUEEZE is specified, after the mask rules have been applied, multiple consecutive blanks are squeezed into a single blank before the record is passed to the fingerprinting process. The SQUEEZE mask can be specified standalone, or with other mask rules.

The CASE=UPPER parameter causes an internal translation of records to uppercase before the record is passed to the fingerprinting process. The default parameter is CASE=MIXED. No translation is performed if this default mode is in effect.

The MASK command translates lowercase characters to uppercase, and also squeezes out the white spaces in the records.

The base environment represents a group of production source data sets that is to be compared to a similar environment to identify the differences.

In the first command stream, the environment is fingerprinted and the resulting tokens and the masks are stored in the fingerprint data set.

```
MASK=(CASE=UPPER,SQUEEZE,SHOW)
BASE=GENERIC=DSN=somnode.SRC.*
BASE=SAVE=DSN=somnode.SRC.FGP
```

Later, this fingerprint data set (that represents the production source) is loaded as a reference to be compared to the source environment on another LPAR.

```
BASE=LOAD=DSN=somnode.SRC.FGP
TARGET=GENERIC=DSN=somnode.SRC*
```

```
COMPARE=DSNBYDSN=SHORT
```

No explicit masking has been specified in the comparison function.

The implicit masking rules embedded in the loaded fingerprint data set force the live environment to be masked accordingly. When a BASE or TARGET command loads a fingerprint data set, the masking rules will be controlled by the embedded masks stored in the fingerprint data set.

Group Masking

If the environment to be fingerprinted is composed of load libraries and source, the source masking is applied only to the source data sets. This feature allows for fingerprinting a software package, each with the appropriate source or load masking.

The MASK commands define the masking rules for the entire software package. Rules are applied to the libraries in the group as needed. The fingerprint data set created will contain the masking rules in effect.

```
MASK=(COL=73,LEN=8)
MASK=(SQUEEZE,SHOW)

BASE=GENERIC=DSN=somnode.FRAS.*
BASE=SAVE=DSN=somnode.FRAS.FGP
```

Fingerprint Data Set Safeguards

When a fingerprint data set is created, ChangeMan SSM stores the masking rules that were in effect at the time of the fingerprinting process in the fingerprint data set.

Subsequently, when this fingerprint data set is used (loaded in the BASE or TARGET command), the same masking rules are enforced. If MASK=NONE (the default) is specified, the rules are not enforced.

This safeguard feature prevents a potential user error, and provides assurance that masking rules are the same. This is important, especially when a fingerprint data set generated at one site is to be used at another site or LPAR by a different user who may not know what masking rules were in effect at the time of the creation.

To view the embedded masking rules in a fingerprint data set, load the fingerprint data set using the BASE or TARGET command.

The MASK command indicates that the automatic Message Format Services masking rule was in effect when the fingerprint data set was created, and will be in effect for the data sets specified in the BASE and TARGET commands.

Both environments must refer to the same masking rules. If you attempt to load the fingerprint data set with no masking (MASK=NONE), the job is halted with an "incompatible masking rules" message.

```
MASK=MFS

BASE=LOAD=DSN=somnode.MFS.AUG.FGP
TARG=DSN=somnode.MFS.SEP
COMPARE=MEMTOMEM
```

In this example, an explicit mask is not specified.

When two fingerprint data sets are loaded, they must have been created with the same embedded masking rules. If the fingerprint data sets have different masking rules, the job will be stopped with an "incompatible mask rules" message.

If you intend to accept the masks anyway, you must specify the ACCEPT parameter. ACCEPT allows the loading of a fingerprint data, set even if the embedded maskings are incompatible.

An example of legitimate, yet incompatible masks is when the first fingerprint data set has an embedded masking of:

```
MASK=STRING=(VALUE='HPS')
```

while the second has a different embedded masking:

```
MASK=STRING=(VALUE='SER').

BASE=LOAD=(DSN=somnode.CICS.SITE1.FGP,ACCEPT)
TARG=LOAD=(DSN=somnode.CICS.SITE2.FGP,ACCEPT)
SETPRINT=(SYNC=NO,MATCH=NO)
COMPARE=DSNBYDSN=SHORT
```

Performance

The CSECT parsing, DATE/TIME masking, and ACB or MFS masking options incur extra processing overhead. Compared to the standard processing, (MASK=NONE), approximately twice the CPU time is required. ChangeMan SSM uses its full range of performance boosts, including full-track or full-cylinder reads. For EXPORT/IMPORT operations, the standard fingerprinting process must be the default: token data set

Environment Synchronization

You can synchronize two local or remote environments with fingerprinting technology.

You can establish a fingerprint representation of an environment before and after the change. The two environment fingerprints can then be compared to EXTRACT and EXPORT. Once the change basket has been created, it can be transferred to other environments and imported to the target system.

In addition to the ChangeMan SSM standard member-level token that represents the content of the file (source or load), another token is generated for the directory entry of each member. This 4-byte token is always generated by the fingerprinting process.

During the environment comparison or synchronization, this member-level token identifies the members that have differences in their user fields (such as creation date, change date, or version), or load module LINKEDIT attributes (such as AC, RENT, RUSE, or link edit date).

This feature is recommended; it is invoked only if the EXTENDED=YES command has been specified.

Following are the primary commands used in the environment synchronization process.

MODEL	TARGET	IMPORT
ENQ	EXTRACT	EXTENDED
BASE	EXPORT	VOLMAP

The following commands are the support functions of the comparison process:

ORPHANCHECK	NOECHO
ONLINE	TRACE

The table below describes the environment synchronization commands.

Command	Description
BASE	Fingerprints a software environment, or establishes an environment representation by loading a previously-generated fingerprint data set. Saves the fingerprint tokens in an external fingerprint data set.
EXPORT	Invokes IEBCOPY or PDSFAST to offload the changed members into a single physical sequential file for a later IMPORT at one or more target environments. Directory information is transferred to make the target system identical to the base.
EXTENDED	EXTENDED SYNC=YES directs ChangeMan SSM to evaluate the member-level directory tokens in addition to the member-level content token in its evaluation of change. If specified, it is used during the compare and the synchronization functions. EXTENDED SYNC can be abbreviated as EXTENDED.

Command	Description
EXTRACT	Identifies and extracts the differences between two environments. EXTRACT produces a Super Directory in memory that is used to offload the updated files or PDS members.
IMPORT	Applies a change basket to one or more target environments. During the APPLY, a reverse change basket can be created to allow the selective undo of data set or member updates. A SIMULATE parameter enables you to project the effects of applying a received change basket. Regressions are identified and controlled at your discretion.
MODEL	Associates data set names according to user-specified name transformations. This allows two or more environments, having different naming conventions, to be synchronized with minimal effort.
NOECHO	Suppresses the echoing of the specified ChangeMan SSM commands.
ONLINE	Specifies that the volumes identified in the list to be online for ChangeMan SSM processing to continue.
TARGET	Fingerprints a software environment, or establishes an environment representation by loading a previously-generated fingerprint data set. TARGET also saves the fingerprint tokens in an external fingerprint data set.
TRACE	Suppresses or activates the type T messages.
VOLMAP	Allows for volume association. At the time of import, data sets are allocated in a DASD pool according to this mapping table. This is an optional parameter.

Change Basket

This is a physical sequential file that has a super directory in the beginning, followed by the offloaded data. The super directory, created at the time of export, is used as a road map during the import process. The change basket contains all types of offloaded data, such as source, load modules, physical sequential, and direct access files.

The DCB attributes of this file are:

```
LRECL=0, BLKSIZE=4096, DSORG=PS, RECFM=U
```

Transferring the Change Basket

Some data transfer software may insert data at the end of an incomplete data block. If ChangeMan SSM detects this condition, a "Change Basket is Damaged" or "Data Continuation Block is incomplete" message is reported.

To handle this situation, ChangeMan SSM supplies the program SERCOPY to compress and convert a change basket into a fixed block, record length 80 file. This prevents other products from inserting trailing data at the end.

At the target environment, SERCOPY expands and converts the file back to its original format, ready for the import. Refer to the *SENDBSK* and *RECVBSK* samples jobs in the *SAMPLES* data set for a setup.

Once the change basket is converted to fixed blocked records, it can be transferred to any platform (MVS, PC, LAN, or Internet) and then downloaded to MVS as a binary file.

Redundancy Management

A group of concatenated libraries, data sets, a DASD volume, or a large pool of volumes can be fingerprinted to identify files with identical contents. This identifies data sets or PDS members with identical contents, regardless of their names. Such data sets or members can be evaluated for possible deletion. Redundant data requires additional administration, causes confusion, violates auditing requirements, and occupies DASD space.

The following table identifies and describes the redundancy management commands.

Command	Description
CLOCK	Allows for a time-controlled, volume-level fingerprinting process.
DASDRED	Produces tokens for the data sets and members for the environment identified by this command.
RESTART	Allows for restart of token generation process.

Fingerprint Token Data Set

As data set members are tokenized, the token values are written to an external physical sequential data set. This file has a checkpoint data that allows for restart. The DCB attributes for the token data set are:

```
LRECL=1000 , BLKSIZE=6000 , DSORG=PS , RECFM=VB
```

DCB Information

The following table provides the DCB information for all synchronization component related files.

Data Set Type	DCB Attributes
Fingerprint	LRECL=512 , BLKSIZE=4096 , DSORG=PS , RECFM=FB
Change Basket	LRECL=0 , BLKSIZE=4096 , DSORG=PS , RECFM=U
Reverse Change Basket	LRECL=0 , BLKSIZE=4096 , DSORG=PS , RECFM=U
VSAM Fingerprint	LRECL=0 , BLKSIZE=4096 , DSORG=PS , RECFM=U
VSAM Change Basket	LRECL=4096 , BLKSIZE=4096 , DSORG=PS , RECFM=F

Data Set Type	DCB Attributes
VSAM Reverse Change Basket	LRECL=4096 , BLKSIZE=4096 , DSORG=PS , RECFM=F
Scratch List	LRECL=80 , BLKSIZE=4000 , DSORG=PS , RECFM=FB
Token	LRECL=1000 , BLKSIZE=6000 , DSORG=PS , RECFM=VB
DETECT DD SYNCTRAC DD	LRECL=133 , BLKSIZE=6650 , DSORG=FBA , RECFM=PS
'userid.SSM.LOG'	LRECL=133 , BLKSIZE=6650 , DSORG=FBA , RECFM=PS

Chapter 4

Commands

This chapter describes formatting and parameters for Detection and Synchronization commands.

Command Structure	114
Synchronization Commands	119
HFS Synchronization Commands	201

Command Structure

The general format for the batch commands in ChangeMan SSM is as follows:

```
Command=(Parameter=(Subparameter=(value,...)),Parameter...)
```

There are no spaces in the commands. The spacing shown in the examples is for clarity only.

Subparameter values must be fully specified on a single line. You can specify multiple subparameter values on separate, consecutive lines.

Enclose the parameters in parentheses. If there is only one parameter in the command, you may omit the parentheses.

The commands and command parameters can be abbreviated by specifying the minimum characters required to uniquely identify them.

TRUE, YES, and ON are synonymous, as are FALSE, NO, and OFF.

Command Summary

```
AUTORECALL    = ( y | n )

BASE          = ( dsn | generic | load | reset | save | vol )

CLOCK        = ( time-in-minutes )

COMPARE      = ( composite | dsnbydsn | dsntodsn |
                filter | memtomem )

DASDRED      = ( exclvol | inclvol,
                excldsn | incldsn,
                dd | dsn,
                type )

ENQ          = ( y | n )

EXPORT       = ( dd | dsn, comment = ' ',
                excldsn,
                incldsn,
                haltonerr,
                members)

EXTENDEDSYNC = ( y | n )

EXTRACT      = ( all,
                dsnbydsn | nbyone ,
                newvol,
                newdsn,
                scrdsn)
```

```

HFSBASE      = ( file=('_', '_') | load=_ | reset | save=_ ,
                recurse,
                include=('_', '_'),
                exclude=('_', '_') )

HFSCOMPARE   = ( file,
                include=('_', '_'),
                exclude=('_', '_'),
                model=_ )

HFSEXPRESS   = ( basketdsn=_, | basketdd=_,
                include=('_', '_'),
                exclude=('_', '_'),
                comment=_ )

HFSEXTRACT   = ( include=('_', '_'),
                exclude=('_', '_'),
                all | delete, update, add,
                model=_ )

HFSIMPORT    = ( basketdsn=_, | basketdd=_,
                reversedsn=_, | reversedd=_,
                model=_,
                force,
                include=('_', '_'),
                exclude=('_', '_'),
                reverse )

HFSMODEL     = ( name=_,
                transform=('_' => '_' ) )

HFSTARGET    = ( file=('_', '_') | load=_ | reset | save=_ ,
                recurse,
                include=('_', '_'),
                exclude=('_', '_') )

IDRLOAD

IMPORT       = ( apply | simulate,
                basketdd | basketdsn ,
                catlg | indirect = (catlg1, catlg2, ... ) ,
                symbolic=_,
                delete,
                excludsn | incldsn,
                force,
                haltonerr,
                model,
                newsntovol,
                olddsntovol,
                precompress,
                reject | members,
                replace | noreplace,
                reverse,
                reversdd | reversdsn ,
                scratch )

```

LIST = (base | target)

MASK = (acb
area,
case,
csect,
date,
language,
mfs,
show,
squeeze,
string,
time)

MODEL = (name,
transform)

NOECHO

ONLINE = (volser,volser,...)

ORPHANCHECK = (y | n)

REFTARG

RESTART = (dd | dsn)

SETPRINT = (detail,
dir,
dsnassociation,
dsnismatch,
dsnoutsync,
dsnsync,
exceptiononly,
excldsn,
incldsn,
mismatch ,
multiple ,
notincldsn,
outsync ,
rejdsn,
single ,
sync,
vtoc,
vvds)

SIMULATE

SMSMAP = (STORCLASS=(basevalue1,targetvalue1),
STORCLASS=(basevalue2,targervalue2),
MGMTCLASS=(basevalue,targetvalue),
DATACLASS=(basevalue1,targetvalue1),
DATACLASS=(basevalue2,targetvalue2))

TARGET = (dsn | generic | load | reset | save | vol)

TRACE = (on | off)

```

VIEWFGP      = (base | targ )

VIEWBASKET   = basketdd | basketdsn

VOLMAP       = (volser=(oldvol1,newvol1),
               volser=(oldvol2,newvol2),
               volser=(oldvol3,newvol3),...)

```

Command Stacking

Command stacking is provided for enhanced performance, improved flexibility, and simplicity. Stacked commands are executed in the order of entry. An example of command stacking follows:

```

BASE=VOL=SER=IPLAA2
TARG=VOL=SER=IPLBB2
*
SETPRINT=(DSNSYNC=Y,DSNOUTSYNC=N,DSNMISMATCH=N)
COMPARE=DSNTODSN=(DSNAME,DISPLAYTOKEN=N)
*
SETPRINT=EXCEPTIONSONLY
COMPARE=DSNTODSN=TOKEN

```

In the above example, the BASE command includes all data sets on volume IPLAA2. The software derives fingerprints for all members in each of the partitioned data sets selected and one composite token for each entire partitioned data set, physical sequential data sets, and direct access files. The TARGET command performs a similar function on the other volume.

The first SETPRINT command requests a report of all data set names in the two volumes that are identical in content (that have identical data set fingerprint tokens).

While the fingerprints representing the base and target are still available in memory, a second SETPRINT command is issued with a different COMPARE request. The second COMPARE command directs the software to consider the data set fingerprint tokens. Variances in the data set tokens indicate a change in the data set. Only those data sets that changed are reported.

The advantage of stacking commands allows users to obtain different views with minimum overhead because the overhead of fingerprinting the base and target environments for a second time is avoided.

Reserved DDs

DD	Description:
HPSIN	All commands input to ChangeMan SSM must be provided through the specification of this DD statement.
HPSPLIB	This DD points to the PARMLIB data set where the source PARM members HPSPARMx are found. At least one PARM member (HPSPARMS) must be created. This DD is required only if the default PARMLIB data set and the default PARM member are not going to be used. In this case, it must be specified in the \$SYNC CLIST.
HPSSCR	A temporary data set is required for IMPORT functions when data sets are required to be scratched at the target location.
HPSWORK	A temporary data set required for any job that uses the commands EXPORT or IMPORT. Five cylinders of work space is sufficient for most tasks.
HPSWORK2	A temporary data set required for any job that creates a change basket. Five cylinders of work space is sufficient for most tasks.
DETECT or SYNCTRAC	This DD differentiates the output generated by ChangeMan SSM from output generated by other called programs (such as IEBCOPY). The presence of a //DETECT DD causes the messages produced by ChangeMan SSM to be placed in the data set pointed by this DD. Other messages produced by the system, or the IEBCOPY invocations are placed in the SYSPRINT data set. Because repetitive invocations of IEBCOPY causes the SYSPRINT file to be opened and closed multiple times, the disposition of this file is set to MOD (refer to the member (EXPGDG) for an example). This DD, if present, points to a standard physical sequential file or a GDG with the following attributes: (RECFM=PS, DSORG=FBA, LRECL=133, BLKSIZE=6650)
SYSPLEX	IMPORTANT! Do not use this DD if you will be configuring ChangeMan SSM to use RLS. This DD points to the common file used by HPSRTO when in a multiple LPAR environment or SYSPLEX. The presence of this DD enables SYSPLEX or multiple LPAR operations. This DD points to a physical sequential file, which is common to all HPSRTO started tasks. The number of tracks in this file should be equal to the number of HPSRTO started tasks running. This DD must be in the SSMSTAR started task for each LPAR operating with the shared set of databases.

Comments

An asterisk in the first column indicates comments in the HPSIN DD. This facility improves the readability when there is a complex set of commands. An example of comments in the synchronization function follows:

```
//STEP EXEC PGM=HPSTRANS,REGION=4096K
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR
// DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//FGPDD0 DD DISP=SHR,DSN=somnode.FGP0
//FGPDD1 DD DISP=SHR,DSN=somnode.FGP1
//BASKET DD DISP=SHR,DSN=somnode.BASKET
//SYSIN DD SPACE=(TRK,(1,1)),DISP=(,DELETE),UNIT=SYSDA
//HPSWORK DD SPACE=(CYL,(5,5)),DISP=(,DELETE),UNIT=SYSDA
//HPSWORK2 DD SPACE=(CYL,(5,5,100)),DISP=(,DELETE),UNIT=SYSDA,
// DCB=(RECFM=U,BLKSIZE=27998)
//HPSIN DD *
* -----
* Compare the new FGP of the volume after maintenance to
* identify the updates and export them to a Change Basket.
* -----
BASE=LOAD=DD=FGPDD1 /* FGP OF VOL001 FEB */
TARG=LOAD=DD=FGPDD0 /* FGP OF VOL001 JAN */
EXTRACT=ALL
EXPORT=(DD=BASKET,COMMENT='MAINT BASKET')
```

Using REXX Programs

The HPSTRANS batch program can also be called from a REXX program. For an example, see the REXXTRAN sample program in the ChangeMan SSM SAMPLES library.

Synchronization Commands

Synchronization component commands are described in detail in the following sections. Descriptions of each of the parameters and subparameters, and examples are provided. For more information regarding the commands, such as structure, formatting, and stacking, see ["Command Structure" on page 114](#).

AUTORECALL Command

The AUTORECALL command allows ChangeMan SSM to recall a migrated data set before continuing with a task.

HSM-migrated data sets and SAMS:DISK archived data sets are detected by this command.

Use AUTORECALL for any system locate function issued by ChangeMan SSM.

Specifying the AUTORECALL command does not perform a recall of the fingerprint data set specified in a BASE=LOAD=fgp_dsname.

The general format of the AUTORECALL command is:

```
AUTORECALL = ( Y | N )
```

This table describes the AUTORECALL command parameters.

Parameter	Description
YES	Indicates ChangeMan SSM should wait before continuing with an operation for a migrated data set to be recalled. Specify AUTORECALL=Y for ChangeMan SSM to identify any migrated data sets associated with the task, recall them from the archive system, and then proceed with the requested task.
NO	Indicates ChangeMan SSM should not attempt to recall migrated data sets.

AUTORECALL must be the first command specified. Comments can be entered before the command.

If AUTORECALL=Y is specified, the data set names or patterns of the migrated data sets must be exactly and unambiguously specified.

AUTORECALL Command Example 1

In this example, if a data set resolved by the pattern has been archived, the fingerprinting task requested by the BASE or TARGET commands waits until the data set is recalled.

AUTORECALL must be the first command specified.

```
AUTORECALL=Y
BASE=VOL=(INCLVOL=RESDD1),
        EXCLDSN=SYS1.BROADCAST)
TARG=VOL=(INCLVOL=RESDD2),
        EXCLDSN=SYS1.BROADCAST)
COMPARE=DSNBYDSN=SHORT
```

AUTORECALL Command Example 2

In this example, the three data set patterns are fingerprinted. If the fingerprint data set referenced in the third command was archived, the task waits until the data set is recalled from HSM or SAMS:Disk storage system.

```
AUTORECALL=Y
BASE=GENERIC=(DSN=(SYS1,SYS2,SYS3))
BASE=SAVE=(DSN=somnode.SYS.FGP)
```

BASE Command

The BASE command fingerprints a software environment, or establishes an environment representation by loading a previously-generated fingerprint data set. This command also saves the fingerprint tokens in an external file.

During the fingerprinting process, if an alias data set is specified, the true physical data set is resolved from the catalog and fingerprinted instead.

Aliases allow you to point to a dynamic data set name at any time. They also alleviate the need to change the JCL when a new version of software is installed.

BASE is invoked with a single parameter only. You can use multiple BASE commands to establish the base environment.

The general format of the BASE command follows:

```
BASE = ( DSN | GENERIC | LOAD | RESET | SAVE | VOL )
```

The following table describes the BASE command parameters.

Parameter	Description
DSN	Specifies explicitly cataloged data sets. Data sets specified by this parameter are fingerprinted first, before any other processing. See " DSN Parameter " on page 123 for additional details.
GENERIC	Performs a catalog search to establish the environment. See " GENERIC Parameter " on page 123 for additional details.
LOAD	Recalls (LOAD) a previously-generated fingerprint data set. When used in the BASE command, it establishes the base environment. See " LOAD Parameter " on page 126 for additional details.
RESET	Resets the base environment. Any subsequent BASE command reestablishes the base environment rather than appending to an existing one. You can define multiple environments (using the BASE command), clear (with the RESET command), or redefine. See " RESET Parameter " on page 129 for additional details.
SAVE	Specifies that the tokens of the current base environment be saved in a fingerprint data set. See " SAVE Parameter " on page 130 for additional details.
VOL	Specifies all DASD volumes that are to participate in data set selection for fingerprinting at the volume level. Data sets selected by this parameter are fingerprinted first before any further processing. See " VOL Parameter " on page 131 for additional details.

BASE Command Example 1

This BASE command resolves the pattern CMNSSM.V810 from the catalog. The TYPE=SRC subparameter designates the inclusion of the source libraries in the base environment. Omitting TYPE indicates that the request is for including all non-VSAM data sets.

Once the data sets are selected, they are fingerprinted.

```
BASE=GENERIC=(DSN=CMNSSM.V810,TYPE=(SRC))
```

BASE Command Example 2

This BASE command specifies that two volumes (VOL001 and VOL003) are to be searched, selecting all data sets with the zero (right-most) qualifier of LINKLIB for fingerprinting.

```
BASE=VOL=(INCLVOL=(VOL001,VOL003),
          INCLDSN=0LINKLIB)
```

BASE Command Example 3

The following BASE command specifies that all volumes with a volser beginning with the generic VOL are to be searched, selecting all data sets with a type determined to be SRC (Source). This includes all partitioned data sets other than load libraries RECFM=U.

```
BASE=VOL=(INCLVOL=(TST*),
          TYPE=SRC)
```

BASE Command Example 4

This BASE command specifies that four explicit data sets are to be fingerprinted.

```
BASE=DSN=(CMNSSM.LOADLIB,
          CMNSSM.SKELS,
          CMNSSM.PANELS,
          CMNSSM.MSGS)
```

BASE Command Example 5

This stack of three BASE commands:

```
BASE=VOL=(INCLVOL=(TST*), INCLDSN=(/), TYPE=(PO))
BASE=VOL=(INCLVOL=(DEV*), INCLDSN=(/), TYPE=(PS))
BASE=SAVE=DSN=LPARA.FGP
```

specifies the following:

- Search all volumes where the volser begins with TST, selecting and fingerprinting all data sets of type PO (partitioned).
- Search all volumes where the volser begins with DEV, selecting and fingerprinting all data sets of type PS (physical sequential).
- The result of the second BASE command is merged into the result of the first BASE command, and the combined result is saved on the fingerprint data set (LPARA.FGP) that now represents the contents of the selected data sets in the LPARA (Base) environment.

BASE Command Example 6

The following BASE command specifies that the environment will be represented by the LPARA.FGP fingerprint data set.

This pattern selects those data sets that have LINKLIB as the third qualifier from the data sets that are in the LPARA.FGP fingerprint data set (created in ["BASE Command Example 5" on page 122](#)).

```
BASE=LOAD=(DSN=LPARA.FGP, INCLDSN=3LINKLIB)
```

BASE Command Example 7

You can establish the base environment by a combination of VOL and GENERIC parameter specifications.

```
BASE=GENERIC=(DSN=SYS1.*)
BASE=VOL=(SER=VOL001, INCLDSN=SYS2/)
BASE=SAVE=(DD=DD1)
```

BASE Command Example 8

This example specifies an alias name defined in the catalog. The true data set pointed by this entry is *somnode.CMNSSM.LOADLIB.VER810*. All references in the fingerprint data set, the change basket, and the reports refer to the physical data set name, not the alias name.

```
BASE=DSN=somnode . CMNSSM . LOADLIB
BASE=SAVE=(DSN=somnode . FGP . CMNSSM . LOADLIB)
```

DSN Parameter

The DSN parameter provides for the inclusion of any number of explicitly specified, cataloged data sets in the system.

The VOL parameter is not required for cataloged data sets.

The general format of the DSN parameter follows:

```
DSN = ( _, ... )
```

Data sets must be cataloged. However, if uncataloged data sets are to be selected, then use the VOL parameter and specify the complete data set name with the INCLDSN subparameter.

There are no subparameters for the DSN parameter.

DSN Parameter Example 1

The following explicitly-named data sets are selected (and fingerprinted, if provided in the BASE or TARGET commands).

```
BASE=DSN=(CMNSSM . LOADLIB ,
          CMNSSM . SKELS ,
          CMNSSM . PANELS ,
          CMNSSM . MSGS)
BASE=SAVE=DSN=CMNSSM . FGP
```

DSN Parameter Example 2

To select an uncataloged data set, use the VOL parameter.

The INCLDSN subparameter shows the explicit data set names. DASD volume VOL001 is searched for the UNCAT.EXPLICIT.DSN data set. Once found, it is fingerprinted.

```
BASE=VOL=(INCLVOL=VOL001 ,
          INCLDSN=UNCAT . EXPLICIT . DSN)
BASE=SAVE=DSN=CMNSSM . FGP
```

GENERIC Parameter

The GENERIC parameter performs a catalog search to establish the environment. You can apply further inclusion or exclusion to the list if you are fingerprinting a desired pattern of data sets and volumes.

The DSN subparameter is required. The general format of the parameter follows:

```

GENERIC = ( DSN = ( _ , ... ) ,
            EXCLDSN = ( _ , ... ) ,
            EXCLVOL = ( _ , ... ) ,
            INCLVOL = ( _ , ... ) ,
            TYPE = ( _ , ... ) )

```

The following table describes the GENERIC subparameters:

Subparameter	Description
DSN	<p>Specifies the high-level qualifiers and the following nodes to be searched for from the catalog.</p> <p>Rules of high-level qualifiers and partial nodes are similar to ISPF Option 3.4. See "Catalog Search" on page 248 for more information about specifying patterns.</p> <p>You can specify multiple patterns as the arguments for this parameter.</p>
EXCLDSN	<p>Excludes the specified explicit or pattern of data sets from the resolved list.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
INCLDSN	<p>Includes the specified explicit or pattern of data sets.</p> <p>If this subparameter is not specified, all data sets are included in the process.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
EXCLVOL	<p>Rejects the data sets that reside on the specified volumes from the resolved list.</p> <p>NOTE When EXCLVOL is specified without INCLVOL, then INCLVOL=/ (all volumes) is implied.</p> <p>NOTE When EXCLVOL is <i>not</i> specified, INCLVOL=/ (all volumes) is implied.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
INCLVOL	<p>Accepts only those data sets that reside on the volumes specified in the resolved list.</p> <p>The default is "/" (all volumes).</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>

Subparameter	Description
TYPE	<p>Allows the selection of data sets by their DSORG. If not specified, all non-VSAM data sets with a DSORG are selected.</p> <p>You can specify multiple types, separated by commas. The valid values for this subparameter are:</p> <ul style="list-style-type: none"> ■ DA – Direct access data sets. ■ LOAD – Load libraries. ■ PDS – Equivalent to PO (alternate description). ■ PO – Partitioned data sets - PDS and PDSE. ■ PS – Physical sequential data sets. ■ SRC – Source libraries (including JCL, panels, and messages). ■ TEXT – Equivalent to SRC (alternate description). <p>Mutual specifications of SRC and TEXT are redundant. TYPE=LOAD indicates all load libraries. TYPE=(PS,DA,TEXT) indicates all other data set types that can be fingerprinted.</p>

GENERIC Parameter Example 1

The BASE command resolves the three specified patterns. From this list, only the source data sets are fingerprinted to establish the base environment.

```
BASE=GENERIC=(DSN=(SYS1 , SYS2 , SYS3) ,
              TYPE=SRC)
BASE=SAVE=(DSN=SYSTEM.FGP)
```

GENERIC Parameter Example 2

The BASE command resolves the SYS1 data sets, then excludes the data set specified by the EXCLDSN subparameter.

Once the list is determined, the fingerprinting starts. The fingerprint tokens are written to the fingerprint data set as specified in the next command.

```
BASE=GENERIC=(DSN=SYS1 . * ,
              EXCLDSN=(SYS1 . BRODCAST))
BASE=SAVE=(DSN=SYS1 . FGP)
```

GENERIC Parameter Example 3

The BASE command resolves all DEVL data sets, then excludes all data sets with either the LIST or TEMP floating node.

The LIST command produces a list of the data sets, members, and their fingerprint tokens.

```
BASE=GENERIC=(DSN=DEVL ,
              EXCLDSN=( !LIST! , !TEMP! ))
LIST=BASE
```

GENERIC Parameter Example 4

The BASE command resolves the 3 data set patterns from the catalog, then includes only the partitioned data sets. Once fingerprinted, the result is saved in a fingerprint data set for a future process.

```
BASE=GENERIC=(DSN=(CMNSSM.V810,
                  CMNSSM.V730,
                  CMNSSM.V720) ,
              TYPE=PO)
BASE=SAVE=(DSN=CMNSSM.FGP.APR22)
```

GENERIC Parameter Example 5

The following example demonstrates how a combination of GENERIC and VOLUME specifications can define an environment for fingerprinting.

The result of this example is that all SYS1 prefixed data sets are fingerprinted, wherever the catalog finds them, as well as all data sets (cataloged/uncataloged) that exist on volume VOL001 having the prefix of SYS2, and the one explicit data set name that is being specified.

```
BASE=GENERIC=(DSN=SYS1.*)
BASE=VOL=SER=(VOL001, INCLDSN=(SYS2/, EXPLICIT.UNCAT.DSN)
BASE=SAVE=(DD=FGPDD)
```

GENERIC Parameter Example 6

The following complex example shows the use of HLQ, node, and wild card (field positional) specifications.

All SYS1, SYS2A, and SYS2B prefixed data sets, and data sets having an exact match of NYC as the 2nd node, are identified. Additionally, these identified data sets having a last node of OLD, NEW, or BKUP will be excluded. To further limit the data sets to be operated on, those data sets having the above qualifications, on volumes with the 3rd to 5th position of PRD or DEV, will not be operated on.

```
BASE=GENERIC=(DSN=(SYS1, SYS2A, SYS2B) ,
              EXCLVOL=(??PRD?, ??DEV?) ,
              INCLDSN=(2NYC./) ,
              EXCLDSN=(0OLD, 0NEW, 0BKUP))
BASE=SAVE=DD=FGPDD
LIST=BASE
```

LOAD Parameter

The LOAD parameter recalls a previously-generated fingerprint data set (or its subset), using patterns. The parameter has four subparameters.

The general format of the LOAD parameter follows:

```
LOAD = ( DD = ( _ ) | DSN = ( _ ),
        EXCLDSN = ( _ , _ , ... ) ,
        INCLDSN = ( _ , _ , ... ) ,
        TYPE=( _ , _ , ... ))
```

When a fingerprint data set is loaded in memory (using the LOAD parameter of the BASE or TARGET command), all data sets are included in the process, unless they are explicitly excluded.

The MEMBERS parameter of the EXPORT command, for example, does not limit the process to the specified data set. It selects only the members of the specified data set. To restrict the data set selections, use the INCLDSN or EXCLDSN parameters.

The following table describes the LOAD subparameters

Subparameter	Description
DD	<p>Refers to a sequential data set that contains fingerprint tokens and other synchronization data.</p> <p>This information is used throughout all software functions. It is a means to retrieve a previously-generated data set.</p> <p>The retrieval and use of the contents of this data set may be further restricted by using the INCLDSN or EXCLDSN subparameter.</p> <p>See "SAVE Parameter" on page 130 for additional information</p>
DSN	<p>Names a cataloged physical sequential data set that contains fingerprint tokens and other synchronization data.</p> <p>This facility retrieves a previously-generated data set (see "SAVE Parameter" on page 130). The retrieval and use of the contents of this data set may be further restricted by using the INCLDSN or EXCLDSN subparameter.</p>
EXCLDSN	<p>Specifies a pattern to exclude various subsets of the synchronization data from fingerprint data set processing.</p> <p>Data sets that meet the pattern are not loaded into memory.</p> <p>If this subparameter is not specified, all data sets are included in the process.</p> <p>This subparameter can be specified only if INCLDSN is specified.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
INCLDSN	<p>Selects various subsets of the synchronization data from a fingerprint data set through a pattern.</p> <p>If this subparameter is not specified, all data sets are included in the process.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>

Subparameter	Description
TYPE	<p>Allows the selection of data sets by their DSORG. If not specified, all non-VSAM data sets with a DSORG are selected. You can specify multiple types, separated by commas. The valid values for this subparameter follow:</p> <ul style="list-style-type: none"> ■ DA – Direct access data sets. ■ LOAD – Load libraries. ■ PDS – Equivalent to PO (alternate description). ■ PO – Partitioned data sets - PDS and PDSE. ■ PS – Physical sequential data sets. ■ SRC – Source libraries (including JCL, panels, and messages). ■ TEXT – Equivalent to SRC (alternate description). <p>Mutual specifications of SRC and TEXT are redundant. TYPE=LOAD indicates all load libraries. TYPE=(PS,DA,TEXT) indicates all other data set types that can be fingerprinted.</p>

LOAD Parameter Example 1

All data sets are selected from the RECOVER.SITE.FGP fingerprint data set (in this case, the same would be true if the INCLDSN subparameter were omitted).

```
BASE=LOAD=(DSN=RECOVER.SITE.FGP,INCLDSN=/)
```

LOAD Parameter Example 2

The fingerprint data sets referred by the DDs are recalled and appended in memory.

```
BASE=LOAD=DD=FGPDD1
TARG=LOAD=DD=FGPDD2
```

LOAD Parameter Example 3

The fingerprint tokens of all data sets starting with SYS1 are loaded in memory. In this example, the fingerprint data set is referred to by the FGPDD in the JCL.

```
BASE=LOAD=(DD=FGPDD,INCLDSN=SYS1/)
```

LOAD Parameter Example 4

All data set fingerprints, except SYS1.PARMLIB, are loaded in memory.

```
BASE=LOAD=(DD=FGPDD,INCLDSN=SYS/,EXCLDSN=SYS1.PARMLIB)
```

LOAD Parameter Example 5

A fingerprint data set represents all data sets in a software package. Use the TYPE subparameter to load a subset of the data set tokens into memory.

In the following example, only source (source and text types are equivalent) data sets such as macros, panels, or JCL are included.


```
BASE=LOAD=(DD=DD1,TYPE=TEXT)
```

LOAD Parameter Example 6

In this example, all LOAD library tokens are loaded into memory.

```
BASE=LOAD=(DD=DD1,TYPE=LOAD)
```

LOAD Parameter Example 7

In this example, all physical sequential and direct access file tokens are loaded into memory.

```
BASE=LOAD=(DD=DD1,TYPE=(PS,DA))
```

RESET Parameter

The RESET parameter clears the base or target environment.

If BASE or TARGET commands are stacked, the data extracted in a previous command is appended by subsequent commands.

If it is necessary to begin with a new environment, then use the RESET subparameter. This provides added flexibility to the command stacking capabilities.

When RESET is encountered, the particular environment is refreshed and then re-specified by subsequent BASE or TARGET commands.

The general format of the RESET parameter follows:

```
RESET
```

There are no subparameters.

RESET Parameter Example

The first grouping of the stacked commands establishes two environments.

The base group is generated by the BASE command, which loads source data sets (third node being SRC) from the PRD.SITE.FGP fingerprint data set.

The target group is generated with a TARGET command loading source data sets (third node being SRC) from the DEV.SITE.FGP fingerprint data set.

A corresponding data set-by-data set comparison is performed by investigating each data set fingerprint and, when required, each member's fingerprint token.

The second section of the stacked commands resets the base and target groups when the RESET parameters are encountered.

Both base and target environments are re-established by recalling the previously-generated fingerprint data sets. The software then proceeds to do another, separate composite comparison between the two new environments, this time selecting data sets that have LOAD as the last data set name node.

```
BASE=LOAD=(DSN=PRD.SITE.FGP,INCLDSN=*.*.SRC)
TARG=LOAD=(DSN=DEV.SITE.FGP,INCLDSN=*.*.SRC)
COMPARE=DSNBYSN
```

```
BASE=RESET
TARG=RESET
```

```
BASE=LOAD=(DSN=PRD.SITE.FGP,INCLDSN=0LOAD)
TARG=LOAD=(DSN=DEV.SITE.FGP,INCLDSN=0LOAD)
COMPARE=DSNBYDSN
```

This approach allows for the creation of separate reports for desired groups of data sets.

SAVE Parameter

The SAVE parameter allows you to save the generated fingerprint data set or a subset of it (using patterning). The fingerprint data set (FGP) can be referred by a DSN or DD subparameter. The SAVE parameter has four subparameters.

The general format of the SAVE parameter follows:

```
SAVE = ( DD = ( _ ) | DSN = ( _ ),
        EXCLDSN = ( _, ... ),
        INCLDSN = ( _, ... ) )
```

This table describes the SAVE subparameters.

Subparameter	Description
DD	Refers to a physical sequential data set that is to contain fingerprint tokens and other synchronization data. This information is used throughout the software functions. The facility retains that data by its incorporation into a data set. Restrict the contents of this data set by using the INCLDSN or EXCLDSN subparameter.
DSN	Names a pre-allocated physical sequential data set that is to contain fingerprint tokens and other synchronization data. This information is used throughout the software functions. The facility retains that data by its incorporation into a data set. The DCB attributes of the fingerprint data set are: DCB=(DSORG=PS,RECFM=FB,LRECL=512,BLKSIZE=4096) Restrict the contents of this data set by using the INCLDSN or EXCLDSN subparameters.
EXCLDSN	Allows you to specify a pattern that excludes various subsets of the fingerprinted data. Only those data sets that meet the pattern specified in this subparameter are excluded from the saved fingerprint data set. Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.
INCLDSN	Allows you to specify a pattern that includes various subsets of the fingerprinted data. If this subparameter is not specified, all data sets are included. Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.

SAVE Parameter Example 1

The base environment is established and fingerprinted by the first command. It contains all source and load libraries found on volumes VOL001 and VOL002.

Concurrently, the second, third, and fourth commands each create individual fingerprint data sets from the environment identified by the first BASE command.

The second command generates a fingerprint data set for data sets that have LINKLIB as the last node.

The third base command also generates a separate fingerprint data set for three-node data sets that have SRC as the last node.

The fourth command fingerprint data set contains all data sets selected by the first command (the default).

```
BASE=VOL=(INCLVOL=(VOL001,VOL002),
          TYPE=(SRC,LOAD))
BASE=SAVE=(DSN=LPARA.LINKLIB.FGP,
           INCLDSN=0LINKLIB)
BASE=SAVE=(DSN=LPARA.SRC.FGP,
           INCLDSN=*.*.SRC)
BASE=SAVE=(DSN=LPARA.FGP)
```

SAVE Parameter Example 2

All non-VSAM data sets on volume IPLAA2 are fingerprinted. The result of the process is stored on the two dual-copy fingerprint data sets referred by the DDs.

One of these fingerprint data sets could, for example, be sent to an alternate site to verify the synchronization between the two volumes.

```
BASE=VOL=INCLVOL=IPLAA2
BASE=SAVE=DD=FGPDD1
BASE=SAVE=DD=FGPDD2
```

VOL Parameter

VOL defines the DASD volumes that are to be searched in the process of generating a fingerprint data set.

Use VOL when volume processing is required instead of the catalog processing. It is not required when explicit cataloged data sets are referenced.

To see all of the data sets in the generated fingerprint, specify TRACE=Y as the first command in the command list (see ["VOL Parameter Example 4" on page 133](#)).

The general format of the VOL parameter follows :

```
VOL = (EXCLDSN = ( _,... ),
       EXCLVOL = ( _,... ),
       INCLDSN = ( _,... ),
       INCLVOL = ( _,... ),
       TYPE = ( _,... ) )
```

The following table describes the VOL subparameters.

Subparameter	Description
EXCLDSN	<p>Allows you to specify a pattern that excludes various data sets when using the VOL subparameter.</p> <p>The default is that all partitioned data sets (PO), physical sequential data sets (PS), and direct access data sets (DA) are selected (if not limited by the TYPE subparameter).</p> <p>You can specify multiple patterns, separated by commas.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
INCLDSN	<p>Allows you to specify a pattern that includes various data sets when using the VOL subparameter.</p> <p>The default is that all partitioned data sets (PO), physical sequential data sets (PS), and direct access data sets (DA) are selected (if not limited by the TYPE subparameter).</p> <p>You can specify multiple patterns, separated by commas.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
EXCLVOL	<p>Allows you to specify volume serial numbers to be excluded from the already-selected INCLVOL group.</p> <p>You specify an explicit six-character volume serial number, or a generic volume serial number pattern ending with an asterisk.</p> <p>You can specify multiple volume serial numbers, separated by commas.</p> <p>Patterns and explicit volume serial numbers can be present in the same definition.</p>
INCLVOL	<p>Allows you to specify volume serial numbers to be included for processing.</p> <p>The INCLVOL subparameter is either an explicit six-character volume serial number, or a generic volume serial number pattern ending with an asterisk.</p> <p>You can specify multiple volume serial numbers, separated by commas.</p> <p>Patterns and explicit volume serial numbers can be present in the same definition.</p> <p>INCLVOL is mutually exclusive with the GENERIC parameter, which is used for catalog processing versus volume processing.</p> <p>Use of INCLVOL=* is not recommended for data centers with over 200 DASD volumes; instead, use the GENERIC parameter to perform a catalog search (if data sets are cataloged).</p>

Subparameter	Description
TYPE	<p>Allows the selection of data sets by their DSORG. If not specified, all non-VSAM data sets with a DSORG are selected.</p> <p>You can specify multiple types, separated by commas. The valid values for this subparameter follow:</p> <ul style="list-style-type: none"> ■ DA – Direct access data sets. ■ LOAD – Load libraries. ■ PDS – Equivalent to PO (alternate description). ■ PO – Partitioned data sets - PDS and PDSE. ■ PS – Physical sequential data sets. ■ SRC – Source libraries (including JCL, panels, and messages). ■ TEXT – Equivalent to SRC (alternate description). <p>Mutual specifications of SRC and TEXT are redundant. TYPE=LOAD indicates all load libraries. TYPE=(PS,DA,TEXT) indicates all other data set types that can be fingerprinted.</p>

VOL Parameter Example 1

All volume serial numbers beginning with TEST, excluding TEST09, are selected.

```
BASE=VOL=(INCLVOL=TEST*,EXCLVOL=TEST09)
```

VOL Parameter Example 2

All volumes in the entire system are selected.

This command requires a very large region size. If there are more than 200 volumes in the system, an insufficient storage condition may arise. In such cases, specify two or more pools by specifying their volume patterns.

To fingerprint a large group of cataloged data sets, use the GENERIC parameter.

```
BASE=VOL=INCLVOL=*
```

VOL Parameter Example 3

All partitioned data sets on the specified volume (VOL001) are selected and fingerprinted.

```
BASE=VOL=(INCLVOL=VOL001,TYPE=PO)
```

VOL Parameter Example 4

All data sets (source and load) on the specified volume (VOL001) are selected.

TRACE=Y provides a list of all data sets selected and fingerprinted. This report shows the contents of the saved fingerprint data set.

```
TRACE=Y
BASE=VOL=(INCLVOL=VOL001,TYPE=(LOAD,SRC))
BASE=SAVE=(DSN=VOL001.FGP)
```

VOL Parameter Example 5

The following is another example of the VOL parameter.

```
BASE=VOL=(INCLVOL=(APPL*),
          EXCLVOL=(APPL07,APPL08),
          INCLDSN=APPLABC.TEST/,
          EXCLDSN=(APPLABC.TEST.LOAD4,
                  APPLABC.TEST.LOAD5))
BASE=SAVE=(DSN=VOL001.FGP)
```

CLOCK Command

The CLOCK command allows for a time controlled volume-level token fingerprint data set process.

If a large pool of volumes needs to be fingerprinted for redundancy detection, use the CLOCK command to manage the task in several time segments.

The only parameter allowed for this command is the clock time_in_minutes.

The general format of the CLOCK command is:

```
CLOCK = (time_in_minutes)
```

The absence of this command indicates that there is no ChangeMan SSM-imposed time limit for the task. The CPU time specified for the job still determines the ultimate CPU resource allowed for the task.

The following table describes the CLOCK command parameters.

Values	Description
0	Turns clocking off.
Non-zero	Specifies the number of minutes the fingerprint process will run. When the time expires, ChangeMan SSM continues to fingerprint the data set on which it is working, then saves the location where it should restart the halted process. See " DASDRED Command Example 2 " on page 150 for additional details.

CLOCK Command Example 1

The first command defines the clock time limit for fingerprinting this large pool of volumes. As each selected data set is fingerprinted, the fingerprint tokens are written to the external token fingerprint data set referenced by the DD. When the time expires, it closes the file and terminates. This allows restart processing.

```
CLOCK=120
DASDRED=(INCLVOL=TST*,DD=OUTDD)
```

CLOCK Command Example 2

The first command sets the time limit for fingerprinting the pool of volumes. All physical sequential data sets are fingerprinted, and as each data set is fingerprinted, a record is written to the external token fingerprinted data set referenced by the DD. When the time expires, it closes the file and terminates.

This function is restartable.

```
CLOCK=240
DASDRED=(INCLVOL=TST*,TYPE=PS,DD=OUTDD)
```

COMPARE Command

The COMPARE command defines the actual mode of the high volume software environment synchronization verification.

At any one time, the command uses one of its four parameters.

The LKED for load modules is not included in the process of generating the directory token. However, if you specify the EXTENDED=YES command, it includes the directory token and LKED date as part of the comparison report.

The LKED date is stored in the fingerprint data set; however, it is not part of the directory token or fingerprint token.

The general format of the COMPARE command follows:

```
COMPARE = ( COMPOSITE = ( _ ) | DSNBYDSN = ( _ ) |
            DSNTODSN = ( _ ) | FILTER = ( _,_ ) |
            MEMTOMEM = ( _,_ ) )
```

The following table describes the COMPARE command parameters.

Parameter	Description
COMPOSITE	Verifies the synchronization of two groups of data sets, regardless of data set name associations or PDS member name associations. This mode uses the fingerprint tokens as the sole consideration for synchronization verification. The presence of two identical fingerprint tokens in the base and target determines a sync condition; the absence of identical fingerprint tokens indicates an out-of-sync condition. For details, see "COMPOSITE Parameter" on page 137 .
DSNBYDSN	Verifies the synchronization of two groups of data sets, "one for one", based on their names. For example, data set Alpha in the base is compared to data set Alpha in the target, and so forth. Once a matching data set name is established, the MEMTOMEM mode for that data set is in effect. See "EXTRACT Command" on page 154 for additional details.
DSNTODSN	Verifies synchronization at the data set level by investigating data set fingerprints, without going into a detailed member-level verification. For example, when comparing two volumes, this method can identify all in-sync or out-of-sync data sets. Subsequently, a detailed member-level verification can be requested for out-of-sync data sets in the two groups. See "DSNTODSN Parameter" on page 141 for additional details.

Parameter	Description
FILTER	Provides synchronization verification of subsets of two partitioned data sets, determined by various member level filtering subparameters. After the selected names are matched, the members' fingerprint tokens are evaluated to determine equality or inequality for the matched members. See " FILTER Parameter " on page 142 for additional detail.
MEMTOMEM	Specifies the member-to-member comparison mode between two designated data sets. For example, member Alpha in a base data set is verified against member Alpha in a target data set. If the fingerprint tokens of these two members are identical, they are considered to be name-matched and content-identical. If the member fingerprint tokens are different, then the contents of the members are different. Through this process, the match, mismatch, sync, and out-of-sync members are identified. For details see " MEMTOMEM Parameter " on page 144.

The following examples demonstrate the proper use of the COMPARE command. Refer to the *ChangeMan SSM Getting Started Guide* for additional examples. The ChangeMan SSM SAMPLES library also contains examples of the COMPARE command.

COMPARE Command Example 1

The two fingerprint data sets represent the contents of a volume (at two different times).

The following commands establish the volume representations by loading their fingerprints. The fingerprint data set tokens are evaluated to identify the changed data sets between the two times for the volume.

```
BASE=LOAD=DD=FGPDD1
TARG=LOAD=DD=FGPDD2
SETPRINT=(OUTSYNC=Y, SYNC=N, MISMATCH=Y)
COMPARE=DSNTODSN=TOKEN
```

COMPARE Command Example 2

The two data sets in the base (active data set) and target (previously-generated fingerprint data set) environment are compared by investigating member fingerprint tokens.

```
BASE=DSN=SYS1.LINKLIB
TARG=LOAD=(DD=FGPDD2, INCLDSN=SYS1.LINKLIB)
SETPRINT=(OUTSYNC=Y, SYNC=N, MISMATCH=Y)
COMPARE=MEMTOMEM
```

COMPARE Command Example 3

The two PDSE data sets are internally fingerprinted and compared to report the out-of-sync members.


```

BASE=DSN=PDSE . SOURCE
TARG=DSN=PDSE . SOURCE
SETPRINT=(OUTSYNC=Y , SYNC=N , MISMATCH=Y)
COMPARE=MEMTOMEM

```

COMPOSITE Parameter

The COMPOSITE parameter verifies the synchronization of two groups of data sets, regardless of data set name associations or PDS member name associations.

This verification process uses the fingerprint token as the sole consideration for synchronization. Identical fingerprint tokens in the base and target environments determines a sync condition; the absence of identical tokens indicates an out-of-sync condition.

This method of verification is used when data set name association is not feasible. Data sets and member fingerprint tokens are the criteria for synchronization verification.

The format of the COMPOSITE parameter follows:

```

COMPOSITE = ( EXCLMEM op ( _,... ) ,
              INCLMEM op ( _,... ) )

```

where *op* represents a Boolean operator (=, <, or >).

The following table describes the COMPOSITE subparameters.

Subparameter	Description
EXCLMEM = (_,...) EXCLMEM < (_,...) EXCLMEM > (_,...)	Provides the facility to exclude explicit or pattern members from the report. If this subparameter is not specified, no members are excluded. You can use wildcards in this subparameter (see "Member Name Wildcards" on page 249 for details).
INCLMEM = (_,...) INCLMEM < (_,...) INCLMEM > (_,...)	Provides the facility to include explicit or pattern members in the report. If this subparameter is not specified, all members are included. You can use wildcards in this subparameter (see "Member Name Wildcards" on page 249 for details).

The following examples demonstrate the proper use of the COMPOSITE parameter. Refer to the *ChangeMan SSM Getting Started Guide* for additional examples.

COMPOSITE Parameter Example 1

The two fingerprint data sets each represent a group of data sets at a particular version level.

Data set naming conventions do not allow a corresponding data set-by-data set (DSNBYDSN mode) compare between the two groups. By investigating member fingerprint tokens, the COMPOSITE mode of compare can identify the out-of-sync members between the two groups.

```

BASE=LOAD=(DD=FGP630 , INCLDSN=SER.V810.CMNSSM/)
TARG=LOAD=(DD=FGP620 , INCLDSN=SER.V730.CMNSSM/)
SETPRINT=(OUTSYNC=Y , SYNC=N)
COMPARE=COMPOSITE

```

COMPOSITE Parameter Example 2

Two versions of a software package are represented by their fingerprint data sets, and different types of reports are requested.

```

BASE=LOAD=DD=FGP630
TARG=LOAD=DD=FGP620
COMPARE=COMPOSITE=INCLMEM=( !CPX! , !HPS! )
COMPARE=COMPOSITE=( INCLMEM>A , INCLMEM<C )
COMPARE=COMPOSITE=( INCLMEM=B / )
COMPARE=COMPOSITE=( EXCLMEM=X??Y )
COMPARE=COMPOSITE=( INCLMEM=X' C0?? ' )

```

DSNBYDSN Parameter

The DSNBYDSN parameter compares two groups of data sets, one by one, in name order.

For example, data set Alpha in the base is compared to data set Alpha in the target. Once you establish a matching data set name, if the data set tokens were not identical, the MEMTOMEM mode for that data set begins.

Using this mode, a SYSRES volume can be compared with an alternate SYSRES volume, using a minimum set of commands.

```

DSNBYDSN = ( DETECT | COMPAREX | SUPERC ,
             LONG | SHORT ,
             DISPLAYTOKEN = ( Y / N ) ,
             MODEL=model_name )

```

The following table describes the DSNBYDSN subparameters:

Subparameter	Description
DETECT	Passes only those members with different contents (determined by their fingerprint tokens) to the ChangeMan SSM internal DATACOMPARE facility. Refer to the jobs DATA*CMP in the SAMPLES data set for examples of the use of this feature.
Comparex	Passes only those members with different contents (determined by their fingerprint tokens) to Comparex or SuperC. Refer to the jobs CPXFAST in the SAMPLES data set for examples of the use of this feature.
SUPERC	Passes only those members with different contents (determined by their fingerprint tokens) to SuperC. Refer to the jobs SUPERC* in the SAMPLES data set for examples of the use of this feature.

Subparameter	Description
LONG	<p>Default. Requests long style reports.</p> <p>The COMPARE=DSNBYDSN=LONG command honors all of the SETPRINT flags.</p> <p>The process is as follows:</p> <ul style="list-style-type: none"> ■ Pass 1 : Base data set mismatch. Superseded if DSNMISMATCH=N. ■ Pass 2: Target data set mismatch. Superseded if DSNMATCH=N. ■ Pass 3: For matched data sets, MEMtoMEM is invoked. <p>NOTE For a summary report only, specify: SETPRINT=(DSNMATCH=N, DSNMISMATCH=N).</p>
SHORT	<p>Requests short style reports.</p> <p>The process is as follows:</p> <ul style="list-style-type: none"> ■ Pass 1: Base data set mismatches. ■ Pass 2: Target data set mismatches. ■ Pass 3: In-sync data sets. ■ Pass 4: MEMtoMEM reporting for out-of-sync data sets. <p>This report cannot be tailored. The SHORT command ignores the SETPRINT command and sets the SETPRINT parameters to MISMATCH=Y, OUTSYNC=Y, and SYNC=N.</p>
DISPLAYTOKEN	<p>Specifies that the fingerprint tokens are to be displayed in the report. If Sync, OutSync and MisMatch are to be displayed instead.</p>
MODEL	<p>Use a transformation model in comparing two environments when data set names do not match.</p> <p>A MODEL command establishes the associations between data set names; refer to "MODEL Command" on page 183 for additional details.</p>

DSNBYDSN Parameter Example 1

The following example compares two entire DASD volumes. Each volume is represented by its fingerprint data set.

An exception report has been requested.

All data sets that are matched by their names are included in the process.

A MEMTOMEM mode applies to each pair of data sets if their data set tokens are not identical. The members that do not match in each pair are reported. Also, those that match, but have different contents, are reported.

```

BASE=LOAD=DSN=IPLAA2 .FGP
TARG=LOAD=DSN=IPLBB2 .FGP
SETPRINT=(EXCEPTIONSONLY=Y)
COMPARE=DSNBYDSN=SHORT

```

DSNBYDSN Parameter Example 2

This MODEL command performs the transformation of the second node LPAR to LPARA. Only data sets whose second node is LPAR will have this transformation applied.

```
EXTENDED=YES
MODEL=(NAME=MODEL01, TRANSFORM=(2LPAR=>LPARA))
*
BASE=GENERIC=(DSN=USER17.LPAR)
TARG=GENERIC=(DSN=USER17.LPARA)
SETPRINT=(SYNC=N, OUTSYNC=Y, MISMATCH=Y)
COMPARE=DSNBYDSN=(MODEL=MODEL01, SHORT)
```

DSNBYDSN Parameter Example 3

The SHORT command process makes four passes, as follows:

Pass 1: Base D/S mismatch.

Pass 2: Targ D/S mismatch.

Pass 3: One line per in-sync data set.

Pass 4: Member processing; MEMtoMEM is invoked for out-of-sync data sets.

SHORT ignores the SETPRINT command and sets the SETPRINT parameters to MISMATCH=Y, OUTSYNC=Y, and SYNC=N.

```
COMPARE=DSNBYDSN=(SHORT, DISPLAY TOKENS = NO)
```

Refer to the *ChangeMan SSM Getting Started Guide* for examples of the DSNBYDSN parameters.

DSNBYDSN Parameter Example 4

In this example, two groups of load libraries are first fingerprinted, after which the members that have different tokens are passed to Comparex.

Refer to the sample jobs (CPXFAST*) in the SAMPLES data set for JCL requirements.

```
BASE=GENERIC=(DSN=SYS7.CICS41.*, TYPE=LOAD)
TARG=GENERIC=(DSN=SYS7.CICS31.*, TYPE=LOAD)
MODEL=(NAME=MODEL01, TRANSFORM=2CICS41=>CICS31)
COMPARE=DSNBYDSN=(MODEL=MODEL01, COMPAREX)
```

DSNBYDSN Parameter Example 5

In this example, two groups of load libraries are represented by their fingerprint data sets. Only those members that have different tokens are passed to SuperC for a line-by-line comparison.

Refer to the sample jobs (SUPERC*) in the SAMPLES data set for JCL requirements.

```
BASE=LOAD=(DD=DD1, TYPE=TEXT)
TARG=LOAD=(DD=DD2, TYPE=TEXT)
MODEL=(NAME=MODEL01, TRANSFORM=2CICS41=>CICS31)
COMPARE=DSNBYDSN=(MODEL=MODEL01, SUPERC)
```

DSNTODSN Parameter

The DSNTODSN parameter verifies synchronization at the data set level by investigating data set fingerprint tokens, without consideration of the members. The data set fingerprints determine equality for the matched data sets.

For example, when comparing two DASD volumes, this method can identify all in-sync or out-of-sync data sets. A detailed member-level verification can then be requested for out-of-sync data sets in the two groups.

The parameter has three subparameters.

The general format of the DSNTODSN parameter follows:

```
DSNTODSN = ( DSNAME | SEQUENCE | TOKEN,
             MODEL=model_name,
             DISPLAYTOKEN=Y/N)
```

The following table describes the DSNTODSN subparameters.

Subparameter	Description
DSNAME	Specifies that the data set name is the only criterion for synchronization verification. Missing data sets in either the base or target group are identified.
SEQUENCE	Specifies that synchronization verification is by the sequence presented. Contents are considered in the sequence of the data sets in the base and target groups. The <i>n</i> th data set contents (fingerprint) of the base group is verified against the <i>n</i> th data set of the target group. This approach is helpful when naming conventions do not allow any other comparison method.
TOKEN	Specifies that the data set fingerprint token is the only criterion for synchronization verification.
DISPLAYTOKEN=	Specifies that the fingerprint tokens are to be displayed in the report (or if Sync, OutSync and MisMatch are to be displayed instead). If you specify DISPLAYTOKEN=Y, fingerprint tokens are displayed in the report. If you specify DISPLAYTOKEN=N, Sync, OutSync and MisMatch are displayed in the report.
MODEL	Establishes associations between data set names. Use a transformation model when comparing two environments where data set names do not match. Refer to "MODEL Command" on page 183 for additional detail.

DSNTODSN Parameter Example

The first command loads, selects, and fingerprints all data sets whose names start with SYS1.

On the second DASD volume, all data sets starting with the nodes SYS1.NEW are loaded and selected.

A request for all out-of-sync data sets is requested.

Because the naming convention is inappropriate for a DSNBYDSN verification, DSNTODSN is specified. In this example, the data set tokens are the only criteria for verification.

```
BASE=VOL=(INCLVOL=IPLAA2,INCLDSN=SYS1. /)
TARG=VOL=(INCLVOL=IPLBB2,INCLDSN=SYS1.NEW. /)
SETPRINT=(OUTSYNC=Y,SYNC=N)
COMPARE=DSNTODSN=TOKEN
```

FILTER Parameter

The FILTER parameter provides synchronization verification of subsets of two partitioned data sets, as determined by various filtering sub-parameters.

After the selected filtered member names are matched, the members' fingerprint tokens are evaluated to determine the in-sync/out-of-sync conditions for the matched members. A return code of zero (RC=0) indicates that all selected members are identical in their contents. A return code of four (RC=4) indicates that two or more selected subsets are out-of-sync.

If you use FILTER to verify the synchrony of two software environments that have multiple partitioned data sets, each partitioned data set must be named explicitly using the BASEDSN and TARGDSN subparameters (otherwise, these two subparameters are optional).

Either INCLMEM or EXCLMEM (or both) is required.

The general format of the FILTER parameter follows:

```
FILTER = ( BASEDSN = ( _ ),
           TARGDSN= ( _ ),
           INCLMEM op ( _ ,... ),
           EXCLMEM op ( _ ,... ),
           DISPLAYTOKEN = Y / N ,
           DETECT | COMPAREX | SUPERC )
```

where *op* represents a Boolean operator (=, <, or >).

The following table describes the FILTER subparameters.

Subparameter	Description
DETECT	Passes only the members that have different contents (determined by their fingerprint tokens) to the ChangeMan SSM internal DATACOMPARE facility. Refer to the jobs DATA*CMP in the SAMPLES data set for examples of the use of this feature.
COMPAREX	Passes only the members that have different contents (determined by their fingerprint tokens) to Comparex. Refer to the jobs CPXFAST* in the SAMPLES data set for examples of the use of this feature.
SUPERC	Passes only the members that have different contents (determined by their fingerprint tokens) to SuperC. Refer to the jobs SUPERC* in the SAMPLES data set for examples of the use of this feature.

Subparameter	Description
DISPLAYTOKEN	Specifies that the fingerprint tokens are to be displayed in the report (or if Sync, OutSync and MisMatch are to be displayed instead). If you specify DISPLAYTOKEN=Y, fingerprint tokens display in the report. If you specify DISPLAYTOKEN=N, Sync, OutSync and MisMatch display in the report.
BASEDSN	Names the base environment partitioned data set that is to be member-name to member-name compared with the data set named in the TARGDSN subparameter.
EXCLMEM = (_,...) EXCLMEM < (_,...) EXCLMEM > (_,...)	Excludes explicit or pattern members from the process and the report.
INCLMEM = (_,...) INCLMEM < (_,...) INCLMEM > (_,...)	Includes explicit or pattern members in the process and the report.
TARGDSN	Names the target partitioned data set that is to be member-name to member-name compared with the data set named in the BASEDSN subparameter.

FILTER Parameter Example 1

Two versions of a software package are represented by their fingerprint data sets. A subset of members is to be evaluated to determine a change or no change condition. A return code of zero (RC=0) indicates that the selected members are unchanged.

```
BASE=LOAD=DD=APPLD.JAN4.FGP
TARG=LOAD=DD=APPLD.JAN1.FGP
COMPARE=FILTER=INCLMEM=(A,B,C)
```

FILTER Parameter Example 2

Two versions of a software package are represented by their fingerprint data sets. Different types of reports are requested by a variety of filtering specifications of the FILTER parameter.

```
BASE=LOAD=DD=CMNSSM.FGP730
TARG=LOAD=DD=CMNSSM.FGP810
COMPARE=FILTER=INCLMEM=(!CPX!,!HPS!)
COMPARE=FILTER=(INCLMEM>A,INCLMEM<C)
COMPARE=FILTER=(INCLMEM=B/)
COMPARE=FILTER=(EXCLMEM=X??Y)
COMPARE=FILTER=(INCLMEM=X'C0??')
```

FILTER Parameter Example 3

Two versions of a library residing on two LPARs are compared. These two libraries are each represented by a fingerprint data set. The COMPARE commands detect the match, mismatch, sync, and out-of-sync conditions between the two libraries.

```
BASE=LOAD=DSN=LPARA.SYS1.PARMLIB.FGP
TARG=LOAD=DSN=LPARB.SYS1.PARMLIB.FGP
COMPARE=FILTER=(BASE=SYS1.LINKLIB,
                TARG=SYS1.LINKLIB,
                EXCLMEM=HPS/)
COMPARE=FILTER=(BASE=SYS1.PARMLIB,
                TARG=SYS1.PARMLIB,
                INCLMEM=APF/)
```

FILTER Parameter Example 4

In this example, two individual load libraries are fingerprinted. Those members that have different tokens and that match the member patterns are passed to Comparex.

Refer to the sample jobs (CPXFAST*) in the SAMPLES data set for JCL requirements.

```
SETPRINT=(SYNC=N,MISMATCH=N)
BASE=DSN=SYS7.CICS41.LOADLIB
TARG=DSN=SYS7.CICS31.LOADLIB
COMPARE=FILTER=(INCLMEM=(A,B,C*),COMPAREX)
```

FILTER Parameter Example 5

In this example, two individual panel libraries are fingerprinted. Only the differing members that match the specified pattern are passed to SuperC for a line-by-line comparison.

Refer to the sample jobs (SUPER*) in the SAMPLES data set for JCL requirements.

```
BASE=DSN=SYS7.CICS41.PANELS
TARG=DSN=SYS7.CICS31.PANELS
COMPARE=FILTER=(INCLMEM=ABC*,SUPER)
```

MEMTOMEM Parameter

The MEMTOMEM parameter provides synchronization verification by member-name to member-name comparison in two partitioned data sets. The member fingerprint tokens are subsequently used to determine sync/out-of-sync conditions for the matched members.

The general format of the MEMTOMEM parameter follows:

```
MEMTOMEM = ( BASEDSN = ( _ ),
             TARGDSN = ( _ ),
             DISPLAYTOKEN= ( Y | N ),
             DETECT | COMPAREX | SUPER)
```

If the base and target each have fingerprint tokens of a single data set, BASEDSN or TARGDSN are not required. Otherwise, the command must explicitly specify the desired data sets.

The following table describes the MEMTOMEM subparameters.

Subparameter	Description
BASEDSN	Names the partitioned data set in the base environment that is to be member-name to member-name compared to the data set named in the TARGDSN.
TARGDSN	Names the partitioned data set in the target environment that is to be member-name-to-member-name compared to the data set named in the BASEDSN.
DISPLAYTOKEN	Specifies if the fingerprint tokens or Sync, OutSync, and MisMatch are to be displayed in the report.
COMPAREX	Passes only the members that have different contents (determined by their fingerprint tokens) to Comparex. Refer to the jobs CPXFAST* in the SAMPLES data set for examples of the use of this feature.
SUPERC	Passes only the members that have different contents (determined by their fingerprint tokens) to IBM SuperC. Refer to the jobs SUPERC* in the SAMPLES data set for examples of the use of this feature.
DETECT	Passes only the members that have different contents (determined by their fingerprint tokens) to the ChangeMan SSM internal DATACOMPARE facility. Refer to the jobs DATA*CMP in the SAMPLES data set for examples of the use of this feature.

The following examples demonstrate the use of the MEMTOMEM parameter. Refer to the *ChangeMan SSM Getting Started Guide* for additional examples.

MEMTOMEM Parameter Example 1

Two versions of a library residing on two different LPARs are loaded and compared. These two libraries are each represented by a fingerprint data set. The COMPARE command detects the match, mismatch, sync, and out-of-sync conditions between the two libraries.

```
BASE=LOAD=DSN=LPARA.SYS1.PARMLIB.FGP
TARG=LOAD=DSN=LPARB.SYS1.PARMLIB.FGP
COMPARE=MEMTOMEM
```

MEMTOMEM Parameter Example 2

Two versions of libraries residing on two different LPARs are loaded and compared. These libraries are represented by their fingerprint data sets.

The MEMTOMEM subparameters designate the two libraries to be compared. Only the out-of-sync modules are requested to be identified (default of the SETPRINT command).

```
BASE=LOAD=DSN=SYS51P.DEC31.FGP
TARG=LOAD=DSN=SYS51P.MAR31.FGP
SETPRINT=SYNC=NO
COMPARE=MEMTOMEM=(BASEDSN=SYS1.LINKLIB,
                   TARGDSN=SYS1.LINKLIB)
COMPARE=MEMTOMEM=(BASEDSN=SYS1.PARMLIB,
                   TARGDSN=SYS1.PARMLIB)
```

MEMTOMEM Parameter Example 3

Two partitioned data sets are compared. Only the members that have different tokens are passed to SuperC for a byte-level comparison. SuperC default settings are used in the comparison.

```
SETPRINT=EXCEPTIONSONLY=YES
EXTENDED=YES
*
BASE=DSN=somnode.TEST.PDS1
TARG=DSN=somnode.TEST.PDS2
*
COMPARE=MEMTOMEM=SUPERC
```

MEMTOMEM Parameter Example 4

Two partitioned data sets are compared. Only the load modules that have different tokens are passed to Comparex for a byte-level comparison.

```
SETPRINT=EXCEPTIONSONLY=YES
EXTENDED=YES
*
BASE=DSN=somnode.TEST.LOADLIB1
TARG=DSN=somnode.TEST.LOADLIB2
*
COMPARE=MEMTOMEM=COMPAREX
*
//*--- SYSIN for COMPAREX
//SYSIN DD *
DATA=CSECT,MODE=SYSTEMS,BUFF=1024,FORMAT=06
```

MEMTOMEM Parameter Example 5

This example uses ChangeMan SSM to fingerprint two groups of data sets, then passes on the differences to Comparex.

A MODEL command associates the data sets.

To invoke Comparex to compare two source data sets, the SYSIN data of TEXT= must be specified.

```
SETPRINT=EXCEPTIONSONLY=YES
EXTENDED=YES
*
MODEL=(NAME=MODEL01,TRANSFORM=2TESTDMS==>TESTHSM)
*
BASE=GENERIC=(somnode.TESTDMS.*,TYPE=TEXT)
TARG=GENERIC=(somnode.TESTHSM.*,TYPE=TEXT)
*
COMPARE=DSNBYSN=(COMPAREX,MODEL=MODEL01)
*
//*--- SYSIN for COMPAREX
//SYSIN DD *
TEXT=.
//*
```

DETECT Subparameter

The DETECT subparameter provides byte-level comparison of out-of sync members. DETECT is available to MEMTOMEM, DSNBYDSN, and FILTER style reports.

This subparameter has two optional subparameters.

The general format of the DETECT parameter follows:

```
DETECT = ( SBS=Y | N,
           COMPLETE = Y | N,
           ADD, UPD, DEL, EQUAL | ALL | NONE )
```

If the base and target each have fingerprint tokens of a single data set, BASEDSN or TARGDSN are not required. Otherwise, the command must explicitly specify the desired data sets.

The following table describes the DETECT subparameters.

Parameter	Description
SBS	Specifies report format. The target records display on the right, and the base records display on the left. SBS=Y (or SBS) specifies side-by-side reporting. SBS=N (or no specification) indicates the default horizontal reporting.
COMPLETE	Specifies amount to display. COMPLETE=N (default) specifies that only the part of a record that fits on a single line displays. COMPLETE=Y specifies that as many lines as necessary print to display the entire record. During CSECT processing, the entire CSECT prints.
ADD	Displays added records (includes moved records).
ALL	Displays all records in the files being compared. This is mutually exclusive with ADD, DELETE, EQUAL, UPDATE, and NONE.
DELETE	Displays deleted records (includes moved records).
EQUAL	Displays equal records. These are records that have not changed.
NONE	Displays summary information only; no records display. This is mutually exclusive with ADD, ALL, DELETE, EQUAL, and UPDATE.
UPDATE	Displays updated records.

DETECT Subparameter Example

Two versions of a library residing on two different LPARs are loaded and compared. The COMPARE command detects the match, mismatch, sync and out-of-sync conditions between the two libraries. For each out-of-sync condition, invoke the DATACOMPARE function to provide a byte-level comparison.

```

BASE=DSN=LPARA.SYS1.PARMLIB
TARG=DSN=LPARB.SYS1.PARMLIB
COMPARE=MEMTOMEM=DETECT=(SBS=NO,
    COMPLETE=YES,
    ADD,UPD,DEL)

```

DASDRED Command

The DASDRED command performs DASD redundancy management, in which a large pool of volumes is fingerprinted to identify data sets and members with identical contents, independent of their data set names. Such data sets can be evaluated for possible deletion.

The command is clock controlled and restartable; see ["CLOCK Command" on page 134](#) and ["RESTART Command" on page 190](#).

See *Chapter 5: Redundancy Detection and Management*, in the *Getting Started Guide* for additional discussion of redundancy management functions.

The general format of the DASDRED command follows:

```

DASDRED = ( EXCLDSN = ( _,... ),
            EXCLVOL = ( _,... ),
            DD = ( _ ) | DSN = ( _ ),
            INCLDSN = ( _,... ),
            INCLVOL = ( _,... ),
            TYPE = ( _ ) )

```

Either DD or DSN is required; both refer to the fingerprint token data set.

DASD redundancy management requires a pre-allocated token data set. The size of this file varies with the number of data sets and PDS members to be fingerprinted. For an entire DASD volume, approximately three cylinders of space is required.

The DCB attributes for a token data set are:

```
DSORG=PS, RECFM=VB, LRECL=1000, BLKSIZE=6000
```

This table describes the DASDRED command parameters.

Parameter	Description
EXCLDSN	<p>Excludes data sets when using the INCLVOL or EXCLVOL parameter. The default is that all partitioned data sets, physical sequential data sets, and direct access data sets are selected (if not limited by the TYPE parameter).</p> <p>You can specify multiple patterns, separated by commas.</p> <p>You can use wildcards (see "Member Name Wildcards" on page 249 for details).</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>

Parameter	Description
EXCLVOL	Excludes explicit or pattern of DASD volumes from the process. The parameters are either explicit six-character volume serial numbers, or generic volume serial numbers ending with an asterisk. Patterns and explicit volume serial numbers can be present in the same definition. Multiple serial numbers can be specified, separated by commas.
DD	Specifies an external data set (the token data set) in which the results of volume fingerprint tokens are stored. The design of this data set provides restart for the DASDRED process. Refer to "DASDRED Command Example 2" on page 150 for additional information.
DSN	Specifies the name of an external data set (the token data set) in which the results of volume fingerprint tokens are stored.
INCLDSN	Selects data sets when using the INCLVOL or EXCLVOL parameter. The default is that all partitioned data sets, physical sequential data sets, and direct access data sets are selected (if not limited by the TYPE parameter). You can specify multiple patterns, separated by commas. Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.
INCLVOL	Selects explicit or pattern of DASD volumes. The subparameters are either explicit six-character volume serial numbers, or generic volume serial numbers ending with an asterisk. You can specify multiple volume serial numbers, separated by commas. Patterns and explicit volume serial numbers can be in the same definition. You can use the * and ? wildcards; refer to "Volume Name Wildcards" on page 250 for details.
TYPE	Selects data sets by their DSORG. If not specified, all non-VSAM data sets with a DSORG are selected, unless restricted by the INCLDSN or EXCLDSN subparameters. You can specify more than one type when separated by commas. The valid values for this parameter are: <ul style="list-style-type: none"> PO – Partitioned data sets (source, load, or text). PS – Physical sequential data sets. DA – Direct access data sets. SRC – Source libraries (JCL, panels, messages). LOAD – Load libraries. PDS – Equivalent to PO (alternate description). TEXT – Equivalent to SRC (alternate description). Mutual specifications of SRC and TEXT are redundant. <p>TYPE=LOAD indicates all load libraries; TYPE=(PS,DA,TEXT) indicates all other data set types that can be fingerprinted.</p>

DASDRED Command Example 1

The first command controls the time allowed for the task before it stops the fingerprint process. When the allocated time is expired, the task continues with normal wrap-up, and allows the subsequent restart of the job.

```
CLOCK=30
DASDRED=(INCLVOL=PROD*,EXCLVOL=PROD09,
         DSN=CMNSSM.TOKEN)
```

DASDRED Command Example 2

This is an example of the restart process. The first command allows the job to continue the process for 2 hours.

When the RESTART command is initiated, it knows about the original DASDRED command (through the RESTART DD specification), and continues to finish it.

The in-progress token data set must be referred to by the DD=OUTDD parameter, where the OUTDD is supplied in the job stream.

If the time expires again, the program will remember to restart on the volume where it left off, and at the data set that was last processed.

```
CLOCK=120
RESTART=DD=OUTDD
```

DASDRED Command Example 3

This example selects all the available DASD volumes, excluding the ones that start with TEST. From this pool of volumes, it fingerprints only load libraries encountered. The absence of the CLOCK parameter indicates that there is no time limit for the task.

If the job terminates, it can be restarted without losing any of the work that had been completed.

```
DASDRED=(INCLVOL=*,EXCLVOL=TEST*,
         TYPE=LOAD,DD=OUTDD)
```

DASDRED Command Example 4

All volume serial numbers beginning with TEST are selected.

```
DASDRED=(INCLVOL=TEST*,DD=OUTDD)
```

DASDRED Command Example 5

All available DASD volumes are selected.

```
DASDRED=(INCLVOL=*,DD=OUTDD)
```

DASDRED Command Example 6

All partitioned data sets on the specified volume (VOL001) are selected.

```
DASDRED=(INCLVOL=VOL001,TYPE=PO,DD=OUTDD)
```

ENQUEUE Command

Requests that ENQUEUE processing be turned on or off during the EXPORT or IMPORT operations.

You must determine and specify the appropriate ENQUEUE parameter for the EXPORT/IMPORT functions that internally invoke IEBCOPY or PDSFAST.

The general format of the ENQUEUE command follows:

```
ENQUEUE = ( Y | N )
```

This table describes the ENQ command parameters.

Parameter	Description
YES	Attempts to allocate all data sets involved with DISP=OLD. If another task already has exclusive control over the data set, the DISP=OLD allocation fails, and the EXPORT from the data set or the IMPORT into the data set fails.
NO	Attempts to allocate the data set with the DISP=SHR.

ENQUEUE Command Example

In this example, the differences between the two environments are identified using the EXTRACT command.

Before invoking IEBCOPY to offload the changes from the base environment, the data set is allocated with DISP=OLD to provide exclusive control, and if successful, the offload begins.

If a data set in the base environment fails the allocation, the process continues with the next data set to be fingerprinted.

```
ENQ=Y
BASE=LOAD=(DD=FGPNEW)
TARG=LOAD=(DD=FGPOLD)
EXTRACT=DSNBYDSN
EXPORT=(DD=BASKET)
```

EXPORT Command

The EXPORT command unloads the PDS changes (ADD, DELETE, and UPDATE), along with the PS, DA, and new data sets, to a change basket.

EXPORT internally invokes IEBCOPY (or its functional equivalent, PDSFAST), and automatically excludes VSAM data sets.

If one or more orphaned members are identified and not corrected during the export process, the process is stopped. To force the export process to continue despite the occurrence of orphaned members, specify the ORPHANCHECK=N (refer to ["ORPHANCHECK Command" on page 188](#) for details).

The report output produced by the EXPORT command may contain keywords to the left of the data set name in the Data Set Actions section. The keywords and explanations follow:

Keyword	Explanation
Enc	The data set contains members with encrypted names (the member name contains non-printable characters).
Scr	The data set will be scratched from the Target environment.
Upd	There are updated members in the data set that will be copied into the change basket.

The general format of the EXPORT command follows:

```
EXPORT = ( DD = ( _ ) | DSN=( _ ), COMMENT= ' ',
          EXCLDSN = ( _ , ... ),
          INCLDSN = ( _ , ... ),
          HALTONERR,
          MEMBERS )
```

The following table describes the EXPORT command parameters.

Parameter	Description
DD	Identifies an output change basket data set. The change basket is a single repository of changed PDS members, physical sequential, or direct access files determined in the super directory in the beginning of the change basket data set.
DSN	Identifies a pre-allocated change basket data set. A change basket is a single repository of changed PDS members, physical sequential, or direct access files determined in the super directory, during extract processing.
COMMENT	Allows the input of a comment (up to 40 characters).
EXCLDSN	Excludes a pattern of data sets during the export of changes. If EXCLDSN is not supplied, all data sets are eligible for inclusion (default). You can specify the INCLDSN parameter with EXCLDSN Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.
INCLDSN	Includes a specific pattern of data sets. If this parameter is not supplied, all data sets represented by the base or target environments are included (default). Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.
HALTONERR	Causes processing to stop with a return code of 8 if errors such as, but not limited to, the following occur: <ul style="list-style-type: none"> ■ Data sets in use ■ SERCOPY RC=8 ■ Damaged basket detected If the HALTONERR parameter is not specified, processing tries to continue if errors such as the above are encountered. However, the return code is always set to 8 if these types of errors occur, whether processing continues or not.

Parameter	Description
MEMBERS	Selects only the members of the specified data sets. To restrict the data set selections, use the INCLDSN or EXCLDSN parameters.

EXPORT Command Example 1

The two BASE and TARGET LOAD commands establish the base and target environments. The base reflects a newly-changed environment.

The first EXTRACT command directs ChangeMan SSM to identify all changed data sets and PDS members (added, deleted, renamed, or updated).

The second EXTRACT command specifies the inclusion of any newly-found data set in the base that was not in the target. This process constructs a super directory in memory.

The EXPORT command creates a single change basket pointed to by the BASKET DD specified in the JCL.

ORPHANCHECK=N indicates that the presence of orphaned members in the data sets do not prevent the EXTRACT command from processing.

```
ORPHANCHECK=N
BASE=LOAD=DD=FGPNEW
TARG=LOAD=DD=FGPOLD
EXTRACT=DSNBYDSN
EXTRACT=NEWDSN
EXPORT=DD=BASKET
```

EXPORT Command Example 2

The EXPORT command invokes IEBCOPY (or PDSFAST) which physically unloads the changed members and any physical sequential or direct access files to the change basket data set. This single data set can then be transmitted/transported to another site, or another LPAR for IMPORT processing.

```
EXPORT=DSN=CMNSSM.BASKET
```

EXPORT Command Example 3

Use the COMMENT command to type a 40-character description.

```
EXPORT=DD=(ddname,COMMENT = 'ddname has a wicker basket')
```

EXTENDED SYNC Command

EXTENDED SYNC allows you to specify that PDS directory information (such as the following) be used in the change determination process during comparison and synchronization functions:

- PDS member-level directory information (ID, size, mod, VVMM, date created/changed).
- Load module attributes (AC, RENT, RUSE).
- Link-edit date in the translator record (the non-executable section of the module).

This additional information is in the fingerprint data set separately, and is not part of the directory token.

You can abbreviate EXTENDED SYNC as EXTENDED.

The format of the EXTENDED SYNC command is:

```
EXTENDED SYNC= ( Y | N )
```

The following table describes the EXTENDED SYNC command parameters.

Parameter	Description
Y	Directs ChangeMan SSM to evaluate the member-level directory token in addition to the token generated for its content. If two members have equal contents but different directory information, they are considered as being different, and are reported or extracted.
N	Directs ChangeMan SSM to not consider the directory differences when reporting or extracting changes. This specification identifies members that have different contents only. N is the default.

EXTRACT Command

EXTRACT investigates base and target data sets, identifying updated physical sequential and direct access files and modified (added, deleted, or updated) PDS members.

This is the first command executed when synchronizing base and target data sets.

If you specified NEWDSN, the data sets present in the base, but absent in the target, are also detected. All such findings result in the creation of a super directory in memory.

To achieve synchronization between two environments, the super directory created in this process is used as a guideline to unload the necessary files and PDS members from the appropriate base data sets to a single change basket data set.

Refer to ["EXPORT Command" on page 151](#) and the ["IMPORT Command" on page 164](#) for more details.

The EXTRACT process applies only to those data sets that match in name. Data sets that do not match in name are not considered, unless the SCRDSN and NEWDSN parameters are specified.

If one or more orphaned members are identified during the extract process and are not corrected, the process stops. To force the extract process to continue despite the occurrence of orphaned members, specify ORPHANCHECK=N (refer to ["ORPHANCHECK Command" on page 188](#) for additional details).

The report output produced by the EXTRACT command may contain keywords to the left of the data set name in the Data Set Actions section. The keywords and explanations follow.

Keyword	Explanation
Enc	The data set contains members with encrypted names (the member name contains non-printable characters).
Scr	The data set will be scratched from the Target environment.
Upd	Updated members in the data set are copied into the change basket.

The general format of the EXTRACT command is:

```
EXTRACT = ( DSNBYDSN | NBYONE,
            NEWDSN, SCRDSN | ALL,
            NEWVOL )
```

This table describes the EXTRACT command parameters.

Parameter	Description
DSNBYDSN	Defines the extract mode as many-to-many, in which the data set name can associate data sets in the base with data sets in the target. Data sets in the base that do not have counterparts in the target can also be extracted using the NEWDSN parameter.
NBYONE	Defines the extract mode as many-to-one, in which a group of <i>n</i> similar type PDS data sets is synchronized with a single consolidated library. NBYONE can break up many-to-one relations into many one-to-one associations if data sets names in the base do not have identical counterparts in the target (for example, <i>SYS1.TEST./</i> and <i>SYS1.PROD./</i>).
NEWDSN	Indicates that any data set that is in the base, but not in the target, must be exported to the change basket. This parameter is applicable when the DSNBYDSN mode is in effect.
SCRDSN	Directs ChangeMan SSM to identify the data sets that are in the target, but not in the base. Such data sets can optionally be scratched during import processing to achieve total synchronization.
ALL	Indicates that both NEWDSN and SCRDSN are in effect.
NEWVOL	Optional. Allows users to assign the target volser for the new data sets in the change basket when EXTRACT is run. The encoded volser can be overridden by the NEWDSNTOVOL parameter during the IMPORT operation. This is the recommended approach.

EXTRACT Command Example 1

The following set of commands loads fingerprint data sets that represent the contents of two volumes.

The ORPHANCHECK=N command indicates that the presence of orphaned members in the data sets will not prevent the EXTRACT command from processing.

The base volume (IPLBB2) is the more recent (maintenance) volume, after all the SMP/E maintenance has been applied.

The first EXTRACT command identifies the changed PDS members or files from the base.

The second EXTRACT command detects any new data sets; the names and attributes of such data sets are stored in the change basket for an export/import function.

```
ORPHANCHECK=N
BASE=LOAD=DSN=IPLBB2.FGP
TARG=LOAD=DSN=IPLAA2.FGP
EXTRACT=DSNBYDSN
EXTRACT=NEWDSN
EXPORT=DD=BASKET
```

EXTRACT Command Example 2

This is the simplest form of the command.

```
EXTRACT=ALL
```

ALL Parameter

ALL is similar to the DSNBYDSN parameter of the EXTRACT command, except that ALL includes new data sets and data sets that are to be scratched in the change basket. The effect is equivalent to specifying NEWDSN and SCRDSN as separate parameters.

Refer to the ["DSNBYDSN Parameter of EXTRACT Command" on page 157](#) for more detailed information.

The general format of the ALL parameter follows:

```
ALL = (INCLDSN = ( _ ),
      EXCLDSN = ( _ ),
      FORCE,
      MODEL = ( _ ),
      REJECT = ( _ ) | MEMBERS = ( _ ) )
```

ALL Parameter Example 1

ALL identifies the data sets that should be newly allocated in the target environment, as well as the ones that are to be scratched. This parameter creates a file that is pointed to by the subsequent statement in the JCL.

```
//HPSSCR DD DSN=CMNSSM.SCRDSN.LIST,
          DISP=(,CATLG,DELETE)...
BASE=LOAD=DSN=IPLBB2.FGP
TARG=LOAD=DSN=IPLAA2.FGP
EXTRACT=ALL
EXPORT=DD=BASKET
```

ALL Parameter Example 2

Two fingerprint data sets representing two software environments are loaded in memory. A pattern of members are excluded from the the change basket.

```
BASE=LOAD=(DD=FGPDD1)
TARG=LOAD=(DD=FGPDD2)
EXTRACT=ALL=(INCLDSN= /,
              EXCLDSN=SYS1.BROADCAST,
              REJECT=SYS1.APFLIB(IEAAPF00,EAAPF01))
EXPORT=DD=BASKETDD
```

DSNBYDSN Parameter of EXTRACT Command

DSNBYDSN synchronizes a base group of m data sets with a target group of n data sets ($m : n$ relationship).

In this mode, the EXTRACT command investigates the two groups of data sets established by the BASE and the TARGET commands, and then identifies the data sets that are absent in the target (but present in the base), and the files that have been updated (as indicated by their fingerprint tokens).

After establishing a match on two PDS data sets, PDS members that were modified (added, deleted, updated, or renamed) are detected. The process continues for all identical data sets; a list of all such data sets and members is created in memory.

To achieve synchronization between two environments, the list created in this process is used as a guideline to unload the necessary files and PDS members from the appropriate base environment to a change basket data set.

See the ["EXPORT Command" on page 151](#) and ["IMPORT Command" on page 164](#) for details.

The general format of this parameter follows:

```
DSNBYDSN = INCLDSN = ( _ ),
           EXCLDSN = ( _ ),
           FORCE,
           MODEL = ( _ ),
           REJECT = ( _ ) | MEMBERS = ( _ ) )
```

The following table describes the DSNBYDSN subparameters.

Subparameter	Description
EXCLDSN	Excludes various subsets of the synchronization data from a fingerprint data set, using patterns. If EXCLDSN is not specified, no data sets are excluded. EXCLDSN can be specified only if INCLDSN has been specified. This specification is for members during the EXPORT process. Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.
FORCE	Allows PS data sets whose LRECL in the BASE is greater than the TARGET to be included in the change basket.

Subparameter	Description
INCLDSN	<p>Selects various subsets of the synchronization data from a fingerprint data set, using patterns.</p> <p>If INCLDSN is not specified, all data sets are included.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
MEMBERS	<p>Specifies a list or a pattern of selected members that are to be extracted and exported to the change basket.</p> <p>If MEMBERS is not specified, all members are extracted, except those that are explicitly rejected by the REJECT parameter.</p> <p>REJECT and MEMBERS are mutually exclusive.</p> <p>MEMBERS limits the members placed in the change basket for the specified data set. All other data sets identified by the base and target environment specifications will be processed normally.</p>
REJECT	<p>Allows you to exclude a list or a pattern of member names from the change basket.</p> <p>REJECT also allows conflicts to be resolved in the EXTRACT process. These conflicting members must be resolved before the super directory data set can be created.</p> <p>REJECT is useable in any of the three forms of the EXTRACT command (DSNBYDSN, NBYONE, or ALL).</p> <p>REJECT and MEMBERS are mutually exclusive.</p>
MODEL	<p>Transforms names from base and target environments that have different naming conventions to correspond with the data set names in both environments.</p> <p>See the discussion on the MODEL for additional details.</p>

DSNBYDSN Parameter Example 1

The DSNBYDSN parameter specifies the data set-by-data set synchronization mode. Only the data sets specified in the pattern are included in the process. The reported conflicts are explicitly rejected to allow the successful completion of the EXTRACT function.

A separate EXTRACT command considers the entry of new data sets (found in the base) to the super directory in memory.

The EXPORT command creates the change basket using the directives in the super directory.

```

BASE=LOAD=DD=FGPNEW
TARG=LOAD=DD=FGPOLD
EXTRACT=DSNBYDSN=(INCLDSN=SYS1/ ,REJECT=SYS1.HELP(SECTA) ,
REJECT=SYS1.LINKLIB(PARSTLIB) ,
REJECT=SYS1.LPALIB(IFG0551L))
EXTRACT=NEWDSN
EXPORT=DD=BASKETDD

```

DSNBYDSN Parameter Example 2

A selected list of members from the two specified data sets is extracted. If there are any other data sets represented by the loaded fingerprint data sets, they are extracted in their entirety.

Refer to ["DSNBYDSN Parameter Example 2" on page 140](#) for another variation.

```
BASE=LOAD=(DD=DD1)
TARG=LOAD=(DD=DD2)
EXTRACT=DSNBYDSN=(INCLDSN=/,
MEMBERS=somnode.DEV.SRC(A,B,C),
MEMBERS=somnode.DEV.LOD(A,B,C))
EXPORT=DD=BASKETDD
```

DSNBYDSN Parameter Example 3

A subset of the fingerprint representations are loaded in memory. From all the members in the two data sets, a selected list is specified.

```
BASE=LOAD=(DD=DD1,
            INCLDSN=(somnode.DEV.SRC,
                    somnode.DEV.LOD))
TARG=LOAD=(DD=DD2,
            INCLDSN=(somnode.DEV.SRC,
                    somnode.DEV.LOD))
EXTRACT=DSNBYDSN=(MEMBERS=somnode.DEV.SRC(A,B,C),
                  MEMBERS=somnode.DEV.LOD(A,B,C))
EXPORT=DSN=somnode.BASKET.A.B.C
```

DSNBYDSN Parameter Example 4

A subset of the fingerprint representations are loaded in memory to generate a change basket for specific members of the specified data set. All other data set representations in the previously-generated fingerprint data set are excluded.

```
ORPHANCHECK=N
MODEL=(NAME=MODEL01,
       TRANSFORM=2TEST=>PROD)
BASE=LOAD=APPLABC.PROD.FGP
TARG=LOAD=APPLABC.TEST.FGP

EXTRACT=DSNBYDSN=(MODEL=MODEL01,
                  INCLDSN=/,
                  EXCLDSN=(APPLABC.TEST.PROCS,
                            APPLABC.TEST.JCL,
                            APPLABC.TEST.SKELS,
                            APPLABC.TEST.PANELS),
                  MEMBERS=APPLABC.TEST.LINKLIB(MOD1,MOD2))

EXPORT=DSN=APPLABC.CHANGE.BASKET
```

DSNBYDSN Parameter Example 5

Consider only the data sets that match in name (explicitly or by transformation model).

```
EXTRACT=DSNBYDSN
```

DSNBYDSN Parameter Example 6

In addition to the ones that match in name, those data sets that are present in the base and absent in the target (NEW) are also considered in the process (offloaded in entirety).

```
EXTRACT=DSNBYDSN
EXTRACT=NEWDSN
```

DSNBYDSN Parameter Example 7

This is similar to the preceding example, with one exception. Those data sets that are present in the target and missing in the base (scratched) are also considered (and noted in the scratch list).

```
EXTRACT=DSNBYDSN
EXTRACT=NEWDSN
EXTRACT=SCRDSN
```

DSNBYDSN Parameter Example 8

The following example

```
EXTRACT=DSNBYDSN
EXTRACT=ALL
```

shows incorrect coding because the first command indicates:

```
Process all matched data sets names
```

while the second command indicates:

```
Process all matched and unmatched data sets
```

NBYPONE Parameter

NBYPONE synchronizes a group of *N* similar-type PDS data sets with a single consolidated library (many to one).

This command investigates the data sets established by the BASE and the TARGET commands, and detects the PDS members that are different (updated, added, deleted, and renamed). It creates a super directory of all such members.

If you are performing an import, synchronizing *N* libraries with a single library, the file attributes of the base and target environments must be identical.

If there are same-named members in the base group that have different fingerprints, a conflict is detected.

Multiple fingerprint tokens belonging to different members constitute a conflict. Association conflicts must be rejected with the REJECT subparameter.

The following restrictions apply with the NBYPONE association:

- the target environment must contain exactly one data set.
- the base environment may contain one or more data sets.
- the base and target data sets must have compatible DCB attributes.
- the DSORGs must be the same.

- if the target LRECL is non-zero, then each base data set must be non-zero, but not greater than the target LRECL.
- if the target LRECL is zero, then each base LRECL must be zero and a blocksize not greater than the target.

The general format of this parameter follows:

```

NBYONE = ( INCLDSN = ( _ ),
           REJECT = ( _ ) | MEMBERS = ( _ ),
           TONAME = ( _ ) )

```

The following table describes the NBYONE subparameters.

Subparameter	Description
INCLDSN	Specifies one or more data sets in the base environment. If the base environment has only one data set, this specification is optional. Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.
MEMBERS	Specifies a list or a pattern of selected members that are to be extracted and exported to the change basket. If MEMBERS is not specified, all members are extracted except those explicitly rejected by the REJECT parameter. REJECT and MEMBERS are mutually exclusive. MEMBERS limits the members placed in the change basket for the specified data set. All other data sets identified by the BASE or TARGET environment specifications are processed normally
REJECT	Allows you to exclude a list or a pattern of member names from the super directory. REJECT allows conflicts to be resolved in the EXTRACT process. These conflicting members must be resolved before the Change Basket data set can be created. REJECT and MEMBERS are mutually exclusive.
TONAME	Specifies a single target data set of the target environment, which is considered in the synchronization function. If the target environment has only one data set, this specification is optional.

NBYONE Parameter Example 1

NBYONE specifies an (N:1) mode of synchronization.

The three libraries from the test environment are to be synchronized with the single consolidated library in the production environment. This sequence of commands identifies the differences, and packages them into the change basket data set.

```

BASE=DSN=(TEST.PROD1,
          TEST.PROD2,
          TEST.PROD3)
TARG=DSN=PROD.LOAD
EXTRACT=NBYONE
EXPORT=DSN=CHANGE.BASKET

```

NBYONE Parameter Example 2

The BASE and TARGET commands fingerprint two software groups, forming a (2:2) environment. Two EXTRACT commands establish the associations of data sets.

An alternative approach would be the use of the MODEL command to define the data set associations.

```
BASE=DSN=(TEST.SRC,  
          TEST.LINKLIB)  
TARG=DSN=(PROD.SRC,  
          PROD.LINKLIB)  
EXTRACT=NBYONE=(TONAME=PROD.SRC,  
                INCLDSN=TEST.SRC)  
EXTRACT=NBYONE=(TONAME=PROD.LINKLIB,  
                INCLDSN=TEST.LINKLIB)  
EXPORT=DSN=CHANGE.BASKET
```

NBYONE Parameter Example 3

A subset of the fingerprint representations are loaded in memory to generate a change basket for specific members of the specified data set. All other data set representations in the previously-generated fingerprint data set are excluded.

```
ORPHANCHECK=N  
BASE=LOAD=(DD=DD1,INCLDSN=SYS7.PROD.LINKLIB)  
*           Load tokens of this data set  
TARG=LOAD=(DD=DD2,INCLDSN=SYS7.TEST.LINKLIB)  
*           Load tokens of this data set  
EXTRACT=NBYONE=(MEMBERS=SYS7.PROD.LINKLIB(MOD1,MOD2))  
BASE=RESET  
TARG=RESET  
*  
BASE=LOAD=(DD=DD1,INCLDSN=SYS7.PROD.PANELS)  
*           Load tokens of this data set  
TARG=LOAD=(DD=DD2,INCLDSN=SYS7.TEST.PANELS)  
*           Load tokens of this data set  
EXTRACT=NBYONE=(MEMBERS=SYS7.PROD.PANELS(PAN001*))  
EXPORT=DSN=SYS7.CHANGE.BASKET
```

NEWDSN Parameter

NEWDSN specifies that data sets in the base that are not in the target must be considered in the export to the change basket. Identification of these data sets is reflected in the change basket.

NEWDSN is applicable only if the DSNBYDSN mode of synchronization has been requested.

The general format for this parameter follows:

```
NEWDSN
```

There are no subparameters for NEWDSN.

NEWDSN Parameter Example

The BASE command fingerprints a package (identified by GENERIC DSN). Because the target environment is null (not specified), all the data sets are packaged by ChangeMan SSM for transfer to one or more target sites.

The EXTRACT commands DSNBYDSN and NEWDSN specify copy (export) all datasets identified by the GENERIC DSN specification (in their entirety) into the change basket.

```
BASE=GENERIC=(DSN=SYS1.CICSR410.*)
EXTRACT=DSNBYDSN
EXTRACT=NEWDSN
EXPORT=DSN=CICS.BASKET
```

NEWVOL Parameter

NEWVOL specifies the target volser of the newly found data sets for import processing.

If not specified, the originating volser is assumed to be used as the target volser.

This volume override parameter can also be specified by an IMPORT command at the target environment.

The general format of this parameter follows:

```
NEWVOL = ( volser )
```

The only NEWVOL subparameter, *volser*, specifies a six-character volser.

NEWVOL Subparameter Example 1

The DSNBYDSN parameter of this EXTRACT command specifies the data set-by-data set synchronization mode.

The second EXTRACT command considers the entry of new data sets (found in the base) to the change basket in memory. The NEWVOL subparameter determines the volser of the new data as it exists in the target environment after the import processing.

```
BASE=LOAD=DD=FGPNEW
TARG=LOAD=DD=FGPOLD
EXTRACT=DSNBYDSN
EXTRACT=(NEWDSN=NEWVOL=VOL001)
EXPORT=DD=BASKETDD
```

NEWVOL Subparameter Example 2

In this example, new data sets are exported to the change basket with the specified encoded volser.

The receiving system allocates the data sets on these predetermined volumes. You can override this with the IMPORT NEWDSNTOVOL parameter.

```
BASE=LOAD=(DSN=SYS7.NEWFGP)
TARG=LOAD=(DSN=SYS7.OLDFGP)
EXTRACT=DSNBYDSN
EXTRACT=(NEWDSN=NEWVOL=PROD01)
EXPORT=DSN=SYS7.BASKET
```

SCRDSN Parameter

SCRDSN directs ChangeMan SSM to identify target data sets for scratching that do not exist in the base environment.

This is an optional parameter.

The general format of the SCRDSN parameter follows:

```
SCRDSN
```

There are no subparameters for SCRDSN.

SCRDSN Parameter Example

The SCRDSN parameter identifies the data sets that should be scratched. The presence of this parameter creates a file that is pointed to by the subsequent JCL statement.

```
//HPSSCR DD DSN=CMNSSM.SCRATCH.LIST,  
          DISP=(,CATLG,DELETE)  
//HPSIN  DD *
```

```
BASE=LOAD=DSN=IPLBB2.FGP  
TARG=LOAD=DSN=IPLAA2.FGP
```

```
EXTRACT=DSNBYSNDSN  
EXTRACT=NEWDSN  
EXTRACT=SCRDSN  
EXPORT=DD=BASKET
```

IDRLOAD Command

The IDRLOAD command causes IDR data and other load module control records to be included in the fingerprinting process.

NOTE For PDSE load libraries, this command causes the entire Program Object to be fingerprinted. In the absence of this command, only the load module is fingerprinted.

The IDRLOAD command is mutually exclusive with load module masking: MASK=(CSECT) and its optional parameters: DATE, TIME, ACB, and MFS.

This command is also supported by the ISPF interface with the Load Module Masking option.

IMPORT Command

Use the IMPORT command to simulate or apply the exported change basket to a target environment. At appropriate points, this function invokes IEBCOPY (or its functional equivalent, PDSFAST) to update the target environment.

The general format of the IMPORT command follows:

```
IMPORT = (APPLY | SIMULATE,
         BASKETDD= | BASKETDSN= ,
         CATLG= (Y|N) | INDIRECT= (CATLG1,CATLG2,...) | SYMBOLIC=( _ ) ,
         DELETE,
         EXCLDSN = ( _ ),
         INCLDSN = ( _ ),
         FORCE,
         HALTONERR,
         MODEL = ( _ ),
         NEWSNTOVOL = (*|volser),
         OLDDSNTOVOL = (*|volser),
         PRECOMPRESS = ( Y|N ),
         REJECT= ( _ ) | MEMBERS=data set_name(mem1,mem2,... ) ,
         REPLACE | NOREPLACE,
         REVERSE,
         REVERSDD | REVERSDSN,
         SCRATCH )
```

To eliminate potential problems when using data transfer products to transmit change baskets, ChangeMan SSM supplies program SERCOPY to compress and convert the change basket into a fixed block, record length 80 file. Refer to the SENDBSK and RECVBSK sample JCL for additional information.

The following table describes the IMPORT command parameters.

Parameter	Description
APPLY	Enables a change basket to be imported into the target environment.
BASKETDD	Identifies an input change basket data set. The change basket is a single repository of changed PDS members, physical sequential, or direct access data sets created by the export process.
BASKETDSN	Identifies an exported change basket data set. The change basket is a single repository of changed PDS members, physical sequential, or direct access files created at a base site or environment.
CATLG	Specifies cataloging of the newly-found data sets in the change basket. CATLG=Y – catalog the data sets in the target environment. (Default) If you specify the INDIRECT subparameter, the data sets should be indirectly cataloged to the user-specified list of catalogs, using the default string of "*****". CATLG=N – do not catalog the data sets. Example: CATLG=(INDIRECT=(catalog1,catalog2), SYMBOLIC=&SYSR2)

Parameter	Description
DELETE	<p>Deletes members that were removed from the base environment from the PDS data sets in the target environment. If this option is not specified, those members are not deleted.</p> <p>NOTE The DELETE parameter can be used with the SIMULATE parameter.</p>
EXCLDSN	<p>Excludes an explicit or pattern of data sets during the import of a change basket.</p> <p>If EXCLDSN is not supplied, all data sets exported in the change basket are eligible for inclusion.</p> <p>During an EXTRACT, EXCLDSN is used for members.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>
FORCE	<p>Forces conflicts detected by ChangeMan SSM onto the target environment.</p> <p>Example 1: If orphaned members are found in the base environment, the process of import will not continue until those conflicts are resolved.</p> <p>Example 2: If the fingerprint token of a data set or a member at the base site has changed since the data was exported, these members will not be imported unless REJECT or FORCE are specified.</p> <p>NOTE The FORCE parameter can be used with the SIMULATE parameter.</p>
HALTONERR	<p>Causes processing to stop with a return code of 8 if errors such as, but not limited to, the following occur:</p> <ul style="list-style-type: none"> ■ Data sets in use ■ SERCOPY RC=8 ■ Damaged basket detected <p>If the HALTONERR parameter is not specified, processing tries to continue if errors such as the above are encountered. However, the return code is always set to 8 if these types of errors occur, whether processing continues or not.</p>
INCLDSN	<p>Specifies an explicit or pattern of data sets during the import of a change basket. INCLDSN affects only the data sets in the change basket.</p> <p>If you do not specify INCLDSN, all data sets exported to the change basket are imported.</p> <p>Refer to Appendix A, "Wildcards, Symbolics, and DCB Attributes" on page 243 for detailed examples.</p>

Parameter	Description
INDIRECT	<p>Indicates that new data sets in the change basket are to be cataloged as an indirect reference.</p> <p>Instead of having a fixed volser in the catalog point to the volume, it is referenced dynamically with "*****" notation. This is applicable to the data sets residing on the SYSRES and the alternate SYSRES. In such cases, NEWDSNTOVOL=<i>volser</i> must be specified, where <i>volser</i> specifies an online DASD volume.</p> <p>For example, consider a new data set such as SYS1.LINKLIB, allocated and populated on the physical volume ESA002. By making the data set indirectly referable, when the alternate SYSRES switches from ESA001 to ESA002, the data set becomes available to the current IPL volume.</p> <p>You can specify multiple catalogs, separated by commas.</p>
MEMBERS	<p>Lists members, or member patterns that are to be imported. If MEMBERS is not specified, all members are imported, except those specified by the REJECT parameter.</p> <p>REJECT and MEMBERS are mutually exclusive.</p> <p>MEMBERS limits the members imported for the specified data set only. All other data sets identified by the base or target environment specifications will be processed normally.</p> <p>To include only the specified members from that specific data set, you must also specify an INCLDSN parameter.</p> <p>MEMBERS affects only members in the change basket.</p>
MODEL	<p>Transforms names in the change basket to correspond with the naming convention used in the target environment.</p> <p>Use MODEL when importing a centrally-created change basket to multiple environments when the data set names are different.</p>

Parameter	Description
NEWDSNTOVOL	<p>Enables you to specify the ultimate target volser. NEWDSNTOVOL applies to the new data sets in the change basket. At the time of import, the target volume allocation may need to be controlled.</p> <ul style="list-style-type: none"> ■ NEWDSNTOVOL=* - allocate new data sets in the storage group allowed by SMS. In this case, SMS determines the target location of the new data sets. <p>If the allocation fails (in the non-SMS environments, for example, or if the data set is not SMS-managed), ChangeMan SSM internally attempts another allocation using the source volume indicated in the change basket.</p> <ul style="list-style-type: none"> ■ NEWDSNTOVOL=volser - attempt to allocate the new data sets on the volume specified (in the SMS environment, the allocation will be intercepted). <p>If SMS denies the allocation, another allocation is attempted without the volser specification.</p> <p>If the second allocation attempt fails, a third allocation is attempted using the source volser specified in the change basket.</p> <p>If NEWDSNTOVOL is not specified, the allocation is attempted using the source volser in the change basket. The source volser could have been overridden at the time of EXTRACT using the NEWVOL parameter.</p>
OLDDSNTVOL	<p>Enables you to specify the ultimate target volser. OLDDSNTVOL applies to the old data sets in the change basket.</p> <ul style="list-style-type: none"> ■ OLDDSNTVOL=* - catalog processing is desired. A system LOCATE is issued to determine if the data set is cataloged, and if so, it is allocated. Otherwise, the allocation is made using the source volser indicated in the change basket. ■ OLDDSNTVOL=volser - updates are confined to the specified volume. Therefore, the volume is allocated and the data set is searched for an allocation and then updated. <p>If OLDDSNTVOL is not specified, the source volser is used to allocate the data set. In case of failure, another attempt is made to allocate the existing data set without the volser specification.</p>
PRECOMPRESS	<p>Compresses the target PDS before applying the updates. To minimize the X37 problems, members that are to be deleted are removed from the library, members that are to be updated are deleted, and the target PDS is compressed. Do not compress the library containing your IEBCOPY/PDSFAST module.</p>

Parameter	Description
REJECT	Rejects conflicts. For example, if members in the target environment are found to have tokens that differ from their original tokens, the import will not proceed until those members are rejected or forced. This is the safeguard feature of ChangeMan SSM that prevents regression. REJECT and MEMBERS are mutually exclusive.
REPLACE	Replaces existing target data sets during the import of a change basket.
NOREPLACE	Default. Prevents existing target data sets from being replaced.
REVERSE	Reverses the import process using the specified reverse change basket created during the original APPLY processing. During IMPORT/APPLY processing, member deletes are noted and performed internally by ChangeMan SSM. No IEBCOPY statements are displayed in the output report.
REVERSDD	Points to the DD of the reverse change basket.
REVERSDSN	Points to the data set name of the reverse change basket.
SCRATCH	Directs ChangeMan SSM to scratch unmatched data sets in the target. This removes data sets from the target that no longer exist at the base.
SIMULATE	Specifies that a simulate run will be performed. The actions that IEBCOPY or PDSFAST will take during APPLY processing are reported. Conflicts and potential regressions are detected before the change basket is applied. The SIMULATE report can be used as a change ticket for the APPLY process. This is the default mode of operation.

IMPORT Command Example 1

The change basket data set is not applied to the target environment. It runs in a simulate mode.

```
IMPORT=(DSN=CHANGE.BASKET,SIMULATE)
```

IMPORT Command Example 2

The change basket is applied, and the designated target PDS members that do not exist in the base are deleted from the target environment.

The new data sets are allocated in the pool. The existing data sets are located through the catalog, then updated by the data stored in the change basket.

Specify the APPLY parameter after the simulation report has been verified.

```
IMPORT=(BASKETDD=DD1,NEWDSNTOVOL=*,OLDDSNTOVOL=*,APPLY,DELETE)
```

Refer to the *ChangeMan SSM Getting Started Guide* for additional examples.

If a data set name is specified with EXCLDSN, it will not generate an IDCAMS statement, nor will it be deleted.

The scratch list data set is a subset of the IMPORT process.

IMPORT Command Example 3

NEWDSNTOVOL=VOL001 indicates volume processing. Every attempt is made to allocate the data set on the desired volume. The new data sets in the change basket are allocated using the volser VOL001. If SMS denies the allocation, another attempt is made without the volser specification. A final allocation attempt is made using the source volser.

The old (existing) data sets are located according to the catalog (OLDDSNTOVOL=*), then allocated on the returned volumes (the asterisk may be interpreted as "allocate the data set wherever it is possible"). Therefore, another attempt is made to use the source in the allocation of the data set.

Because PRECOMPRESS was specified, the data sets involved in the update process are compressed first (internally invoking IEBCOPY or PDSFAST). Prior to compressing the library, the members to be deleted are physically deleted, and the members to be updated are deleted.

Because the SCRATCH parameter was specified, a file referred by the //HPSSCR DD is created containing the data set names of the data sets scratched, from information supplied for the change basket.

An IDCAMS step following the IMPORT step causes the data sets that were in the target but missing from the base to be scratched.

During extract/export processing, if a data set exists in the target fingerprint data set, but not in the base fingerprint data set, ChangeMan SSM notes the missing (scratched) data set name in the generated change basket.

If you specify the SCRATCH parameter, when the change basket is used to perform the subsequent IMPORT, ChangeMan SSM appends an additional scratch step to the IMPORT, and the data sets identified for scratch are deleted.

```
IMPORT=(BASKETDD=DD1,NEWDSNTOVOL=VOL001,OLDDSNTOVOL=*,
        PRECOMPRESS=Y,APPLY,DELETE,SCRATCH)
```

IMPORT Command Example 4

The new data sets in the change basket are allocated using the volume specified (NEWDSNTOVOL). They are allocated on the volume VOL001.

The old (existing) data sets are not cataloged. They are allocated on the volume specified by OLDDSNTOVOL.

Because the PRECOMPRESS parameter was specified, the data sets involved in the update process are compressed first (internally, invoking IEBCOPY or PDSFAST). Prior to compressing the library, the members to be deleted are physically deleted, and the members to be updated are deleted.

REVERSDD points to a DD in the JCL, indicating that a reverse change basket is requested. The IMPORT command can be reversed (selectively or entirely) using this file.

If a regression is detected during the update, it is forced.

Conflicts can be identified during a simulation run of the IMPORT command. They may be rejected or forced.

```

IMPORT=(BASKETDD=DD1,
        NEWSNTOVOL=VOL001,
        OLDDSNTOVOL=VOL001,
        REVERSDD=DD2,
        APPLY,
        DELETE,
        FORCE)

```

IMPORT Command Example 5

This is the simplest example of an IMPORT command. Because the data set allocation parameters (OLDDSNTOVOL and NEWSNTOVOL) are not specified, ChangeMan SSM allocates the new and old data sets using the source volser. If the allocation fails they are allocated without volser specification.

```

IMPORT=(BASKETDD=DD1,APPLY,DELETE)

```

IMPORT Command Example 6

In this example, the new data sets are allocated on the specified volume and cataloged as indirect. NEWSNTOVOL=*volser* must be specified.

```

IMPORT=(BASKETDD=DD1,
        CATLG=INDIRECT,
        NEWSNTOVOL=SER002,
        APPLY,
        DELETE)

```

IMPORT Command Example 7

In this example, only the three members for the specified data set are imported. The remaining data sets in the change basket are not used.

If the INCLDSN parameter was not specified, all updates other than for SYS1.PROCLIB are applied as usual.

```

IMPORT=BASKETDD=DD1,
        INCLDSN=SYS1.PROCLIB,
        MEMBERS=SYS1.PROCLIB(PROC1,PROC2,PROC3),
        APPLY)

```

IMPORT Command Example 8

The INDIRECT sub-parameter applies to the new data sets in the change basket. They are cataloged using the MVS indirect cataloging facility.

Because the SYMBOLIC parameter was specified, that value is used in the catalog instead of the ***** string.

```

IMPORT=(BASKETDD=DD1,
        CATLG=(INDIRECT=(catalog1,catalog2),SYMBOLIC=&SYSR2),
        OLDDSNTOVOL=MVSV5R,
        NEWSNTOVOL=MVSV5R)

```

IMPORT Command Example 9

This example illustrates the use of the MODEL and IMPORT commands.

The MODEL command, used in combination with TRANSFORM, transforms data sets in the change basket to match the target data sets that are to be updated.

The MODEL command is performed first. The naming convention used after the direction symbol (=>) is the same as that used in the INCLDSN and MEMBERS parameters.

A change basket can contain many data sets; however, you can select a particular data set (or sets), along with the selected members from each to IMPORT.

The INCLDSN parameter can contain a list or pattern of data sets. Multiple data set names for the INCLDSN parameters must be placed inside parenthesis, and must be separated by commas.

The MEMBERS parameter must be repeated to allow references to multiple data sets and the corresponding members of interest.

A simple relationship exists between INCLDSN and MEMBERS parameters. During the IMPORT of a change basket, if the INCLDSN parameter is not supplied, all data sets that had been exported to the change basket will be imported; if INCLDSN is used, an explicit or pattern of data sets is specified, so only the specified data sets become effective for the IMPORT job.

Working in conjunction with INCLDSN parameter, the MEMBERS parameter specifies a list of members or member patterns to be imported. If MEMBERS is not specified, all members are imported, except those specified by the REJECT parameter.

```
MODEL=(NAME=MODEL01,
        TRANSFORM=(' 2DETECT'=>'DETNEW' ))
IMPORT=(OLDDSNTOVOL=*,
        NEWSNTOVOL=*,
        MODEL=MODEL01,
        BASKETDD=DD1,
        REVERSDD=DD2,
        DELETE,
        SCRATCH,
        CATLG=Y,
        PRECOMPRESS=YES,
        APPLY,
        INCLDSN=somnode.DETNEW.JCL,
        MEMBERS=somnode.DETNEW.JCL(mem1,mem2))
```

LIST Command

The LIST command provides a hardcopy listing of all data sets or PDS members (and their fingerprints) that comprise the base or target environments.

The general format of the LIST command follows:

```
LIST = ( BASE | TARGET )
```

The following table describes the LIST command parameters.

Parameter	Description
BASE	Provides for the listing of the base environment.
TARGET	Provides for the listing of the target environment.

LIST Command Example 1

The first command explicitly selects the CMNSSM.V810.LOADLIB data set for the base environment.

The second command specifies that the base environment is to be listed as it is generated.

```
BASE=DSN=CMNSSM.V810.LOADLIB
LIST=BASE
```

LIST Command Example 2

The first command establishes the base environment by loading a fingerprint data set that represents a local or remote environment.

The second command list the contents of the fingerprint data set.

```
BASE=LOAD=DSN=PROD.FGP
LIST=BASE
```

BASE or TARGET Parameter of LIST Command

The BASE parameter specifies that the base environment is to be listed with all data sets or data set members, along with their corresponding fingerprints, alias, orphan, and non-standard indication marks. The listing is further controlled with the options of the SETPRINT command.

The general format of the parameter follows:

```
BASE = ( ALIAS | NONSTD | ORPHAN )
TARGET = ( ALIAS | NONSTD | ORPHAN )
```

The following table describes the BASE subparameters.

Subparameter	Description
ALIAS	Specifies a request for identification and reporting of all alias members in the base environment. For example, if X is an alias of primary member Alpha, it points to the same TTR as the primary member. A primary member may have zero or many aliases. In the output report, alias members are enclosed in angle brackets (< and >).
NONSTD	Specifies a request for identification and reporting of all non-standard member names in the base environment. Member names having non-standard characters (such as blank or }) are reported as hexadecimal values enclosed by quotes, preceded by an X.
ORPHAN	Specifies a request for identification and reporting of all orphaned aliases in the base environment. If the primary member of an alias is deleted, the alias becomes an orphan if it is also not deleted. In the output report, orphan members are enclosed by two exclamation marks.

BASE Parameter Example 1

The first command explicitly selects the CMNSSM.V820.SAMPLES data set for the base environment.

The second command specifies that the base environment is to be listed as it is generated.

```
BASE=DSN=CMNSSM.V820.SAMPLES
LIST=BASE
```

BASE Parameter Example 2

The first command selects all system data sets that have names that start with SYS1 from the specified volume.

They are then fingerprinted to establish the base environment.

The two LIST commands list the orphaned and alias members, if any.

```
BASE=VOL=(INCLVOL=SYS42P,INCLDSN=SYS1/)
LIST=BASE=ORPHAN
LIST=BASE=ALIAS
```

MASK Command

MASK identifies and excludes selected data strings in load modules, IMS-generated control blocks, or other text files when creating fingerprints. The masking facility applies masking rules in each environment.

For example, when fingerprinting load modules, the masking facility interrogates the load module at the CSECT level to determine the type of compiler used, and then invokes the proper compile date/time mask.

MASK provides masking facilities in the following areas:

- IMS control blocks (ACB and MFS parameters)
- Text strings (STRING parameter)
- Columns and rows (AREA parameter)
- Language format rules (LANGUAGE parameter)
- Uppercase and lowercase sensitivity (CASE parameter)
- Blanks, spaces, and other white space (SQUEEZE parameter)

If you load two fingerprints with masking, the masking rules must be identical. If they are not identical, an incompatible mask message is returned.

If you want to set masking when loading two fingerprint sets and if only one of the fingerprint data sets was generated with masking, it must be loaded first.

Masking is set (encoded) in the fingerprint data set.

Masking cannot be specified when loading two previously-generated data sets created with no masking. To do so causes an "incompatible mask" message.

The general format of the MASK command is:

```
MASK = ( ACB,
        AREA = ( _ )
        CASE = ( _ ),
        CSECT,
        DATE,
        LANGUAGE = ( _ ),
        MFS,
        SHOW,
        SQUEEZE,
        STRING = ( _ ),
        TIME )
```

This table describes the MASK command parameters.

Parameter	Description
ACB	Applies predetermined masking rules for ACB (Application Control Blocks) of IMS.
AREA	Directs ChangeMan SSM to mask the specified areas. Refer to "AREA Parameter of MASK Command" on page 177 for additional details.
CASE	Indicates UPPER or LOWER conversions for the text before generating the tokens. Refer to "CASE Parameter of MASK Command" on page 178 for additional details.
CSECT	Directs ChangeMan SSM to generate tokens for each individual CSECT in the load module. The load module fingerprint token is a composite of these individual tokens. The order of CSECTs does not affect the generated load module token.
DATE	Directs ChangeMan SSM to find and mask the embedded compile date in load modules. Based on the information in the translator record, the appropriate date format is searched for. Once the date is found, it is masked with binary zeros before generating the load module tokens. As a result, load modules created on different dates produce identical tokens.
LANGUAGE	Directs ChangeMan SSM to apply its pre-defined masking rules for the type of language specified. Refer to "LANGUAGE Parameter of MASK Command" on page 179 for additional detail.
MFS	Applies the defined masking rules for message format services type modules (for IMS).
SHOW	Displays the masking rules in effect. It also reports the areas that are masked. Refer to "SHOW Parameter of MASK Command" on page 180 for additional detail.
SQUEEZE	Converts multiple consecutive blank spaces into a single space. Refer to "SQUEEZE Parameter of MASK Command" on page 181 for additional detail.

Parameter	Description
STRING	Directs ChangeMan SSM to search for the specified string and if found, to mask it with binary zeros. Refer to " STRING Parameter of MASK Command " on page 181 for additional detail.
TIME	Directs ChangeMan SSM to find and mask the embedded compile time in load modules. Based on the information in the translator record, the appropriate time format is searched for. Once the time is found, it is masked with binary zeros before generating the load module tokens. As a result, load modules created at different times produce identical tokens.

MASK Command Example 1

ChangeMan SSM applies the specified masking rules to the environments designated by the BASE and TARGET commands.

This MASK command can appear only once in the command stream, and remains in effect for all the subsequent BASE and TARGET commands.

This command is optional; if used, it must be specified at the beginning of the HPSIN command stream.

```
MASK=(CSECT, DATE, TIME)
BASE=DSN=CMNSSM.V810.LOADLIB
TARG=DSN=CMNSSM.V730.LOADLIB
COMPARE=MEMTOMEM
```

CSECT parsing, as well as automatic DATE and TIME masking are in effect for both sets of base and target environments. These three parameters must be specified together.

MASK Command Example 2

ChangeMan SSM is directed to apply the IMS Application Control Block (ACB) masking rules to the environment designated by the BASE and TARGET commands.

```
MASK=ACB
BASE=DSN=somnode.ACB.MONDAY
TARG=DSN=somnode.ACB.TUESDAY
COMPARE=MEMTOMEM
```

The MASK command indicates that the automatic ACB masking rule is to be applied to the base and target environments.

MASK Command Example 3

If this MASK command is specified, ChangeMan SSM is directed to apply the IMS Message Format Services (MFS) masking rules to the environments designated by the BASE commands.

```
MASK=MFS
BASE=DSN=somnode.MFS.AUG
BASE=SAVE=DSN=somnode.MFS.AUG.FGP
```


The MASK command above indicates that the automatic MFS masking rule is to be applied during the generation and saving of the fingerprint data set. The fingerprint data set will be referenced to detect out-of-sync modules.

MASK Command Example 4

The following MASK command directs ChangeMan SSM to apply the appropriate rule to the relevant data set in a group.

If the environment to be fingerprinted is composed of ACB-type, MFS-type, or standard load libraries, the libraries are identified by ChangeMan SSM during processing, and the proper rule is applied to each type of load module.

When MASK=MFS or MASK=ACB is specified, MASK=CSECT is implied.

```
MASK=(MFS,ACB,CSECT,DATE,TIME)
BASE=GENERIC=DSN=somnode.HPS.*
TARG=LOAD=DSN=AUG.FGP.MASKALL
COMPARE=DSNBYDSN=SHORT
```

This MASK command indicates that five masking rules are in effect; the rules are applied to the libraries in the group as needed.

A fingerprint data set created in August using all masking rules is now used as a reference to detect out-of-sync modules.

AREA Parameter of MASK Command

Area masking indicates masking of one or many areas of fixed or variable records.

The general format of the AREA parameter is:

```
AREA= ( COL =  _ ,
        LEN =  _ )
```

The following table describes the AREA subparameters.

Subparameter	Description
COL	Specifies the starting column of the area to be masked. COL can be a signed integer with the absolute value of 1 through 32760. A negative value indicates the position relative to the end of the record.
LEN	Specifies the length of the area to masked. LEN must be a positive value, not greater than 255.

AREA Parameter Example 1

The first MASK command in this example specifies the area to be masked. Columns 73 through 80 are to be masked with blanks.

The second MASK command squeezes out the white spaces and displays the commands in effect.

```

MASK=AREA=(COL=73,LEN=80)
MASK=(SQUEEZE,SHOW)
BASE=DSN=somnode.TEST.JCL.*
TARG=DSN=somnode.PROD.JCL.*
COMPARE=COMPOSITE

```

AREA Parameter Example 2

The first two MASK commands in the example that follows specify the areas to be masked.

The + symbol (if specified), indicates a column relative to the beginning of the record.

The - symbol indicates a column relative to the end of the record. In this example, The first two columns, and the last eight columns are masked.

```

MASK=AREA=(COL=1,LEN=2)
MASK=AREA=(COL=-8,LEN=8)
MASK=(SQUEEZE,SHOW)

BASE=DSN=somnode.TEST.JCL.*
TARG=DSN=somnode.PROD.JCL.*
COMPARE=COMPOSITE

```

CASE Parameter of MASK Command

CASE is a general masking rule that can be enforced with any other masking rule in effect.

CASE=UPPER causes an internal translation of records to uppercase before the record is passed to the fingerprinting process.

CASE=MIXED is the default, in which no translation is performed.

The general format of CASE is:

```
CASE = ( MIXED | UPPER )
```

This table describes the CASE parameters.

Subparameter	Description
MIXED	Optional. If MIXED is specified, the contents of the file are converted to uppercase before the token is generated. If MIXED is not specified, no conversion is performed.
UPPER	Optional (default). If UPPER is specified, the content of the file is not converted.

CASE Parameter Example

The MASK command translates all lowercase characters to uppercase characters, and also replaces multiple consecutive blanks with a single blank.

The base environment represents a group of production source data sets that is to be compared against a similar environment to identify the differences. In the first command stream, the environment is fingerprinted and the resulting tokens and the masks are stored in the fingerprint data set.

```
MASK=(CASE=UPPER,SQUEEZE)
```

```
BASE=GENERIC=somnode.SRC.*
BASE=SAVE=DSN=somnode.SRC.FGP
```

Later, this fingerprint data set (that represents the production source) is loaded as a reference to be compared against the source environment on another LPAR.

```
BASE=LOAD=DSN=somnode.SRC.FGP
TARGET=GENERIC=somnode.SRC
COMPARE=DSNBYDSN=SHORT
```

No explicit masking has been specified in the comparison function. The implicit masking rules embedded in the loaded fingerprint data set (base) force the live environment (target) to be masked accordingly.

If a fingerprint data set is loaded via a BASE or TARGET command, masking rules will be controlled by the masks embedded in the fingerprint data set.

LANGUAGE Parameter of MASK Command

This category of masking applies a set of pre-defined masking rules to the source or text data sets in the designated base or target environments.

COBOL, Assembler, and JCL are supported, each with its own unique set of masking rules. SQUEEZE masking is applied automatically for all language masking.

You must verify that all data sets involved in the process are of the correct type.

The general format of the LANGUAGE parameter is:

```
LANGUAGE = ( ASSEMBLE ,
             COBOL ,
             JCL )
```

The following table describes the LANGUAGE subparameters.

Subparameter	Description
ASSEMBLE	Treats the source as a source assembler program. BAL and ASSEMBLE are equivalent. <ul style="list-style-type: none"> ■ Columns 73-80 are masked with blanks. ■ Comment lines are masked in entirety. ■ White spaces are squeezed.
COBOL	Treats the source as a source COBOL program. <ul style="list-style-type: none"> ■ Columns 1-6 and 73-80 are masked with blanks. ■ Comment lines are masked in entirety. ■ White spaces are squeezed.

Subparameter	Description
JCL	<p>Treats the source as JCL.</p> <ul style="list-style-type: none"> ■ Columns 73-80 are masked with blanks. ■ SYSIN data is not involved in the masking process. ■ White spaces are squeezed.

Typically, blanks enclosed by quotes are not squeezed in the squeeze process.

LANGUAGE Parameter Example

In this example, the MASK command indicates that COBOL masking rules are to be applied to all source data sets in the base and target environments.

```
MASK=LANG=COBOL

BASE=DSN=somnode . TEST . COB*
TARG=DSN=somnode . PROD . COB*
COMPARE=COMPOSITE
```

SHOW Parameter of MASK Command

SHOW may be used with any other masking rule.

The general format of SHOW is:

```
SHOW
```

SHOW has no subparameters.

SHOW Parameter Example

In this example, MASK translates all lowercase characters to uppercase, and also squeezes out the white spaces in the records.

The base environment represents a group of production source data sets that is to be compared against a similar environment to identify the differences.

In the first command stream the environment is fingerprinted, and the resulting tokens and the masks are stored in the fingerprint data set. The mask areas are displayed by specifying the SHOW parameter.

```
MASK=(CASE=UPPER, SQUEEZE, SHOW)

BASE=GENERIC=somnode . SRC . *
BASE=SAVE=DSN=somnode . SRC . FGP
```

Later, the fingerprint data set is loaded as a reference to be compared against the source environment on another LPAR.

```
BASE=LOAD=DSN=somnode . SRC . FGP
TARGET=GENERIC=somnode . SRC . *
COMPARE=DSNBYDSN=SHORT
```

No explicit masking was specified in the comparison function. The implicit masking rules embedded in the loaded fingerprint data set forces the live environment to be masked accordingly.

Anytime a fingerprint data set is loaded in the BASE or TARGET command, masking rules are controlled by the masks that were embedded in the fingerprint data set when it was generated.

SQUEEZE Parameter of MASK Command

If you specify SQUEEZE, all multiple consecutive blanks are squeezed into a single blank after the MASK rules have been applied, and before the record is passed to the fingerprinting process.

You can use SQUEEZE alone or with any other masking rule.

```
SQUEEZE
```

SQUEEZE has no subparameters.

SQUEEZE Parameter Example

In this example, the MASK command translates all lowercase characters to uppercase characters, and also squeezes out the white spaces in the records (contracts multiple consecutive blanks into a single blank).

The base environment represents a group of production source data sets that are to be compared to a similar environment to identify the differences. In the first command stream, the environment is fingerprinted and the resulting tokens and the masks are stored in the fingerprint data set.

```
MASK=(CASE=UPPER,SQUEEZE)

BASE=GENERIC=somnode.SRC.*
BASE=SAVE=DSN=somnode.SRC.FGP3
```

This fingerprint data set (representing the production source) subsequently is loaded as a reference to be compared against the source environment on another LPAR.

```
BASE=LOAD=DSN=somnode.SRC.FGP
TARGET=GENERIC=somnode.SRC.*

COMPARE=DSNBYDSN=SHORT
```

No explicit masking was specified in the comparison function. The implicit masking rules embedded in the loaded fingerprint data set forces the live environment to be masked accordingly.

Anytime a fingerprint data set is loaded in the BASE or TARGET command, masking rules are controlled by the masks that were embedded in the fingerprint data set when it was generated.

STRING Parameter of MASK Command

STRING causes ChangeMan SSM to mask user-defined strings in the source data sets.

The mask rules apply to the environments designated by the BASE and the TARGET commands.

Multiple MASK commands are allowed in the command stream. They are in effect for all the subsequent BASE or TARGET commands.

STRING is optional; if used, it must be specified at the beginning of the command stream before the environment is defined by the BASE or TARGET commands.

The general format of the STRING parameter is:

```
STRING = ( AFTER = _ ,
           BEFORE = _ ,
           HIBOUND = _ ,
           LOBOUND = _ ,
           VALUE = ' _ ' )
```

The following table describes the STRING subparameters.

Subparameter	Description
AFTER	Optional. Indicates the number of bytes to be masked following the found string. AFTER must be a positive value, not greater than 255.
BEFORE	Optional. Indicates the number of bytes to be masked preceding the found string. BEFORE must be a positive value not greater than 255.
HIBOUND= <i>n</i>	Optional. Defines the high boundary of the area to be searched. The default is -1 (end of the record.)
LOBOUND= <i>n</i>	Optional. Defines the low boundary of the area to be searched. The default is 1 (beginning of the record.)
VALUE=< <i>string</i> >	Specifies the character string to search for. < <i>string</i> > can be a maximum of 24 characters. If < <i>string</i> > is a hex value, it must be preceded by an X, thus: X'0111'

STRING Parameter Example 1

The first MASK command indicates the string to search for, and if found, to mask with blanks.

The next MASK command squeezes multiple consecutive blanks into a single blank. The SHOW parameter in this command reports the masks that are in effect. The MASK rule applies to both base and target environments.

```
MASK=STRING=(VALUE='string1',
              LOBOUND=4,
              HIBOUND=72)
```

```
MASK=(SQUEEZE, SHOW)
```

```
BASE=DSN=CMNSSM.V810.SRC
TARG=DSN=CMNSSM.V730.SRC
COMPARE=MEMTOMEM
```

```
BASE=DSN=CMNSSM.V810.MAC
TARG=DSN=CMNSSM.V730.MAC
COMPARE=MEMTOMEM
```

STRING Parameter Example 2

In this example, the first MASK command indicates that the specified string should be searched for and masked with blanks, along with four bytes before the string and three bytes following the string.

The second MASK command looks for string HPS in the last eight bytes.

The last MASK command squeezes multiple consecutive blanks into a single blank.

SOURCE or TEXT masking does not apply to the load libraries encountered in the specified data set name pattern. These commands are ignored for load libraries.

```
MASK=STRING=(VALUE=X'0111',
              BEFORE=4,
              AFTER=3)
```

```
MASK=STRING=(VALUE='HPS',
              LOBOUND=73,
              HIBOUND=80)
```

```
MASK=(SQUEEZE, SHOW)
```

```
BASE=GENERIC=CMNSSM.V810.*
TARG=GENERIC=CMNSSM.V730.*
COMPARE=MEMTOMEM=(BASEDSN=CMNSSM.V810.SRC,
                  TARGDSN=CMNSSM.V730.SRC)
COMPARE=MEMTOMEM=(BASE=CMNSSM.V810.PANELS,
                  TARG=CMNSSM.V730.PANELS)
```

LOBOUND=1 and HIBOUND= -1 define the entire search area. This is the default setting.

MODEL Command

Data sets involved in operations that compare and synchronize large groups of data sets in local or remote environments may not always adhere to the same naming standards. The MODEL command allows names to be transformed to associate data sets between the two environments. MODEL establishes data set name associations using a text string or specific data set name node replacements.

An example is comparing two SYSRES packs, one with SYS1 high level qualifier nodes in the data sets, and the other with SYS2 high level qualifier nodes. In previous releases of ChangeMan SSM, you would specify each data set, due to the differences in naming

standards. The MODEL command allows you to compare SYS1 to SYS2 groups of data sets with one command.

Other commands (such as COMPARE and EXTRACT) can reference the MODEL command as a subparameter.

The general format of the MODEL command follows:

```
MODEL = ( NAME = ( _ ),
          TRANSFORM = ( ' _ ' => ' _ ' ) )
```



NOTE Only one MODEL command is allowed, but you can specify several TRANSFORM parameters. See "[MODEL Command Example 2](#)" on page 185 for an example of the MODEL command with two TRANSFORM parameters.

This table describes the MODEL command parameters.

Parameter	Description
NAME	Specifies the model name. The name can be up to eight characters; it is referenced in all commands.
TRANSFORM	<p>Specifies the transform elements.</p> <p>Each element consists of a search argument, a direction symbol, and a replace argument.</p> <p>If a search argument is found in a data set name, it is replaced by the replace argument.</p> <p>The arguments need not be the same length. If multiple elements are present, each is applied in the order it appears in the command.</p> <p>The search argument is defined as follows:</p> <ul style="list-style-type: none"> ■ If the search argument starts and ends with single quotes, this is a string search. The entire data set name is searched for the specified character string. ■ If the search argument starts with a number, it is a node pointer. The range is 0 through 22 (the maximum number of nodes in a data set name). The number points to the node position in the data set name that is to be tested for the search argument. Node pointer zero designates the last node. Once found, the node is replaced by the replace argument. ■ If the search argument is not surrounded by quotes and does not start with a number, then the first node is assumed.

The following examples demonstrate the proper use of the MODEL command. Refer to member patterns MDL* in the ChangeMan SSM *Samples* distribution library for more examples.

MODEL Command Example 1

Two remote environments with different naming conventions are to be compared. In the base environment, data set names start with SYS1. In the target, they start with SYS2.

The transformation model is set up to make the name associations.

Data sets that are not matched according to the model are mismatches. Data sets that are associated are compared, member-by-member, to report the out-of-sync or mismatch conditions.

This function allows very large environments, local or remote, to be compared or synchronized with minimal user set-up.

```
MODEL=(NAME=MODEL01,
        TRANSFORM=(SYS1=>SYS2))

BASE=LOAD=DD=SYS1.FGP
TARG=LOAD=DD=SYS2.FGP
COMPARE=DSNBYDSN=(MODEL=MODEL01)
```

MODEL Command Example 2

In the first environment, SYST.LOAD in the data set name will be associated with SYSP.LINKLIB in the second environment. The first node model from MODEL Command Example 1 is also specified. There are no limits to the number of transforms in a single MODEL command.

```
MODEL=(NAME=MODEL01,
        TRANSFORM=('SYST.LOAD'=>'SYSP.LINKLIB',
                  SYS1=>SYS2))

BASE=LOAD=DD=CPU1.FGP
TARG=LOAD=DD=CPU2.FGP
COMPARE=DSNBYDSN=(MODEL=MODEL01)
```

MODEL Command Example 3

This example has a node qualifier that specifies that if the second qualifier starts with SERENA, transform it to HPS.

```
MODEL=(NAME=MODEL01,
        TRANSFORM=(2SERENA=>HPS))
```

MODEL Command Example 4

This example does not have a node qualifier. Therefore, the first qualifiers are considered in the transformation process.

```
MODEL=(NAME=MODEL01,
        TRANSFORM=(SYS1=>SYS3))
```

MODEL Command Example 5

This example has a question mark symbol as part of the node.

The question mark in the search argument matches any characters except a period.

The question mark in the replace argument indicates that the original character in that position is not to be replaced.

```
MODEL=(NAME=MODEL01,
        TRANSFORM=(HPS?61=>SER?62))
```

MODEL Command Example 6

This example has an asterisk as part of the node. The asterisk in the search argument indicates that the rest of the node is considered to be a match.

```
MODEL=(NAME=MODEL01,  
        TRANSFORM=(HPS*=>SER))
```

MODEL Command Example 7

This example creates a change basket of two CICS environments.

```
MODEL=(NAME=MODEL01,  
        TRANS=( 'SYS1.CICS620'=>'SYSP.CICS610' ))  
  
BASE=LOAD=DSN=SYS1.CICS620.FGP  
TARG=LOAD=DSN=SYSP.CICS610.FGP  
EXTRACT=DSNBYDSN=(MODEL=MODEL01)  
EXPORT=(DD=BASKETDD)
```

MODEL Command Example 8

This example applies the change basket created in <Xref>"MODEL Command Example 7" to the target system.

```
ENQ=NO  
IMPORT=(DD=BASKETDD,  
        OLDDSNTOVOL=*,  
        NEWSNTOVOL=*,  
        APPLY,  
        DELETE)
```

MODEL Command Example 9

This example provides a view of additional considerations when specifying multiple transform statements during an IMPORT process.

This is a sample of an incorrect TRANSFORM specification for this particular scenario.

```
MODEL=(NAME=MODEL01,  
        TRANSFORM=('USER271.DEV'=>'USER271.TESTING',  
                  'USER271.TEST'=>'USER271.TESTING'))  
  
*  
IMPORT=(OLDDSNTOVOL=*,  
        NEWSNTOVOL=*,  
        BASKETDD=DD1,  
        MODEL=MODEL01,  
        DELETE,  
        SCRATCH,  
        PRECOMPRESS=YES,  
        INCLDSN=USER271.TEST.PDSE/  
        APPLY)  
  
/*
```

After the first pass (in the preceding sample), the data set USER271.DEV.PDSE is transformed into USER271.TESTING.PDSE.

After the second pass (TEST=>TESTING) the already-transformed data set USER271.TESTING.PDSE becomes USER271.TESTINGING.PDSE.

To avoid this confusion, specify the MODEL command as follows:

```
MODEL=(NAME=MODEL01,
        TRANSFORM=(' .DEV.'=>' .TESTING.' ,
                  ' .TEST.'=>' .TESTING.' )
```

The dot node separator is included to remove ambiguity.

MODEL Command Example 10

This example shows how to compare fingerprints for data sets referenced by alias. TRANSFORM=(BASE_DSN=>TARG_DSN) associates a single BASE DSN to a single TARGET DSN, regardless of the data set names.

```
ENQ=NO
MODEL=(NAME=MODEL01, TRANSFORM=(BASE_DSN=>TARG_DSN))
BASE=LOAD=DSN=somnode.FGP.DAY1
TARG=LOAD=DSN=somnode.FGP.DAY2
EXTRACT=ALL=(MODEL=MODEL01)
EXPORT=(BASKETDD=BASKET,
        COMMENT='CHANGE BASKET DESCRIPTION')
```

NOECHO Command

The NOECHO command suppresses the echo of subsequent commands in the output. This command can be specified anywhere in the HPSIN input stream.

The general format of the NOECHO command is:

```
NOECHO
```

NOECHO, in conjunction with the commenting facility, allows output documentation.

NOECHO has no parameters.

NOECHO Command Example

The first five cards serve as an internal documentation.

NOECHO suppresses the display of the subsequent commands. BASE causes all data sets with the specified pattern to be selected and fingerprinted.

An internal sort creates a list of all tokens in ascending order. A report of all potential redundant members is printed.

```
* -----
* User department: R & D
* Report title: Redundant members elimination before
*               library consolidation
* -----

NOECHO
BASE=(VOL=SER*, INCLDSN=somnode.RD/)
SETPRINT=MULTIPLE=YES
LIST=BASE
```

ONLINE Command

ONLINE verifies that the volumes specified in the list are online. If any of the volumes in the list is offline, the job is stopped before any further processing.

The general format of the ONLINE command is:

```
ONLINE = ( volser, volser, ... )
```

The single ONLINE command parameter, *volser*, identifies the volume (or volumes) that must be online.

ONLINE Command Example

The first command identifies the three volumes that must be online. If any of the volumes in the list are not online, or do not exist in the UCB list, the process is halted.

The second command fingerprints all partitioned data sets, starting with the node DEV.

The last command reports the result of tokens.

```
ONLINE=(VOL001,VOL002,VOL003)
BASE=VOL=(INCLVOL=VOL*,INCLDSN=DEV/,
          TYPE=PO)

BASE=LIST
```

ORPHANCHECK Command

ORPHANCHECK directs ChangeMan SSM to identify the orphaned PDS members during EXTRACT or EXPORT commands.

The general format of the ORPHANCHECK command is:

```
ORPHANCHECK = ( Y | N )
```

The following table describes the ORPHANCHECK command parameters.

Parameter	Description
YES	Default. Flag the condition and stop propagating orphaned members to other data sets. If an orphaned member (an alias with no primary), is found, a warning message is produced. The EXTRACT and EXPORT operation does not continue until the orphaned members are resolved.
NO	Orphaned members are offloaded to the change basket and propagated to the target environment as they are.

ORPHANCHECK Command Example

The first two commands in the example below establish the base and target environments, with the base being a newly-changed environment.

The first EXTRACT command directs ChangeMan SSM to identify all changed data sets and PDS members (added, deleted, renamed, or updated).

The second EXTRACT command indicates the inclusion of any newly-found data set in the base that has been absent from the target. This process constructs a data set and member list in memory.

The EXPORT command creates a change basket pointed to by a BASKET DD specified in the JCL.

ORPHANCHECK=N indicates that the presence of orphaned members in the data sets will not prevent the EXTRACT command from processing.

```
ORPHANCHECK=N
BASE=LOAD=DD=FGPNEW
TARG=LOAD=DD=FGPOLD
```

```
EXTRACT=DSNBYDSN
EXTRACT=NEWDSN
```

```
EXPORT=DD=BASKET
```

REFTARG Command

REFTARG specifies the base environment to be fingerprinted. Using REFTARG, the base environment is described by the data set list from the existing target environment. This helps to eliminate the number of data sets that are unintentionally included in the fingerprinting of the base environment.

For two environments that must be kept synchronized, a group of volumes may clearly specify the environment at the first site, but not at the second. REFTARG allows the second environment to be easily fingerprinted by referring to the first environment.

REFTARG Command Example 1

The TARG command loads the fingerprint data set, which represents the old environment.

The REFTARG command directs the software to fingerprint only those data sets in the old environment. This ensures that the fingerprint data set created with the BASE command contains only the data sets of interest.

```
TARG=LOAD=DSN=somnode.FGP.OLD
REFTARG
BASE=SAVE=DSN=somnode.FGP.NEW
```

REFTARG Command Example 2

The TARG command loads the fingerprint data set, which represents the target environment.

The REFTARG command references the data set list in the target environment as the list of data sets to fingerprint to establish the base environment. This ensures that when the COMPARE command is executed, the base and the target environments will refer to the same universe of data sets.

```
TARG=LOAD=DD=DD1
REFTARG
COMPARE=DSNBYDSN=SHORT
```

REFTARG Command Example 3

If the SITE 1 (target) environment, as defined by a naming convention, contains a subset of the SITE 2 (base) environment, that subset can be fingerprinted at site 2 using the REFTARG command.

```
*SITE 1
TARG=GENERIC=(DSN=(APPLID1.* ,APPLID2.* ) ,
  EXCLDSN=APPLID1.*.TEST)
TARG=SAVE=DSN=somnode.FGP.SITE1

*SITE 2
TARG=LOAD=DSN=somnode.FGP.SITE1
REFTARG
* Establish base environment
COMPARE=DSNBYDSN=SHORT
```

RESTART Command

The RESTART command allows a terminated task to be restarted when the appropriate token file is identified by the DD or DSN parameters.

Fingerprinting a very large pool of volumes (even the entire data center) may take a long time. The available time window may not be sufficient for the process to complete, or there may be a scheduled IPL that can affect the job. Once a DASDRED function is started, the result of the volume hashing is written to an external file at periodic checkpoints.

Refer to "[DASDRED Command](#)" on page 148 for additional detail.

The general format of the RESTART command follows:

```
RESTART = ( DD = ( _ ) | DSN = ( _ ) )
```

The following table describes the RESTART command parameters.

Parameter	Description
DD=	Identifies the external data set that was specified in the DASDRED command. This file has the checkpoint data at the time of interruption. The DCB attributes for a token data set are: DCB=(DSORG=PS, RECFM=VB, LRECL=1000, BLKSIZE=6000)
DSN	Identifies the external data set name that was specified in the DASDRED command. This file has the checkpoint data at the time of interruption. The DCB attributes for a token data set are: DCB=(DSORG=PS, RECFM=VB, LRECL=1000, BLKSIZE=6000)

RESTART Command Example 1

RESTART alone can restart the fingerprinting task that started for a pool of volumes.

```
RESTART=DD=OUTDD
```

RESTART Command Example 2

RESTART restarts the timed-out or terminated fingerprinting task that started for a pool of volumes.

The DSN refers to the data set name referenced in the DASDRED command. This allows the resumption of an interrupted or an incomplete DASDRED function.

The CLOCK command indicates that this job will execute for 120 minutes "wall clock time" before halting.

```
CLOCK=120
RESTART=DSN=POOL.DASD.RDUNDANT
```

SETPRINT Command

SETPRINT specifies what is to be included or excluded from a report. The command can use any or all of its parameters.



NOTE When using SETPRINT with the HFSCOMPARE command, only the three member-level parameters are allowed: SYNC, OUTSYNC, and MISMATCH.

The general format of the SETPRINT command follows:

```
SETPRINT = ( DETAIL = ( Y | N ),
              DIR = ( Y | N ),
              DSNASSOCIATION = ( Y | N ),
              DSNMISMATCH = ( Y | N ),
              DSNOUTSYNC = ( Y | N ),
              DSNSYNC = ( Y | N ),
              EXCEPTIONSONLY = ( Y | N ),
              EXCLDSN = ( YES | NO ),
              INCLDSN = ( YES | NO ),
              MISMATCH = ( Y | N ),
              MULTIPLE = ( Y | N ),
              NOTINCLDSN = ( YES | NO ),
              OUTSYNC = ( Y | N ),
              REJDSN = ( Y | N ),
              SINGLE = ( Y | N ),
              SYNC = ( Y | N ),
              VTOC = ( Y | N ),
              VVDS = ( Y | N ) )
```

SETPRINT can be specified prior to any COMPARE, HFSCOMPARE, or LIST commands.

SYNC, OUTSYNC, and MISMATCH are member-level commands; they act at the member level rather than the data set level.

This table describes the SETPRINT command parameters.

Parameter	Description								
DETAIL	<p>The DETAIL parameter works with the LIST command. It is useful when a large software environment is evaluated for detailed directory information.</p> <ul style="list-style-type: none"> ■ YES – During the execution of the LIST command, this setting creates a new style of report similar to the ISPF browse function. Fingerprints, directory tokens, and detailed load module attributes are included in the report. ■ <u>NO</u> – The DETAIL parameter is not activated. 								
DIR	<p>Includes or excludes partitioned data set directory information from both environments.</p> <ul style="list-style-type: none"> ■ YES – Include the directory information. ■ <u>NO</u> – Exclude the directory information. 								
DSNASSOCIATION	<p>Includes or excludes data sets that are associated according to the transformation models in effect.</p> <ul style="list-style-type: none"> ■ YES – Include the data sets. ■ <u>NO</u> – Exclude the data sets. 								
DSNMISMATCH	<p>Includes or excludes data sets that do not match in their names (missing data sets).</p> <ul style="list-style-type: none"> ■ <u>YES</u> – Include the data sets. ■ NO – Exclude the data sets. 								
DSNOUTSYNC	<p>Includes or excludes data sets that are out-of-sync.</p> <ul style="list-style-type: none"> ■ <u>YES</u> – Include the data sets. ■ NO – Exclude the data sets. 								
DSNSYNC	<p>Includes or excludes data sets that are matched and synchronized in contents.</p> <ul style="list-style-type: none"> ■ YES – Include the data sets. ■ <u>NO</u> – Exclude the data sets. 								
EXCEPTIONSONLY	<p>Includes or excludes exception conditions when DSNBYDSN mode of comparison is used to compare two environments.</p> <ul style="list-style-type: none"> ■ NO – Exclude the exception conditions. Sets all MEMBER and DSN print options to YES. ■ YES – Include the exception conditions. Sets the MEMBER and DSN flags as follows: <table border="0"> <tr> <td><u>MEMBER flags</u></td> <td><u>DSN flags</u></td> </tr> <tr> <td>SYNC=N</td> <td>DSNSYNC=N</td> </tr> <tr> <td>MISMATCH=Y</td> <td>DSNMISMATCH=Y</td> </tr> <tr> <td>OUTSYNC=Y</td> <td>DSNOUTSYNC=Y</td> </tr> </table> <p>NOTE The EXCEPTIONSONLY parameter has no default and is mutually exclusive with any MEMBER and DSN options.</p>	<u>MEMBER flags</u>	<u>DSN flags</u>	SYNC=N	DSNSYNC=N	MISMATCH=Y	DSNMISMATCH=Y	OUTSYNC=Y	DSNOUTSYNC=Y
<u>MEMBER flags</u>	<u>DSN flags</u>								
SYNC=N	DSNSYNC=N								
MISMATCH=Y	DSNMISMATCH=Y								
OUTSYNC=Y	DSNOUTSYNC=Y								

Parameter	Description
EXCLDSN	YES causes an HPS0925I message to be issued for each data set that is specifically excluded in a fingerprinting operation (the results of BASE or TARGET command exclusion parameters). NO (the default) does not generate these messages.
INCLDSN	YES lists all the data sets included in a fingerprinting operation (the results of BASE or TARGET command inclusion parameters). NO (the default) does not output this list.
MISMATCH	Includes or excludes data set members in one environment (base) that are not name-matched with members in the other environment (target). Content is not considered with this parameter. Any base item that does not have a same-named member in the target group is subject to the parameter. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the data set members. ■ NO – Exclude the data set members.
MULTIPLE	Includes or excludes items in either environment that have identical fingerprints (contents), regardless of any relationship to either environment. Any item that has an equal content (fingerprint) companion in either group is subject to this parameter. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the items. ■ NO – Exclude the items.
NOTINCLDSN	YES causes an HPS0907I message to be issued for each tracked data set in the catalog that is not considered for inclusion in a fingerprinting operation (as the result of a BASE or TARGET inclusion command). NO (the default) does not generate these messages.
OUTSYNC	Includes or excludes items in one environment (base) that are not synchronized using fingerprint (content) with items in the other environment (target). Any base item that does not have a content-equal companion in the target group is subject to the parameter. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the items. ■ NO – Exclude the items.
REJDSN	Includes or excludes any data sets rejected during selection by volume (those not meeting pattern requirements; for example, a VSAM file). This parameter shows what was not selected; it is meaningful if volume processing is involved instead of catalog processing. <ul style="list-style-type: none"> ■ YES – Include the data sets. ■ <u>NO</u> – Exclude the data sets.

Parameter	Description
SINGLE	Includes or excludes an item in either environment that has a unique fingerprint (content), regardless of any relationship to either environment. Any item that has no equal content (fingerprint) companion in either group is subject to the parameter. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the item. ■ <u>NO</u> – Exclude the item.
SYNC	Includes or excludes items in one environment (base) that are synchronized through fingerprint with items in the other environment (target). Any base item that has a content equal to the content in the second group is subject to the parameter. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the items. ■ <u>NO</u> – Exclude the items.
VTOC	Includes or excludes VTOC (volume table of contents) information during volume selection at the time the volume is opened. VTOC information such as name, attributes, and extents are reported. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the VTOC items. ■ <u>NO</u> – Exclude the VTOC items.
VVDS	Includes or excludes VVDS (VSAM volume data set) information during volume selection when the volume is opened. VVDS information, such as component names, is reported. <ul style="list-style-type: none"> ■ <u>YES</u> – Include the VVDS items. ■ <u>NO</u> – Exclude the VVDS items.

SETPRINT Command Example 1

This command allows all selected partitioned data set directory information to be reported. Statistics are also reported.

```
SETPRINT=DIR=Y
```

SETPRINT Command Example 2

This command overrides the default and inhibits the mismatched items from being reported.

```
SETPRINT=(MISMATCH=N)
```

SETPRINT Command Example 3

This command states the default and allows items that have identical fingerprints to be printed. Statistics are also reported.

```
SETPRINT=MULTIPLE=Y
```

SETPRINT Command Example 4

This command causes only the out-of-sync items to be reported.

```
SETPRINT=(SYNC=N,OUTSYNC=Y)
```

SETPRINT Command Example 5

This command causes all data sets that are not selected during a volume search to be reported.

```
SETPRINT=REJDSN=Y
```

SETPRINT Command Example 6

This command inhibits the single items from reporting.

```
SETPRINT=SINGLE=N
```

SETPRINT Command Example 7

This command overrides the defaults inhibiting both the sync and out-of-sync reports. Only the statistics are reported.

```
SETPRINT=(SYNC=N,OUTSYNC=N)
```

SETPRINT Command Example 8

This command causes all VTOC information encountered during volume search to be reported.

```
SETPRINT=VTOC=Y
```

SETPRINT Command Example 9

This next command causes all VVDS information encountered during a volume search to be reported.

```
SETPRINT=VVDS=Y
```

SETPRINT Command Example 10

This example generates a report in which the matched data set names with identical contents (SYNC) are not reported. This is important when large environments (local or remote) are compared.

Refer to the member name DSNXDSN1 in the SAMPLES data set.

```
SETPRINT=DSNSYNC=N  
COMPARE=DSNBYDSN=SHORT
```

SETPRINT Command Example 11

This example disallows reporting of the data set names that are associated directly or by using the transformation models.

Refer to the member name DSNXDSN2 in the SAMPLES data set.

```
SETPRINT=DSNASSOCIATION=N  
COMPARE=DSNBYDSN=SHORT
```

SETPRINT Command Example 12

In this example, the `DETAIL=YES` parameter setting works with the `LIST` command to create a new style of report similar to the ISPF browse function. Fingerprints, directory tokens, and detailed load module attributes are also included in the report.

```
SETPRINT=DETAIL=YES
BASE=GENERIC=(DSN=(SYS1.* ,SYS2.* ) ,TYPE=LOAD
LIST=BASE
```

SIMULATE Command

Use the `SIMULATE` command with the `BASE` and `TARGET` commands to simulate the fingerprint process. This command allows you to review your control statements, search the catalog, and generate a list of what will be fingerprinted without actually creating the fingerprint tokens. You must specify the `SIMULATE` command before a `BASE` or `TARGET` command, and applies to all `BASE` and `TARGET` commands following it in the HPSIN input stream.

The general format of the `SIMULATE` command is:

```
SIMULATE
```

`SIMULATE` has no parameters.

SIMULATE Command Example

The first five comment statements serve as internal documentation.

The `BASE` command selects all data sets with the specified pattern, but no fingerprints are created because of the `SIMULATE` command.

```
* -----
* User department: R & D
* Report title: Redundant members elimination before
*               library consolidation
* -----

SIMULATE
BASE=(VOL=SER* ,INCLDSN=somnode.RD/)
```

SMSMAP Command

`SMSMAP` allocates a new data set on an SMS managed volume where the SMS information at the target system does not match the base. `SMSMAP` allows the SMS Storage Class, Management Class, and Data Class to be changed to appropriate values for the target system.

The general format of the SMSMAP command is:

```
SMSMAP=(STORCLASS=(basevalue1, targetvalue1),
        STORCLASS=(basevalue2, targetvalue2),
        MGMTCLASS=(basevalue1, targetvalue1),
        MGMTCLASS=(basevalue2, targetvalue2),
        DATACLASS=(basevalue1, targetvalue1)
        DATACLASS=(basevalue2, targetvalue2))
```

Where *basevalue* is the current class value for some new data sets residing in the change basket, and *targetvalue* is an acceptable class value found at the target site.

SMSMAP allows you to specify multiple STORCLASS, MGMTCLASS, and DATACLASS values, if multiple mappings are required.

You can use the asterisk wildcard, and NULL to indicate a class of NONE.

You can use the IMPORT command and SIMULATE subcommand to get information on the STORCLASS, MGMTCLASS, and DATACLASS of the new data sets in the change basket.

```
IMPORT=(BASKETDD=DD1,
        SIMULATE)
```

SMSMAP Command Example 1

A matching STORCLASS of DEV000 is changed to SERBASE.

MGMTCLASS data sets that are NULL (NONE) are changed to SERPERM.

For the DATACLASS, NONE(NULL) is mapped to NONE(NULL).

```
MODEL=(NAME=MODEL01,
        TRANSFORM=( ' SYNCDEV' => ' USER271 ' ))
```

```
SMSMAP=(STORCLASS=(DEV000, SERBASE),
        MGMTCLASS=(NULL, SERPERM),
        DATACLASS=(NULL, NULL))
```

```
IMPORT=(OLDDSNTOVOL=*,
        NEWSNTOVOL=*,
        BASKETDD=DD1,
        MODEL=MODEL01,
        DELETE,
        SCRATCH,
        PRECOMPRESS=YES,
        APPLY)
```

SMSMAP Command Example 2

A change basket, created in an SMS environment, is imported into a non-SMS environment.

For STORCLASS, MGMTCLASS, and DATACLASS, the asterisk is mapped to NONE(NULL).

```
SMSMAP=(STORCLASS=(*, NULL),
        MGMTCLASS=(*, NULL),
        DATACLASS=(*, NULL))
```

TARGET Command

The TARGET command fingerprints a software environment, or establishes an environment representation by loading a previously-generated fingerprint data set. This command also saves the fingerprint tokens in an external file.

The command takes only one parameter per use. Multiple TARGET commands can establish the target environment.

The general format of the TARGET command is:

```
TARGET = ( DSN | GENERIC | LOAD | RESET | SAVE | VOL )
```

The TARGET command is similar to the BASE command. Refer to ["BASE Command" on page 120](#) for details and examples.

TRACE Command

TRACE specifies whether or not type T messages are to be reported. This information is useful when a detailed report of ChangeMan SSM operation is required.

Not all ChangeMan SSM mainframe facilities generate trace messages. Specifying the TRACE=ON reports trace messages only if they exist.

The general format of the TRACE command is:

```
TRACE = ( OFF | ON )
```

The following table describes the TRACE command parameters.

Parameter	Description
OFF	Default. Suppresses the printing of type T messages. If these messages are not requested, this command may be omitted.
ON	Allows for printing type T messages.

TRACE Command Example

TRACE=ON provides additional information showing the progress of the job. Type T messages are reported.

TRACE=OFF suppresses the type T messages and provides a more readable report.

```
TRACE=ON
BASE=GENERIC=(DSN=TEST,TYPE=PO)
```

VIEWFGP Command

VIEWFGP allows you to view the fingerprint data set content at the data set or member level. This command is allowed for the base and target at the same time, or individually. The base and target environments must have been defined previously. Refer to the VIEWFGP job in the SAMPLES data set.

The format of VIEWFGP is as follows:

```
VIEWFGP=BASE=MEMBERS= ( Y | N )
VIEWFGP=TARG=MEMBERS= ( Y | N )
```

This table describes the VIEWFGP command parameters.

Parameter	Description
BASE TARG	Specifies whether or not to view the base or target environment.
MEMBERS	Specifies whether or not the member list for each data set is to be printed. Y – print the member list. N – suppress printing.

VIEWFGP Command Example

This example of VIEWFGP reports the data sets and their members, for both the base and target environments established by the fingerprint data sets.

```
BASE=LOAD=DD=DD1
TARG=LOAD=DD=DD2
*
VIEWFGP=BASE=MEMBERS=Y
VIEWFGP=TARG=MEMBERS=Y
```

VIEWBASKET Command

VIEWBASKET allows you to view the detailed activity in a change basket or reverse basket. The data set level activities (new, scratch, replace, update), and the member activities (add, update, delete) are provided for each data set.

VIEWBASKET has the following format.

```
VIEWBASKET=BASKETDD=_ | BASKETDSN=_
```

The following table describes the VIEWBASKET command parameters.

Parameter	Description
BASKETDD	Identifies the DD that points to a change basket or reverse basket. BASKETDD is mutually exclusive with BASKETDSN.
BASKETDSN	Identifies the change basket or reverse basket DSN. BASKETDSN is mutually exclusive with BASKETDD.

VIEWBASKET Command Example

This example prints the data set activities included in the *somnode.BASKET.SITE1* change basket.

```
VIEWBASEKET=BASKETDSN=somnode.BASKET.SITE1
```

VOLMAP Command

This facility allows a single change basket to update a pool of DASD volumes according to a user-specified mapping table.

Compared to NEWDSNTOVOL, which directs ChangeMan SSM to allocate the new data sets on a single volume, VOLMAP specifies a pool.

In an SMS environment, if the attempted allocation fails, then the allocation without the volume specification is retried.

VOLMAP has the following format:

```
VOLMAP= (VOLSER=(oldvol1,newvol1),  
VOLSER=(oldvol2,newvol2),...)
```

Each data set in the change basket has a recorded source volsr that indicates where the data set resided in the base environment.

When importing a change basket, VOLMAP enables you to route the allocations according to a table.

The VOLMAP command parameter, *volsr*, is the only required parameter of the VOLMAP command. There are no limits to the number of VOLSER parameters.

VOLMAP Command Example

In the following example, the VOLMAP command associates the volsers of the data sets in the change basket to the ones in the target environment.

First, the change basket member list is read in memory, and all the volsers are translated according to the table, before the actual import begins.

VOLMAP does not conflict with the OLDDSNTVOL and the NEWDSNTOVOL specifications, which are honored immediately before updating the existing data sets, or allocating the new ones.

```
VOLMAP=(VOLSER=(SER001,HPS001),  
VOLSER=(SER002,HPS002),  
VOLSER=(SER003,HPS003))
```


HFS Synchronization Commands

The following commands are available for HFS files, and are documented in the pages that follow:

- HFSBASE
- HFSCOMPARE
- HFSEXPART
- HFSEXTRACT
- HFSIMPORT
- HFSMODEL
- HFSTARGET

These commands allow you to:

- Fingerprint a group of HFS files.
- Compare two fingerprint data sets and report differences.
- Capture changes into a change basket.
- Apply a change basket to synchronize changes.
- Use path modeling to compare files in different locations, as well as update a location with a different path.

Limitations

The following limitations apply:

- The maximum size of the file to be synchronized cannot exceed 4 GB.
- Synchronization of some file attributes, such as file owner and group fields, may require daemon or super user authority.

Security

Each user must have the necessary privileges to access and update files and file attributes.

Using Pattern Names

You can use pattern names in the path name, as follows:

- a single asterisk or question mark in the final directory or file name
- a double asterisk as the final two characters

Single Asterisk or Question Mark

An asterisk or question mark can only be used with the FILE parameter of the HFSBASE and HFSTARGET commands. An asterisk indicates one or more characters; a question mark indicates a single character.

The following example illustrates the use of an asterisk in a *file name*:

The path name: /u/test/data/user20.test.input
can be coded as: /u/test/data/user20.test.* or /u/test/data/*
but not as: /u/test/*/user20.test.input .

The following example illustrates the use of an asterisk in a *directory name*:

The path name: /u/test/data
can be coded as: /u/test/da* or /u/test/*
but not as: /u/*/data .

Double Asterisk

A double asterisk can only be used with the INCLUDE and EXCLUDE parameters of the following commands:

- HFSBASE
- HFSCOMPARE
- HFSEXPOR
- HFSEXTRACT
- HFSIMPORT
- HFSTARGET

All subdirectories and files under the double asterisk are included. For example:

/u/test/** indicates all subdirectories and files under /u/test .

A double asterisk must be the final two characters of the string. For example:

/u/test/** is valid
/u/test/**/data is invalid

HFSBASE Command

The HFSBASE command fingerprints an HFS software environment or establishes an environment representation by loading a previously-generated fingerprint data set. This command also saves the fingerprint tokens in an external file.

HFSBASE is invoked with a single parameter only. Multiple HFSBASE commands can be used to establish the base environment.

The general format of the HFSBASE command follows:

```
HFSBASE = ( FILE=('_' , '_' ) | LOAD=_ | RESET | SAVE=_ ,  
            RECURSE ,  
            INCLUDE=('_' , '_' ) ,  
            EXCLUDE=('_' , '_' ) )
```

This table describes the HFSBASE command parameters.

Parameter	Description
FILE	Specify one or more explicit file names or patterns. Files specified by this parameter are fingerprinted first, before any other processing.\
LOAD	Use the subparameters DD or DSN to specify a DD name or DSN. For example: <code>LOAD=DD=fileone</code> . The LOAD parameter recalls a previously-generated fingerprint data set. When used in the HFSBASE command, it establishes the base environment.
RESET	Resets the base environment. Any subsequent HFSBASE command reestablishes the base environment rather than appending to an existing one. Therefore, multiple environments can be defined (using the HFSBASE command), cleared (with the RESET command), or redefined.
SAVE	Use the subparameters DD or DSN to specify a DD name or DSN. For example: <code>SAVE=DD=fileone</code> . The SAVE parameter specifies that the tokens of the current base environment be saved in a fingerprint data set.
RECURSE	RECURSE applies only to the FILE parameter. Specify RECURSE to include subdirectories in the fingerprint data set. If you do not specify RECURSE, only files in the specified directory are fingerprinted.
INCLUDE	Specify explicit or pattern path/file names to be included in the fingerprinting process of the Base environment. The names specified apply to all parameters except RESET.
EXCLUDE	Specify explicit or pattern path/file names to be excluded from fingerprinting in the Base environment. The names specified apply to all parameters except RESET.

HFSBASE Command Example 1

This example fingerprints a pattern of files, excluding a specific file.

```
HFSBASE=(FILE='/u/smith/subdirectory/te*',
          EXCLUDE='/u/smith/subdirectory/test3')
```

HFSBASE Command Example 2

This example fingerprints a pattern of files, including subdirectories.

```
HFSBASE=(FILE='/u/smith/*',RECURSE)
```

HFSBASE Command Example 3

These examples save the base environment to a fingerprint data set.

```
HFSBASE=SAVE=DSN=USER20.TEST.HFSFGP1
or
HFSBASE=SAVE=DD=FGPDD1
```

HFSBASE Command Example 4

This example resets the base environment.

```
HFSBASE=RESET
```

HFSBASE Command Example 5

These examples load the base environment from a fingerprint data set.

```
HFSBASE=LOAD=DSN=USER20.TEST.HFSFGP3
or
HFSBASE=LOAD=DD=FGPDD1
```

Using the *SIMULATE* Command with *HFSBASE*

The *SIMULATE* command affects the *HFSBASE* command when using the *FILE* or *SAVE* parameters, as follows:

Parameter	Effect of <i>SIMULATE</i> Command
FILE	Hashing of the files is skipped. Directories are traversed, and file lists are built.
SAVE	The fingerprint DD is allocated and de-allocated, but is never opened, written to, or closed.

The *SIMULATE* command does *not* affect the *HFSBASE* command when used with the *LOAD* or *RESET* parameters.

This information applies to the *HFSTARGET* command as well.

HFSCOMPARE Command

The *HFSCOMPARE* command compares two HFS environments represented by fingerprint data sets.

The general format of the *HFSCOMPARE* command follows:

```
HFSCOMPARE = ( FILE=,
                INCLUDE=('_', '_'),
                EXCLUDE=('_', '_'),
                MODEL=_ )
```

The following table describes the *HFSCOMPARE* command parameters.

Parameter	Description
FILE	Specify this parameter to compare tokens for matching file names.
INCLUDE	Specify explicit or pattern path/file names to be included in the compare process.
EXCLUDE	Specify explicit or pattern path/file names to be excluded from the compare process.
MODEL	Specify the HFS model transform name.

HFSCOMPARE Command Example 1

This example compares a base and target environment using a model.

```
HFSBASE=LOAD=DD=BASEDD
HFSTARG=LOAD=DD=TARGDD
HFSCOMPARE=FILE=MODEL=HFS1
```

HFSCOMPARE Command Example 2

This example compares a base and target environment and uses the SETPRINT command to specify what is to be included or excluded from the report.

```
HFSBASE=LOAD=DD=BASEDD
HFSTARG=LOAD=DD=TARGDD
SETPRINT=(SYNC=N,OUTSYNC=Y,MISMATCH=Y)
HFSCOMPARE=FILE
```



NOTE

When using SETPRINT with HFSCOMPARE, only the three member-level parameters are allowed: SYNC, OUTSYNC, and MISMATCH. See ["SETPRINT Command" on page 191](#).

HFSEXPRT Command

The HFSEXPRT command unloads the file changes (ADD, DELETE, and UPDATE) to a change basket.

The general format of the HFSEXPRT command follows:

```
HFSEXPRT = ( BASKETDSN=_, | BASKETDD=_,
             INCLUDE=(' ','_'),
             EXCLUDE=(' ','_'),
             COMMENT=_ )
```

The following table describes the HFSEXPRT command parameters.

Parameter	Description
BASKETDSN	Identifies a pre-allocated change basket data set. The change basket is a single repository of changed files created at a base site or environment.
BASKETDD	Identifies an output change basket data set. The change basket is a single repository of changed files created by the export process.
INCLUDE	Specify explicit or pattern path/file names to be included in the change basket.
EXCLUDE	Specify explicit or pattern path/file names to be excluded from the change basket.
COMMENT	Specify a comment to be included within the change basket data set for future reference.

HFSEXPOR Command Example

This example creates a change basket with the environment created by the EXPORT command.

```
HFSEXPOR=(BASKETDD=BASKDD2,
          COMMENT='HFS changes')
```

Using the SIMULATE Command with HFSEXPOR

When using the SIMULATE command with the HFSEXPOR command, the fingerprints are compared but nothing is written to the change basket.

HFSEXTRACT Command

The HFSEXTRACT command investigates base and target files, identifying updated and modified (added, deleted, or updated) files.

This is the first command executed when synchronizing the base and target data sets.

The general format of the HFSEXTRACT command follows:

```
HFSEXTRACT = ( INCLUDE=('_', '_'),
              EXCLUDE=('_', '_'),
              ALL | DELETE, UPDATE, ADD,
              MODEL=_ )
```

This table describes the HFSEXTRACT command parameters.

Parameter	Description
INCLUDE	Specify explicit or pattern path/file names to be extracted.
EXCLUDE	Specify explicit or pattern path/file names to be excluded from the extract process.
ALL, DELETE, UPDATE, ADD	Specify ALL to select deleted, updated, and added components. Specify DELETE to select only deleted components. Specify UPDATE to select only updated components. Specify ADD to select only added components.
MODEL	Specify the HFS model transform name.

HFSEXTRACT Command Example 1

This example extracts all changes, except one.

```
HFSEXTRACT=(ALL,
            EXCLUDE=('/u/smith/file1'))
```

HFSEXTRACT Command Example 2

This example extracts new and updated files, using a model.

```
HFSEXTRACT=(ADD, UPD, MODEL=HFS1)
```

HFSIMPORT Command

The HFSIMPORT command is used to simulate or apply the exported change basket to a target environment.

The general format of the HFSIMPORT command follows:

```
HFSIMPORT = ( APPLY | SIMULATE,
              BASKETDSN=_, | BASKETDD=_,
              REVERSEDSDN=_, | REVERSEDSD=_,
              MODEL=_,
              FORCE,
              INCLUDE=( '_ ', '_ ' ),
              EXCLUDE=( '_ ', '_ ' ),
              REVERSE )
```

The following table describes the HFSIMPORT command parameters.

Parameter	Description
APPLY	Enables a change basket to be imported into the target environment.
SIMULATE	Specifies that a simulate run will be performed (the change basket is processed but no action is taken). The actions that will take place during APPLY processing are reported. Conflicts and potential regressions are detected before the change basket is applied. The SIMULATE report can be used as a change ticket to the APPLY process. This is the default mode of operation.
BASKETDSN	Identifies an exported change basket data set. The change basket is a single repository of changed files created at a base site or environment.
BASKETDD	Identifies an input change basket data set. The change basket is a single repository of changed files created by the export process.
REVERSEDSDN	Points to the data set name of the reverse change basket. NOTE The reverse change basket stores the files that are updated when the change basket is applied.
REVERSEDSD	Points to the DD name of the reverse change basket. See note above.
MODEL	Specify the name of the the model to use during the Import process. The MODEL parameter transforms names in the change basket to correspond with the naming convention used in the target environment. Use it when importing a centrally-created change basket to multiple environments, when the data set names are different.

Parameter	Description
FORCE	Forces conflicts detected by ChangeMan SSM onto the target environment, such as: <ul style="list-style-type: none"> ■ If a directory does not exist in the target environment, FORCE will cause it to be created. ■ If the fingerprint token of the file at the target site changed since the data was exported, these files will not be imported unless FORCE is specified.
INCLUDE	Specify explicit or pattern path/file names to be included in the import of a change basket. INCLUDE affects only the files in the change basket.
EXCLUDE	Specify explicit or pattern path/file names to be excluded during the import of a change basket.
REVERSE	Reverses the import process using the specified reverse change basket created during the original APPLY processing.

HFSIMPORT Command Example 1

This example imports a change basket.

```
HFSIMPORT=(BASKETDD=BASKDD1,
           EXCLUDE='/u/smith/subd/big')
```

HFSIMPORT Command Example 2

This example imports a change basket using a model.

```
HFSIMPORT=(BASKETDD=BASKDD1, MODEL=HFS1)
```

HFSMODEL Command

Files involved in operations that compare and synchronize large groups of files in a local or remote environment may not always adhere to the same naming standards. The HFSMODEL command allows names to be transformed to associate file names between the two environments. HFSMODEL can establish file name associations using text string replacements

Other commands (such as HFSCOMPARE, HFSEXTRACT, and HFSIMPORT,) can reference the HFSMODEL command as a subparameter.

The general format of the HFSMODEL command follows:

```
HFSMODEL = ( NAME = ( _ ),
            TRANSFORM = ( ' _ ' => ' _ ' ) )
```



NOTE Only one HFSMODEL command is allowed, but several TRANSFORM parameters may be specified.

The following table describes the HFSMODEL command parameters.

Parameter	Description
NAME	Specify the model name. The name can be up to eight characters; it is referenced in all commands.
TRANSFORM	Specify the transform elements. Each element consists of a search argument, a direction symbol, and a replace argument. The search arguments must be surrounded by single quotes. If a search argument is found in a file name, it is replaced by the replace argument. The arguments need not be the same length. If multiple elements are present, each is applied in the order it appears in the command.

HFSMODEL Command Example

This example models one set of libraries to another.

```
HFSMODEL=(NAME=HFS1, TRANSFORM=('ver720'=>'ver730',
                                'ver721'=>'ver732'))
```

HFSTARGET Command

The HFSTARGET command fingerprints an HFS software environment, or establishes an environment representation by loading a previously-generated fingerprint data set. This command also saves the fingerprint tokens in an external file.

The command takes only one parameter per use. Multiple HFSTARGET commands can be used to establish the target environment.

The general format of the HFSTARGET command follows:

```
HFSTARGET= ( FILE=('_', '_') | LOAD=_ | RESET | SAVE=_,
            RECURSE,
            INCLUDE=('_', '_'),
            EXCLUDE=('_', '_') )
```

The HFSTARGET command is similar to the HFSBASE command. See ["HFSBASE Command" on page 202](#).

Chapter 5

Utilities

This chapter describes the utilities that can be accessed from the ChangeMan SSM Utilities Menu.

Utilities Menu Options	212
Transferring Files	213
Option (1) - Compress	213
Option (2) - FTP to a Remote Host	214
Option (3) - Expand	216
Option (4) - Snapshot	217
Option (5) - Snapshot Comparison	219
Option (6) - Convert VSAM Basket	220
Option (7) - Redundancy Detection	221
Option (D) - Demo	224
Option (H) - HFS Demo	226

Utilities Menu Options

Choose Option U from the ChangeMan SSM Main Menu to display the Utilities Menu:

```

HPSUPRI1 ----- ChangeMan SSM Utilities Menu (Ver 8.4.0)-----
Option ==>

Select one of the following options:

 1 Compress                - Compress FGP or Change Basket for transfer
 2 FTP to a Remote Host    - Transfer compressed FGP or Change Basket
 3 Expand                  - Expand compressed FGP or Basket back to source

 4 Snapshot                - Gather DASD/DSN information; VTOC and VVDS level
 5 Snapshot Comparison     - Report matches/mismatches between two Snap files

 6 Convert VSAM Basket     - Edit a VSAM Change Basket for Partial APPLY

 7 Redundancy Detection    - Report redundant (by content) data sets/members

 D Demo                    - ChangeMan SSM guided tutorial
 H HFS Demo                 - ChangeMan SSM HFS support guided tutorial

 X Exit                    - Exit this Utilities Menu

Press END to Exit

```

Following is a brief description of the utilities options:

Option	Description
1	Compress Generates the JCL to compress an FGP or BASKET. The compressed format is reliable for transferring files over a network. BASKETS cannot be reliably sent in their original format using FTP.
2	FTP to a Remote Host Transfers an FGP or BASKET to a remote system using FTP. BASKETS must be compressed before transferring. FGPs may be transferred in either original or compressed format.
3	Expand Generates the JCL to expand a compressed format file back to the original format of the FGP or BASKET.
4	Snapshot Gathers VTOC and VVDS information on volumes into a snap file. The Snapshot Comparison (Option 5) uses two of these files in reporting matches and mismatches.
5	Snapshot Comparison Detects matches and mismatches between two environments. Two snap files representing different environments are required for this comparison. These snap files are created by the Snapshot function (Option 4).
6	Convert VSAM Basket Converts the basket between editable and standard formats.

Option	Description
7	Redundancy Detection Generates reports identifying redundant (identical content) data sets, and members in concatenated libraries.
D	Demo Provides a guided tutorial of ChangeMan SSM. This tutorial is useful in showing the main functionalities and features of the product, as well as giving users hands on experience with the product.
H	HFS Demo Provides a guided tutorial of HFS support in ChangeMan SSM.
X	Exit Exits the Utilities Menu.

Transferring Files

The ChangeMan SSM ISPF utilities support the transfer of FGP and BASKET data sets.



NOTE PDS files are not supported by these utilities. Refer to members SENDPDS and RECVPDS in the SAMPLES library for examples of how to convert a PDS for transmission using TSO XMIT/RECEIVE.

To transfer an FGP or BASKET to a remote system:

- 1 Select Option 1 (Compress) to compress the file to a format suitable for transferring files with FTP.
- 2 Select Option 2 (FTP to a Remote Host) to specify the appropriate parameters for sending the file.
- 3 At the remote site, select Option 3 (Expand) to expand the compressed format file back to the original format of the FGP or BASKET.

Option (1) - Compress

The Compress utility generates the JCL to convert an FGP or BASKET to a compressed-format file, which is ideal for reliable transfer of files to a remote system. If a change basket is not compressed before transferring, there is no guarantee that it will be successfully transferred to the remote system.

Choose Option 1 for the Compress File for Transfer panel.

```

HPSUSEN4 ----- Compress File For Transfer -----
Command ==>

Enter the name of an FGP or BASKET for compression:

  Input DSN (FGP or BASKET) ==> TEST.BASKET

  Type of file to convert   ==> BASKET           (FGP,BASKET)

Output DSN (compressed)    ==> TEST.BASKET.CMPRS

  (New DSN)      Unit ==>      (SYSALLDA,SYSDA,...)
                 Space Units ==> (TRK,CYL,BLK)
                 Primary   ==> (1-99999)
                 Secondary ==> (0-99999)

Member Name ==> SEND      (Saved member name in the user JCL library)

Press ENTER to continue or END to exit

```

Enter the parameters as described in the following table and press Enter to create the JCL. The JCL can then be edited and submitted for processing.

Parameter	Description
Input DSN (FGP or BASKET)	Type the FGP or BASKET data set name to be compressed. It must be a cataloged DSN.
Type of file to compress	Specify the type of input to be compressed. Valid specifications are FGP and BASKET.
Output DSN (compressed)	Type the name of the output data set. If the output file already exists on the system, the existing data will be overlaid with the new data.
New DSN space allocation	Specify the space allocation parameters for a <i>new</i> output data set.
Member Name	Specify the member name in the user JCL library where the generated JCL will be saved.

Option (2) - FTP to a Remote Host

The File Transfer utility enables FTP transfer of FGPs and BASKETs.

You must compress change baskets before transferring. FGPs may be transferred in original or compressed format.

Choose Option 2 for the File Transfer Protocol panel.

```

HPSUFTP ----- File Transfer Protocol (FTP) -----
Command ==>

To select a HOST from a list, place a '/' in the "Select a host" check box.

FTP Parameters
Send/Receive a file      ==> SEND          (SEND or RECV)
Host Name/Address       ==>                _ Select a host
Target machine Userid   ==>
Password                ==>                (Password not displayed)
Type of file to transfer ==> 1          (1=Compressed FGP or BASKET)
                        ==>                (2=Uncompressed FGP)
Dataset name            ==> TEST.BASKET.CMPRS

NEW Allocation Parameters
Space Units             ==>                (TRACK,CYLINDER,BLOCK)
  Primary               ==>                (1-99999)
  Secondary              ==>                (0-99999)

Member Name    ==> FTP      (Saved member name in the user JCL library)

Press ENTER to continue or END to exit

```

Type the parameters as described in the following table and press Enter to transfer the file.

Parameter	Description
Send/Receive a file	Normally, you send a file to the remote site. Specify RECV to receive the file from the remote site. This is useful when corporate firewalls prevent transferring files out of the LAN, but not into the LAN.
Hostname/Address	Specify the Hostname/Address of the system to FTP to. For example: Hostname/Address ==> 10.10.11.7 . Alternatively, you can type a slash in the <i>Select a host</i> checkbox to display a panel where you can: <ul style="list-style-type: none"> ■ select from a list of Host Names/IP addresses that were previously entered. ■ add new entries or update existing entries.
Target Machine Userid	Specify the userid for the target system. This is the same userid that you use to log on to that system.
Password	Specify the password for the target system. This is the same password that you use to log on to the that system. <p>Note: The password typed into this field is visible in the saved JCL. The job output does not display the password.</p>

Parameter	Description
Type of file to transfer	Specify the type of file to be transferred. This controls file format commands used in the transfer process. Type 1 for a compressed FGP or BASKET. Type 2 for an uncompressed FGP.
Dataset Name	Specify the name of the file to be transferred. Note: You must compress BASKETs before transferring. FGPs may be transferred in original or compressed format.
NEW Allocation Parameters	Specify the space allocation parameters to be used for storing the same-named file at the target site. Note: If the file already exists on the remote system, you need not fill in the new allocation parameters. Note that any information in the existing file will be overlaid.
Member Name	Specify the member name in the user JCL library where the generated JCL will be saved.

Option (3) - Expand

This option generates the JCL to expand a compressed file back to the original format of the FGP or BASKET.

Choose Option 3 for the Expand Compressed File panel.

```

HPSUREC4 ----- Expand Compressed File -----
Command ==>

Enter the name of the compressed FGP or BASKET file:

  Input DSN (compressed)  ==>

  Type of compressed file ==> BASKET      (FGP,BASKET)

Output DSN (FGP or BASKET) ==>

  (New FGP/BASKET)   Unit ==>           (SYSALLDA,SYSDA,...)
                    Space Units ==>    (TRK,CYL,BLK)
                    Primary ==>        (1-99999)
                    Secondary ==>      (0-99999)

Member Name ==> RECV      (Saved member name in the user JCL library)

Press ENTER to continue or END to exit

```


Type the parameters as described in the following table and press Enter to create the JCL. The JCL can then be edited and submitted for processing.

Parameter	Description
Input DSN (compressed)	Specify the name of the compressed file to be expanded.
Type of compressed file	Specify the type of compressed file to be expanded: FGP or BASKET.
Output DSN (FGP or BASKET)	Specify the name of the FGP or BASKET to create from the compressed format file. If the data set already exists, it must be compatible with the original record format.
New FGP/ BASKET space allocation parameters	Optional. Specify the parameters to allocate space for a newly-created output data set. If the output FGP or BASKET already exists on the remote system, you do not need to enter new allocation parameters. Note that any information in the existing output data set will be overlaid.
Member Name	Specify the member name in the user JCL library where the generated JCL will be saved.

Option (4) - Snapshot

The Snapshot facility gathers VTOC and VVDS information from one or more volumes.

The utility generates a Snap file that can identify missing data sets, or data sets that have unequal allocations/attributes between two environments. Also, it can identify allocated or deleted data sets at a single environment over a period of time.

The Snapshot Report (Option 5 of the ChangeMan SSM Utilities menu) performs the actual comparison of Snap files.

```

HPSUSNAP ----- ChangeMan SSM Snapshot Facility -----
Command ==>

Specify the following parameters for the Snapshot facility:

List explicit volsers or volser patterns ending with /: ('/' for all volumes)
VOLSER List           ==>
Exclude List          ==>

New or Existing Snap file ==>
(New only) DASD Unit   ==>                (SYSDA,SYSALLDA...)
      Space Units      ==>                (TRK,CYL,BLK)
      Primary          ==>                (1-99999)
      Secondary        ==>                (0-99999)

      Report?         ==> NO              (YES,NO)

Member Name           ==> SNAPJCL        Member name in user JCL library

Press ENTER to create the JCL or press END to exit.

```

Parameters of the Snapshot Facility menu are described in the following table.

Parameter	Description
VOLSER List	Specifies which volumes to include in the snapshot. For multiple volumes, type a list, separated with commas. For example: <i>volnameA,volnameB,..., volumeZ</i>
Exclude List	Specifies which volumes to exclude from the snapshot VOLSER list specification. To exclude multiple volumes, type a list, separated with commas. For example: <i>volnameA,volnameB,..., volumeZ</i>
New or Existing Snap file	Specifies the name of the Snap file to use for output. If a new file is to be created, type appropriate space allocation parameters.
Report?	Generate report. YES – Generate a list of the data sets involved in the snapshot job, and report them in the output. NO – Default. Do not report the data set list to the output (put it in the snap file).
Member Name	Specifies the saved member name in the user JCL library.

The parameters for the new DSN (space units, primary, secondary, record length, and block size), allocate space for a newly-created snap file. If the snap file already exists on the system, you need not fill in the new allocation parameters.

Option (5) - Snapshot Comparison

The Snapshot Comparison utility reports matches/mismatches in two Snap files (Snap files are generated by Option 4 (Snapshot) of the ChangeMan SSM Utilities menu).

```
HPSUSNPR ----- ChangeMan SSM Snapshot Report Menu (Ver 8.4.0) -----
Command ==>

Specify the following parameters for the Snapshot Report:

Specify two SNAP files to compare:
  First Snapfile on SYSUT1 ==>
  Second Snapfile on SYSUT2 ==>
(Optional) Include/Exclude a data set pattern ending with /:
  INCLDSN                ==>
  EXCLDSN                ==>

Reporting Options:
  _ Matches                Report of data sets present on both files
  _ Mismatches on SYSUT1   Report of SYSUT1 data sets missing on SYSUT2
  _ Mismatches on SYSUT2   Report of SYSUT2 data sets missing on SYSUT1
    - OR -

  _ Last Referenced Date Mismatches

Member Name              ==> SNAPJCL           Member name in user JCL library

Press ENTER to create the JCL or press END to exit.
```

Parameters of the ChangeMan SSM Snapshot Report menu are described in the following table.

Parameter	Description
First Snapfile on SYSUT1	Specifies the name of the first Snap file used in this report.
Second Snapfile on SYSUT2	Specifies the name of the second Snap file used in this report.
INCLDSN	Specify up to three data sets to include in the report. Only the specified data sets are included in the report. Multiple data set names must be separated by commas. To include additional data sets, you may modify the generated JCL.
EXCLDSN	Specify up to three data sets to exclude from the report. Only the specified data sets are excluded from the report. Multiple data set names must be separated by commas. To exclude additional data sets, you may modify the generated JCL.
Matches	Type a '/' in this box to display the matching data sets found between the two Snap files.
Mismatches on SYSUT1	Type a '/' in this box to display data sets found in the SYSUT1 Snap file but not in the SYSUT2 Snap file.
Mismatches on SYSUT2	Type a '/' in this box to display data sets found in the SYSUT2 Snap file but not in the SYSUT1 Snap file.

Parameter	Description
Last Referenced Date Mismatches	Type a '/' in this box to display matching data sets with differing last referenced dates.
Member Name	Allows you to change the default member name to a more meaningful name.

Option (6) - Convert VSAM Basket

The Convert VSAM Basket utility (Option 6 in the ChangeMan SSM Utilities menu) lets you edit added, updated, or deleted records before applying the VSAM change basket.

The following scenario describes how to use the Convert VSAM Change Basket utility to edit and apply a VSAM change basket. The example also describes the typical way in which the VSAM change basket is created.

Once a VSAM change basket is created, it must be converted to editable format before it can be edited. This conversion is done by specifying E in the Editable or Standard? field. After editing is complete and the changes are saved, the VSAM change basket must be converted to standard format by specifying S in the Editable or Standard? field.

A VSAM change basket must be in standard format before it can be applied to an environment

Assume that you create (or already have available), a VSAM file that will be modified by a batch job. Before this VSAM file is modified, you will fingerprint the file so that any changes can be detected and saved in a VSAM change basket. Create this fingerprint file using the FGP option under the ChangeMan SSM Detection and Synchronization menu.

Some time later, the batch job made changes to the VSAM file. You now want to pick up the changes into a VSAM change basket, and apply them to another environment. Therefore, you first re-fingerprint the file. This second fingerprint file detects the changes made by the batch job.

Next, create the VSAM change basket by choosing option EXP under the ChangeMan SSM Data Extraction menu.

Finally, with the VSAM change basket successfully created, convert it to editable format. At this point, you can edit the editable VSAM change basket using StarTool™ or a similar editor.

The following restrictions apply when you edit a VSAM change basket:

- The first three records of the file must not be modified or deleted. The first record is a basket identifier. The second record is the base VSAM fingerprint header, and the third is the target VSAM fingerprint header.
- The last record must not be modified or deleted; it contains important summary information.
- Records in the interior data portion of the VSAM change basket can be modified, deleted, or updated as needed.
- The first 16 bytes of each modifiable record are reserved for control information, and they must not be modified.

After the VSAM change basket has been modified and saved, it must be converted back to a standard format VSAM change basket so that it can be applied.

```

HPSUBLK ----- SSM Convert a VSAM Change Basket -----
Command ==>

Enter the name of a VSAM Change BASKET for conversion:

Editable or Standard?   ==> E (E/S) E: Convert to editable format
                        S: Convert to standard format

Standard VSAM Basket   ==>
Editable VSAM Basket   ==>

      (New DSN)         Unit ==>          (SYSALLDA,SYSDA,...)
      Space Units ==>          (TRK,CYL,BLK)
      Primary ==>             (1-99999)
      Secondary ==>          (0-99999)

Member Name ==> VSCONVT (Saved member name in the user JCL library)

Press ENTER to continue or END to exit

```

Parameters of the Convert a VSAM Change Basket menu are described in the following table.

Parameter	Description
Editable or Standard?	Specifies the conversion type. S – convert editable VSAM basket to standard. E – convert standard VSAM basket to editable.
Standard VSAM Basket	Specifies the standard basket name.
Editable VSAM Basket	Specifies the editable basket name.
Member Name	Specifies the saved member name in the user JCL library.

The parameters for the new DSN (unit, space units, primary, and secondary), allocate space for a newly-created blocked or unblocked basket. If the basket already exists on the system, you need not fill in the new allocation parameters.

Option (7) - Redundancy Detection

The Redundancy Detection utility detects and manages redundant (identical content) data sets and members.

The utility creates the JCL to detect redundant data sets and data set members in an entire application, concatenated libraries, volume, or pool of volumes. Data sets involved in the process are fingerprinted, after which the data set tokens and member tokens that were created are stored in a token data set. A post-step sort and report identifies any data sets (or members) that have the same token.

For example, if there are ten data sets that have the same token, they are reported, showing a redundancy level of nine. This indicates that there are nine other data sets with identical contents, and therefore, are potential candidates for deletion.

The benefits of using redundancy detection include:

- frees disk space by reporting redundant data sets and members, thereby identifying candidates for deletion.
- reduces confusion about which data sets and members to maintain.
- simplifies the task of combining multiple libraries and identifying members and modules that have identical contents in concatenated libraries.

Refer to ["Redundancy Report Examples" on page 233](#) to see examples of Redundancy reports.

```

HPSURED   Data Set Group, Volume or DASD Pool Redundancy Detection
Command =====>

Specify Explicit or Pattern Volser(s) to create the TOKENS Data Set. You may
select SUBSET(s) of data sets by specifying the DSN Pattern(s) and/or Type(s).
Multiple specifications can be separated using commas.

Explicit Volser or data set pattern ending with /:
VOLSER           ==> SER001           (Blank: Use existing Token DSN)
DSN (optional)   ==> SER12.DEV.PANELS
DSN type (optional) ==>                (LOAD, SRC, PDS, PO, PS, DA)

New or Existing Token DSN ==> 'TOKEN.DSN'
(New only) DASD Unit ==> SYSDA         (SYSDA, SYSALLDA...)
Space Units       ==> TRK             (TRK, CYL, BLK)
Primary           ==> 5               (1-99999)
Secondary         ==> 1               (0-99999)

Redundancy Report Type ==> M           M- Member D- Data Set B- Both
Descending Sort Order ==> 1           1- Space 2- Redundancy Level
Member Name        ==> DASDRED        Member name in user JCL library

Press ENTER to create the JCL or press End to exit.

```

Parameters of the redundancy detection panel are described in the following table:

Parameter	Description
Explicit or Pattern Volser	An explicit or a volser pattern to establish the environment to be fingerprinted. Leave this field blank if a previously-created token data set is to be involved in the reporting, and no new fingerprinting is to be performed. The <i>volser</i> specification is required if data set selection by pattern or type is specified.
Optional DSN Pattern	A subset of data sets from the specified volumes. For example, SER12.DEV/ indicates all data sets in which the first node is SER12, and the second node begins with DEV.

Parameter	Description
Optional DSN Type	A subset of data sets from the specified volume. For example, PS,PO indicates the selection of all physical sequential data sets and partitioned data sets on the volumes. If no value is specified, all non-VSAM data sets are selected for the fingerprinting process.
New or Existing Token DSN	Name of a token data set to hold data sets or member tokens. This data set is used in later steps of the job (sort and report) to produce the redundancy report. Enclose the name in quotes for an explicit name; otherwise the current TSO userid is prefixed. If this is a new data set, specify the allocation parameters. Typically for a single volume, specify CYL for the space units, and a value of 5 for both the primary and secondary allocation fields. A token data set structure is different than a fingerprint data set. The two are not interchangeable. The DCB attributes for a token data set are: DSORG=PS, RECFM=FB, LRECL=1000, BLKSIZE=6000
DASD Unit	Enter an esoteric name for the allocation of the new data set.
Space Units	Specify TRK, CYL or BLK for the allocation type of the new data set. CYL is recommended if an entire volume (or multiple DASD volumes) are fingerprinted.
Primary	Specify the primary allocation size of the data set; value 1-99999.
Secondary	Specify the secondary allocation size of the data set; value 0-99999.
Redundancy Report Type	Specify report type, where: M – member-level report; identify members that are identical. D – data set level report; identify data sets that are identical. B – request both reports.
Descending Sort Order	Specify report type, where: 1 – generate a space recovery report; analyze the amount of space that could be recovered if redundant data sets (or members) is reduced to the minimum. 2 – generate a redundancy level report (number of data sets (or members) that are identical).
Member Name	Specify the saved member name in your JCL library. The default member name can be changed.

The parameters for the new DSN (unit, space units, primary, and secondary), allocate space for a newly-created token DSN. If the token DSN already exists on the system, you need not fill in the new allocation parameters.

Option (D) - Demo

Choose the Demo option from the ChangeMan SSM Utilities menu to get to the ChangeMan SSM Demo menu. The Demo starts with Option 1 and proceeds step-by-step to Option C, as follows:

- Two test environments are created.
- The environments are fingerprinted.
- Changes are made to one environment.
- The changes are detected.
- The changes are resolved through the use of a change basket.
- You are guided through some of the basics of change tracking, including recovering backed up members from a tracked data set.

In Options A, B, and C, you learn how to use Member Reference Tracking (MRT) to track and report references to specific data sets and members.

```
HPS@DEMO ----- SSM DEMO MENU (Ver 8.4.0) -----
Option ==>

Select One of the demo options below:

 1 Create a Demo Environment - Create two environments LPARa and LPARb

 2 Fingerprint the Environment - Establish a representation of LPARa and LPARb
 3 Change the Environment      - Make changes in the BASE environment LPARa
 4 Detect Changes              - Detect changes made in the BASE environment
 5 Resolve Changes             - Synchronize TARGET with BASE (LPARa -> LPARb)
 U Undo                        - Undo Changes (Apply Reverse Basket)

 6 Define a Group              - Define a Group for tracking LPARa DSN's
 7 Define Objects              - Define data sets (Objects) to the Group
 8 Incur Change                 - Update the LPARa tracked data sets
 9 Recover a Member Version     - Recover a prior version of PDS member

 A Define MRT Patterns         - Define data sets to Member Reference Tracking
 B Reference Members           - Create some member references
 C MRT Report                   - Report tracked member references

Choose X to Exit
```

Items on the Demo menu are described in the following table:

Option	Description
Create a Demo Environment	Invoke a panel that builds the JCL for creating two demo environments (LPARa and LPARb), by using the distribution libraries as input. The two environments are used for fingerprinting, changing, detecting changes, and resolving the changes. LPARa is the base environment, and LPARb is the target environment.
Fingerprint the Environment	Establish a fingerprint representation of the LPARa and LPARb environments. Later, these fingerprint representations are used as references in detecting changes.

Option	Description
Change the Environment	<p>As a part of the demo, change the base environment (LPARa).</p> <p>The base environment to be changed consists of the following data sets:</p> <ul style="list-style-type: none"> ■ USERID.LPARA.INSTALL ■ USERID.LPARA.PANELS ■ USERID.LPARA.MSGS ■ USERID.LPARA.LINKLIB <p>Changes made to any of these data sets will be detected by Option 4 (Detect Changes).</p>
Detect Changes	<p>Detect changes made to LPARa (base environment).</p> <p>First, the environment is re-fingerprinted. Then, the new fingerprint file is compared to the reference fingerprint to identify the changes.</p>
Resolve Changes	<p>Synchronize the base (LPARa) environment with the target (LPARb) environment. This demonstrates how synchronization can be used to resolve changes.</p>
Undo	<p>Undo the changes made to your environment from running this demo application.</p>
Define a Group	<p>Define a group of data sets for tracking. You must already be defined as a group administrator to perform this.</p> <p>If this option is selected, a panel lists a sequence of steps for defining a data set group for tracking.</p>
Define Objects	<p>Define data sets for the group created by Option 6 (Define a Group).</p>
Incur Change	<p>Update any one of the tracked data sets.</p>
Recover a Member Version	<p>Guides you through the recovery of a previous member version stored in the delta master database.</p>
Define MRT Patterns	<p>In this step, the data sets for the LPARa environment are defined to the Member Reference Tracking system.</p>
Reference Members	<p>In this step, you create some member references by browsing data sets defined in the previous step.</p>
MRT Report	<p>To report member references, the HPSMRTRP program is executed in a batch job. Various parameters can be specified to tailor the report output.</p>
Exit Demo	<p>Exits the demo.</p>

Option (H) - HFS Demo

Choose the HFS Demo option from the ChangeMan SSM Utilities menu to get to the HFS Demo Menu.

```
HPS@HFSD ----- SSM HFS DEMO MENU (Ver 8.4.0) -----
Option ==>

Select One of the demo options below:

 1 Create a Demo Environment - Create two environments HFSBASE and HFSTARG
 2 Fingerprint the Environment - Establish a representation the environments
 3 Change the Environment    - Make changes in the BASE environment
 4 Detect Changes            - Detect changes made in the BASE environment
 5 Resolve Changes          - Synchronize TARGET with BASE
 6 Undo                      - Undo Changes (Apply Reverse Basket)

Choose X to Exit
```

Start with Option 1 and proceed through Option 6 to perform the following functions:

Option	Description
1 - Create a Demo Environment	Invokes a panel that builds the JCL to create two demo environments (HFSBASE and HFSTARG), using the distribution libraries as input. These two environments are used in the remaining options for fingerprinting, changing, detecting changes, and resolving the changes.
2 - Fingerprint the Environment	Invokes a panel that builds the JCL to fingerprint the HFSBASE and HFSTARG environments. The fingerprint representations will be used as references in Option 4 (Detect Changes).
3 - Change the Environment	Invokes a panel that guides you in making changes to the HFSBASE environment. The HFSBASE environment consists of the following data sets: <ul style="list-style-type: none"> ■ /u/userid/ssmdemo/hfsbase/install ■ /u/userid/ssmdemo/hfsbase/panels ■ /u/userid/ssmdemo/hfsbase/msgs Changes made to any of these data sets will be detected in Option 4 (Detect Changes).
4 - Detect Changes	Invokes a panel that builds the JCL to detect changes made to the HFSBASE environment. First, the HFSBASE environment is re-fingerprinted. Then, the new fingerprint file is compared to the reference fingerprint (created in Option 4 - Fingerprint) to identify the changes.

Option	Description
5 - Resolve Changes	<p>Invokes a panel that builds the JCL to synchronize the HFSBASE environment with the HFSTARG environment. To resolve the differences, the HFSBASE environment is fingerprinted to establish a current fingerprint representation in memory. This representation file is then compared against the old fingerprint file, loaded in memory, to identify the changed members.</p> <p>Changes are then packaged into a Change Basket data set, which is applied to the HFSTARG environment to achieve synchronization. After synchronization has completed, both environments are compared to verify the success of the operation.</p>
6 - Undo	<p>Invokes a panel that builds the JCL to undo the changes to the HFSTARG environment.</p> <p>The Reverse Change Basket created in Option 5 (Resolve Changes) will be used to restore the target environment.</p>

Chapter 6

Redundancy Detection and Management

This chapter describes redundancy detection and management for data sets, concatenated libraries, DASD volumes, DASD pools, LPARs, and sites.

The ChangeMan SSM Detection and Synchronization component can locate redundant (by content) PDS members and data sets. Maintaining redundant data may be acceptable to a certain level. However, it is always important to manage the redundancy to prevent overpopulating your DASD pool.

Tokenizing Volumes	230
Restart Tokenizing	231
Determine Redundancy Level	231
Report Redundant Data Sets and Members	233
Redundancy Report Examples	233

Tokenizing Volumes

Consider the following example, in which redundant data sets in a pool of volumes are detected.

```
//JOB CARD
//STEP1 EXEC PGM=HPSTRANS,REGION=4096K
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR
// DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//OUTDD DD DSN=somnode.TOKEN.DATA SET,DISP=SHR
//HPSIN DD *
CLOCK=45
DASDRED=(INCLVOL=(SER01*)
DD=OUTDD,
*)IM $$EXCL
EXCLDSN=(SYS1.BROADCAST,
SYS1.VTOCIX.*,
SYS1.SMPDLOGA,
SYS1.RACF))
```

The following description assumes that a large pool of volumes are fingerprinted over non-contiguous time intervals.

If the job terminates due to clock expiration or any other interruption, it can be re-started with the RESTART command. When restarted, ChangeMan SSM automatically resumes at the point of interruption, without losing the fingerprint tokens already generated.

Once fingerprinting of the pool is completed, the tokens are externally sorted, and identical data sets or members are grouped. In each group of redundant data sets, redundancy level and maximum recoverable space are reported.

If the job is cancelled, it can be restarted without losing any of the processing. In some cases, you may require only redundancy checking at the data set level. This also reduces the size of the token data set PARM=NOMEM that is specified. This excludes the member tokens from being written to the output file. Only data set tokens are written.

A typical benchmark result indicates that for a fully loaded DASD 3390-3 device type, on average, it takes under four minutes (elapsed time), and under one minute CPU time to tokenize an entire volume.

When "tokenizing" a pool of volumes, you may want to run multiple jobs. This allows for balancing of the workload, reducing the time required to complete the task.

The following commands can be specified for redundancy detection and management:

- CLOCK

CLOCK=45 indicates that the job will run for 45 wall clock minutes. When this time expires, the job terminates but may be restarted. All fingerprint tokens created up to this point are saved in a token data set. Do not specify this command if you require the job to run to its successful completion.

If a CLOCK command is not specified, the job runs to completion.

A return code of 15 indicates that the clock expired. If the clock expires, the process can be restarted by using the RESTART function specified in the DASD2 JCL supplied in the ChangeMan SSM Sample Library.

- DASDRED

DASDRED allows you to specify volumes that are to be included in, or excluded from, the tokenizing process.

- OUTDD refers to a physical sequential data set (PS).

The size of this data set depends on the number of data set members in the pool. Based on the number of members, this file can grow as large as two cylinders per fingerprinted volume. A fingerprint token is stored for each member of a PDS and one for each data set.

- The comment, IM \$\$EXCL, indicates that the \$\$EXCL skeleton has been imbedded. This skeleton generates the EXCLDSN parameter which specifies additional data sets to be automatically excluded from volume processing.

Restart Tokenizing

If the tokenizing job is halted, either by the CLOCK parameter or by operator intervention, it can be restarted by executing the following process. The tokenizing process continues from where it left off.

```
//JOB CARD
//STEP1 EXEC PGM=HPSTRANS,REGION=4096K
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR
// DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//RESTRTDD DD DSN=somnode.TOKEN.DATA SET,DISP=SHR
//HPSIN DD *
TRACE=YES
CLOCK=45
RESTART=DD=RESTRTDD Restart volume Fingerprinting
from where it left off.
```

CLOCK=45 is not a required command. Use it only if a clock-controlled function is desired. In this example, the clock is set to 45 minutes (wall clock time).

RESTART requests for a continuation of the job started by the job DASD1, supplied in the ChangeMan SSM Sample Library. The DD specified in the RESTART command in the example refers to the physical sequential data set specified in the DASD1 job.

Determine Redundancy Level

This job does the following:

- The first step sorts the token data set in token number order.
- The second builds a grouped record of redundant data sets or members.

- The last step sorts the file in descending order by either the Maximum Recoverable Space or the Redundancy Level.

The SORTIN data set is input from the tokenizing process.

```
//JOB CARD
//SORT1 EXEC PGM=SORT,REGION=2M
//SORTIN DD DISP=SHR,DSN=somnode.TOKEN.DATA SET
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,5)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,5)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,5)
//SORTOUT DD DISP=(NEW,KEEP,DELETE),
//          DSN=&&TEMP,
//          UNIT=SYSDA,
//          SPACE=(CYL,(1,1)),
//          DCB=(RECFM=VB,LRECL=1000,BLKSIZE=6000)
//SYSIN DD *
          SORT FIELDS=(05,60,CH,A)
          RECORD TYPE=VB,LENGTH=1000
//STEPGRP EXEC PGM=HPSGROUP,REGION=1M
//*-----
/* Input: "Sorted" version of the REDundant file
/* Output: "Grouped" version of the REDundant files/members
/*-----
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR
//          DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//INDD DD DSN=&&TEMP,DISP=(OLD,KEEP,DELETE)
//OUTDD DD DISP=(NEW,CATLG,DELETE),
//          DSN=somnode.DASDRED.GROUPED,
//          UNIT=SYSDA,
//          SPACE=(CYL,(20,5)),
//          DCB=(RECFM=VB,LRECL=23152,BLKSIZE=23200)
//SORT2 EXEC PGM=SORT,REGION=2M
//*-----
/* Use either one of these sort fields:
/* SORT FIELDS=(25,4,CH,D) Max-recoverable
/*                               (excess tracks allocated)
/* SORTmFIELDS=(15,2,CH,D) Highest to lowest Redundancy-
/*                               level
/*-----
//SORTIN DD DISP=SHR,DSN=somnode.DASDRED.GROUPED
//SORTOUT DD DISP=(NEW,CATLG,DELETE),
//          DSN=somnode.DASDRED.GROUPED.SORTED,
//          UNIT=SYSDA,SPACE=(CYL,(20,5)),
//          DCB=(RECFM=VB,LRECL=23152,BLKSIZE=23200)
// SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,5)
// SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,5)
// SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,5)
// SYSOUT DD SYSOUT=*
// SYSIN DD *
          SORT FIELDS=(25,4,CH,D) /* Max recoverable tracks in */
          RECORD TYPE=VB /* redundant groups */
```


Report Redundant Data Sets and Members

The following job reports redundant data sets and members.

The INDD data set is from the token sorting process (the last SORT step in the previous job).

```
//JOB CARD
//REPORT EXEC PGM=HPSREPT,REGION=1M,PARM='DSN,MEM'
//*
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR,
// DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//INDD DD DISP=SHR,DSN=somnode.DASDRED.GROUPED.SORTED
```

The PARM value specified on the EXEC statement determines the types of reports generated:

PARM Values	Type of Report
PARM=DSN	Request for redundant data set report.
PARM=MEM	Request for redundant member report.
PARM='DSN,MEM'	Request for both reports.

Redundancy Report Examples

This section contains examples of the two types of Redundancy Reports, DATASET and MEMBER, requested by PARM='DSN,MEM' on the EXEC statement.

The Redundancy Reports can be generated in two different sort sequences:

- The first reports on the group with the highest recoverable redundancy space to the lowest group.
- The second reports the largest (numerical) redundant data set groups first.



NOTE The sort sequence is specified in the *Descending Sort Order* field on the Redundancy Detection panel (see ["Option \(7\) - Redundancy Detection" on page 221](#)).

The following reports illustrate the second sort sequence. These examples are limited in scope for demonstration purposes. Many shops will be able to identify and recover hundreds or thousands of cylinders of DASD space.

The following example shows the command and parameters used to generate the Redundancy Reports.

```
DASDRED=(INCLVOL=(SRSM*),
          INCLDSN=(BSCHEED.IDENT/,
                  BSCHEED.EMPTY/,
                  BSCHEED.DUPE/,
                  BSCHEED.UNIQUE/),
          DD=REDD)
```

The following example shows some of the messages displayed using the above parameters. ChangeMan SSM has found and fingerprinted the requested data set names as found on each selected VOLSER. There will be many of these messages as each selected DASD volume is searched.

```
HP50902I  HPSTRANS  08:36:29 Volume SRSM5F selected:
HP50911I  HPSTRANS  08:36:29 2 of 273 Data Sets Fingerprinted.

HP50902I  HPSTRANS  08:36:29 Volume SRSM51 selected:
HP50911I  HPSTRANS  08:36:29 0 of 307 Data Sets Fingerprinted.

HP50902I  HPSTRANS  08:36:29 Volume SRSM52 selected:
HP50911I  HPSTRANS  08:36:29 1 of 242 Data Sets Fingerprinted.

HP50197I  HPSTRANS  08:36:39 File successfully created on DD REDD
```

Data Set Redundancy Report

In the following report example, the Redundancy Summary identifies and groups the data sets having identical content. This provides the necessary information for either the elimination of duplicate data sets, or data set sharing of a single version, allowing the deletion of the duplicates and recovery of DASD space. In a normal environment, the empty allocated datasets appear as the first group (highest redundancies). It is not unusual for hundreds of empty data sets to be identified. Note that some of them may be valid for specific reasons.

Data Set Redundancy Report Example

```
( D A T A S E T )   R E D U N D A N C Y   R E P O R T

Red Total-Tracks Recoverable D R Volser          Tracks Tracks
Lev Alloc  Used  Alloc  Used S F Dsname----- Alloc  Used Hash-Time-Stamp Referred

002    3    3    2    2 P F SRSM5F BSCHEED.EMPTY.DSN1      1    1 2009/11/19_08:36 2009/11/19
          P F SRSM4D BSCHEED.EMPTY.DSN2      1    1 2009/11/19_08:36 2009/11/19
          P F SRSM5F BSCHEED.EMPTY.DSN3      1    1 2009/11/19_08:36 2009/11/19

002    3    3    2    2 P F SRSM77 BSCHEED.DUPE.CONTENT1    1    1 2009/11/19_08:36 2009/11/19
          P F SRSM5A BSCHEED.DUPE.CONTENT2    1    1 2009/11/19_08:36 2009/11/19
          P F SRSM73 BSCHEED.DUPE.CONTENT3    1    1 2009/11/19_08:36 2009/11/19

001    2    2    1    1 P F SRSM57 BSCHEED.IDENT.CONTENT1    1    1 2009/11/19_08:36 2009/11/19
          P F SRSM9D BSCHEED.IDENT.CONTENT2    1    1 2009/11/19_08:36 2009/11/19

HP50807I  HPSREPT  08:36:43 ** R E D U N D A N C Y   S U M M A R Y   ( D S N ) **
          3 Redundant groups
          8 Included dsns
          2 Highest redundancy level
          8 Total allocated tracks in all groups
          8 Total used tracks in all groups
          5 Recoverable tracks from all groups
```

Member Redundancy Report

The following Member level report groups members having duplicate content. This type of report is directed more towards groups of concatenated data sets for which you might want to eliminate duplicate content members, parms, procs, and so on, where only one version is desirable. Another use would be to consolidate several libraries into one.



NOTE The summary field *Highest redundancy level* is always one less than the total number of data sets in the redundant group, since at least one version is not considered redundant.

Member Redundancy Report Example

(M E M B E R) R E D U N D A N C Y R E P O R T										
Red Lev	Group-size Bytes	Recoverable Bytes	Member	Volser	Dsname-----	Member-size hash-bytes	Updat-Time-Stamp	Owner		
002	738	492	MEMBER3	SRSM77	BSCHEED.DUPE.CONTENT1	246	2009/11/18_06:10	USER01		
			MEMBER3	SRSM5A	BSCHEED.DUPE.CONTENT2	246	2009/11/18_06:10	USER01		
			MEMBER3	SRSM73	BSCHEED.DUPE.CONTENT3	246	2009/11/18_06:10	USER01		
002	492	328	MEMBER2	SRSM77	BSCHEED.DUPE.CONTENT1	164	2009/11/18_06:10	USER01		
			MEMBER2	SRSM5A	BSCHEED.DUPE.CONTENT2	164	2009/11/18_06:10	USER01		
			MEMBER2	SRSM73	BSCHEED.DUPE.CONTENT3	164	2009/11/18_06:10	USER01		
002	246	164	MEMBER1	SRSM77	BSCHEED.DUPE.CONTENT1	82	2009/11/18_06:10	USER01		
			MEMBER1	SRSM5A	BSCHEED.DUPE.CONTENT2	82	2009/11/18_06:10	USER01		
			MEMBER1	SRSM73	BSCHEED.DUPE.CONTENT3	82	2009/11/18_06:10	USER01		
001	164	82	MEMX	SRSM57	BSCHEED.IDENT.CONTENT1	82	2009/11/18_08:31	USER01		
			MEMX	SRSM9D	BSCHEED.IDENT.CONTENT2	82	2009/11/18_08:31	USER01		
HPS0808I HPSREPRT 08:36:43 ** R E D U N D A N C Y S U M M A R Y (MEM) *										
4 Redundant groups										
11 Included members										
2 Highest redundancy level										
1,640 Total bytes in all groups										
1,066 Recoverable bytes from all groups										

Chapter 7

Snapshot Facility

This chapter describes the Snapshot facility for comparing data set names and attributes in large environments.

Snapshot compares pools of volumes at two data centers. A typical use would be after a data center migration. Missing data sets, and data sets with differing last reference dates are identified.

Snapshot can assist in data center consolidation, data center migration, and the verification of the presence of data sets at a business recovery site. Additionally, exceptions in the DCB attributes and directory blocks of data sets can alert you to potential out-of-space and out-of-directory-space conditions.

Snapshot Process	238
Compare Snapfiles - Mismatched Data Sets	239
Compare Snap Files - Last Referenced Date	241

Snapshot Process

The Snapshot facility collects VTOC and VVDS information from a volume. It records this information, along with DCB attributes, EXTENTS allocated, and SPACE and DIRECTORY information, in a physical sequential file. This file is called a snap file.

The SNAPSHOT command, without the optional PDS directory processing, operates at approximately 50 DASD volumes per minute (wall clock time). The optional directory processing provides additional information such as directory blocks allocated and used, and the number of members in a PDS.

Once two snap files are created, a variety of reports can be produced to identify exceptional conditions.

Following is example JCL to snapshot a pool of DASD volumes.

```
//JOB CARD
//*-----
//*  SNAPSHOT EXAMPLE WITH INCLVOL
//*  INCLVOL=(SYS/,WRK/)                VOLUME PATTERNS
//*  INCLVOL=(SCP/,SPOOL*)             VOLUME PATTERNS
//*  EXCLVOL=(SYSRES,WRKTMP)           EXPLICIT VOLUMES
//*  REPORT=NO                          SHORT REPORT
//*-----
//STEP1  EXEC  PGM=HPSSNAPV,REGION=4096K
//STEPLIB DD  DSN=somnode.CMNSSM.LOAD,DISP=SHR
//        DD  DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SNAPFILE DD  DISP=SHR,DSN=somnode.SNAPFIL1  OUTPUT FILE
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//HPSIN   DD   *
VOLSER=/                                ALL VOLUMES      <==
REPORT=YES                              DETAILED REPORTING <==
```

The following table describes some of the commands and parameters in the example JCL:

Command	Description
SNAPFILE	Identifies the output file of the SNAPSHOT process. This file is unsorted.
HPSIN	Identifies the commands and parameters passed to Snapshot.
EXCLVOL	Identifies explicit volumes and patterns of volumes that are to be excluded.
INCLVOL	Identifies explicit volumes and patterns of volumes that are to be included.
REPORT	Specifies the report type., where: YES – detailed reporting. NO – (default)
VOLSER	Specifies a pattern of volumes. VOLSER=/ processes all volumes.

The following example shows a snapshot sort job.

```
//JOB CARD
//*****
//* Sorts the two snapfiles by DSN in ascending order ***
//*****
//STEPSRT1 EXEC PGM=SORT,REGION=6M,COND=(4,LT)
//SORTIN DD DISP=SHR,DSN=somnode.SNAPFIL1
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTWK02 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTWK03 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTWK04 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTOUT DD DISP=SHR,DSN=somnode.SNAPFIL1
//SYSIN DD *
        SORT FIELDS=(09,46,CH,A)
        RECORD TYPE=FB,LENGTH=68
//*-----
//STEPSRT2 EXEC PGM=SORT,REGION=6M,COND=(4,LT)
//SORTIN DD DISP=SHR,DSN=somnode.SNAPFIL2
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTWK02 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTWK03 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTWK04 DD UNIT=SYSALLDA,SPACE=(CYL,5)
//SORTOUT DD DISP=SHR,DSN=somnode.SNAPFIL2
//SYSIN DD *
        SORT FIELDS=(09,46,CH,A)
        RECORD TYPE=FB,LENGTH=68
```

Compare Snapfiles - Mismatched Data Sets

The following example JCL produces a snapshot report of mismatched data set.

```
//JOB CARD
//REPORT EXEC PGM=HPSSNAPR,REGION=4096K,
// PARM='MISMATCHES' SHOW ONLY MISMATCHES
//* PARM='MATCHES' SHOW ONLY MATCHES
//* PARM='ALL' SHOW ALL
//* PARM='MISMATCH=SYSUT1' SHOW SYSUT1 MISMATCHES
//* PARM='MISMATCH=SYSUT2' SHOW SYSUT2 MISMATCHES
//* PARM='MATCHES MISMATCH=SYSUT1'
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR
// DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSUT1 DD DISP=SHR,DSN=h1q.SNAPFIL1
//SYSUT2 DD DISP=SHR,DSN=h1q.SNAPFIL2
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//HPSIN DD *
INCLVOL=/
INCLDSN=(DETECT/,USER14/)
```

The following table describes some of the commands and parameters in the example:

Command	Description
PARM	Specifies the type of report, where: MISMATCHES – show only mismatched data sets. MATCHES – show only matching data sets. ALL – show all data sets. MISMATCH=SYSUTn – show the mismatched data sets from the snap file pointed to by SYSUTn MATCHES MISMATCH=SYSUTn – show the matching data sets and the mismatched data sets from the snap file pointed to by SYSUTn
SYSUT1	Identifies the first of the snap files.
SYSUT2	Identifies the second of the snap files.
HPSIN	Identifies explicit volumes and patterns of volumes.
INCLVOL	Identifies explicit volumes and patterns of volumes that are to be included.
INCLDSN	Identifies explicit data sets or patterns of data sets to include in the report.

Following is a sample of a mismatched data set report.

Exec-Param="MISMATCHES"

Report parameters: SYSUT1 Mismatches = Yes
 SYSUT2 Mismatches = Yes
 Matches = No

```
HPSIN          1  1  2  2  3  3  4  4  5  5  6  6  7
NR   ....5....0....5....0....5....0....5....0....5....0....5....0....5....0....
```

```
-- S Y S U T 2 --      -- S Y S U T 1 -- Entry
Volser Blksi Lrecl  Volser Blksi Lrecl  Typ  Data Set/Cluster/Component Name
HPS002 3200  80 | ..... | N CMNSSM.PANELS1
HPS002 6000  80 | ..... | N CMNSSM.PANELS2
HPS002 6000  80 | ..... | N CMNSSM.PANELS3
HPS002 6000  80 | ..... | N CMNSSM.PANELS4
..... | HPS002 4096 4096 | N CMNSSM.TABLE1
..... | HPS002 4096  0 | N CMNSSM.TABLE1.FGP
..... | HPS002 4096  0 | N CMNSSM.TABLE2.FGP
HPS002 23472  0 | ..... | N CMNSSM.KSDS1.FGP
HPS002 23472  0 | ..... | N CMNSSM.KSDS2.FGP
..... | HPS002 4096 4096 | N CMNSSM.TABLE2
..... | HPS002 23472  0 | N USER14.DEV.LINKLIB
..... | HPS002 3200  80 | N USER14.OUTFILE
```

```
HPSS3303I  HPSSNAPR  S U M M A R Y :
                SYSUT1      SYSUT2
                533          533 DSNs read
                533          533 DSNs selected
                 6            6 DSNs not matched
                527          527 DSNs matches
```


Compare Snap Files - Last Referenced Date

This job does a comparison of data sets from two sites, and reports only the mismatch dates. It can be used over time to report the data sets that have been referenced since the snapshot. There are no execute parameters for this program.

Following is sample JCL for the report of mismatched last referenced dates.

```
//JOB CARD
//REPORT EXEC PGM=HPSSNAPD,REGION=4096K
//STEPLIB DD DSN=somnode.CMNSSM.LOAD,DISP=SHR
// DD DSN=somnode.SERCOMC.LOAD,DISP=SHR
//SYSUT1 DD DISP=SHR,DSN=USER14.SNAPFILE.SORT1
//SYSUT2 DD DISP=SHR,DSN=USER14.SNAPFILE.SORT2
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//HPSIN DD *
        INCLDVOL=/
```

The following table describes some of the commands and parameters in the example:

Command	Description
SYSUT1	Identifies the first of the snap files.
SYSUT2	Identifies the second of the snap files.
HPSIN	Identifies explicit volumes and patterns of volumes.
INCLVOL	Identifies explicit volumes and patterns of volumes that are to be included.
INCLDSN	Identifies explicit data sets or patterns of data sets to include in the report.

Following is a sample mismatched last referenced date report.

```
HPSIN      1  1  2  2  3  3  4  4  5  5  6  6  7  7
NR   ....5....0....5....0....5....0....5....0....5....0....5....0....5....0....5
```

```
001 "      INCLDVOL=/
```

```
EOF
```

```
-- S Y S U T 2 --      -- S Y S U T 1 -- Entry
```

```
Volser Referenced | Volser Referenced | Typ Data Set/Cluster/Component Name
HPS001 2001/01/09 | HPS002 2000/01/08 | N CMNSSM.PANELS1
HPS001 2000/08/06 | HPS002 ..... | N CMNSSM.PANELS2
HPS001 2000/02/09 | HPS002 2000/07/28 | N CMNSSM.PANELS3
HPS001 2000/04/05 | HPS002 2000/07/19 | N CMNSSM.PANELS4
HPS001 2000/06/11 | HPS002 ..... | N CMNSSM.TABLE1
HPS001 2000/07/26 | HPS002 2000/07/27 | N CMNSSM.TABLE1.FGP
HPS001 2000/05/06 | HPS002 2000/02/18 | N CMNSSM.TABLE2.FGP
HPS001 2000/05/06 | HPS002 2000/02/18 | N CMNSSM.OBJECT.MASTER.BACKUP
HPS001 2000/02/18 | HPS002 2000/05/06 | N CMNSSM.SITE.MASTER.BACKUP
HPS001 2000/06/28 | HPS002 ..... | N CMNSSM.TEST.PDS.VBS
HPS001 2000/07/14 | HPS002 2000/07/27 | N CMNSSM.BASKET
HPS001 2000/12/16 | HPS002 2000/09/04 | N CMNSSM.VER810.INSTALL
HPS001 2000/10/06 | HPS002 2000/09/04 | N CMNSSM.VER810.SAMPLES
```

```
HPS3332I  HPSSNAPR  S U M M A R Y :
                SYSUT1      SYSUT2
                588          327 DSNs read
                588          327 DSNs selected
                574          313 DSNs not matched
                 14           14 DSNs matches
                 13           13 Reference Date mismatches
```

```
HPS0000I  HPSSNAPR  Job Started: 1999/08/11_13:48:29
                Job Ended  : 1999/08/11_13:48:30
                Warning cnt: 0
                Severe msgs: 0
```

Appendix A

Wildcards, Symbolics, and DCB Attributes

This appendix provides:

- DCB attributes for files used in the Synchronization Component.
- Information about using system symbolics in parameters.
- Details and examples for each of the wildcards used in the parameters in the Online and Batch Synchronization facilities.



NOTE For information on using wildcards with HFS files, see [Using Pattern Names for HFS files on page 201](#).

DCB Attributes

The following table provides the DCB attributes for files used in the Synchronization Component.

Data Set Type	DCB Attribute
Fingerprint	LRECL=512 , BLKSIZE=4096 , DSORG=PS , RECFM=FB
Change Basket	LRECL=0 , BLKSIZE=4096 , DSORG=PS , RECFM=U
Reverse Change Basket	LRECL=0 , BLKSIZE=4096 , DSORG=PS , RECFM=U
VSAM Fingerprint	LRECL=0 , BLKSIZE=4096 , DSORG=PS , RECFM=U
VSAM Change Basket	LRECL=4096 , BLKSIZE=4096 , DSORG=PS , RECFM=F
VSAM Reverse Change Basket	LRECL=4096 , BLKSIZE=4096 , DSORG=PS , RECFM=F
HFS Fingerprint	DSORG=PS , RECFM=FB LRECL= 80 or greater , but < or = 32K BLKSIZE= any multiple of LRECL up to 32K
HFS Change Basket	DSORG=PS , RECFM=FB LRECL= 80 or greater , but < or = 32K BLKSIZE= any multiple of LRECL up to 32K
HFS Reverse Change Basket	DSORG=PS , RECFM=FB LRECL= 80 or greater , but < or = 32K BLKSIZE= any multiple of LRECL up to 32K
Scratch List	LRECL=80 , BLKSIZE=4000 , DSORG=PS , RECFM=FB

Data Set Type	DCB Attribute
Token	LRECL=1000 , BLKSIZE=6000 , DSORG=PS , RECFM=VB
DETECT DD SYNCTRAC DD	LRECL=133 , BLKSIZE=6650 , DSORG=PS , RECFM=FBA
"userid.SSM.LOG"	LRECL=133 , BLKSIZE=6650 , DSORG=PS , RECFM=FBA

System Symbolics

System symbolics can be used in a number of parameters in the Batch Synchronization facility.

NOTE System symbolics are not supported in the ISPF panels.

Using Symbolics in Command Parameters

The following batch command parameters support system symbolic values:

VOLSER
INCLVOL
EXCLVOL
NEWDSNTOVOL
DSN
INCLDSN
EXCLDSN
CLUSTER
INCLCLUS
EXCLCLUS
GENERIC

Using Symbolics in HPSPARMS Parameters

The following HPSPARMS parameters support system symbolic values within the DSN specification:

SITEMASTER
OBJECTMASTER
DELTAMASTER

Wildcards

Wildcards, or patterns, can be used in a number of parameters in the Online and Batch Synchronization facilities. This section provides details and examples for each of the wildcards.

Data Set, Object, and Cluster Name Wildcards

The following special characters can be used to specify a pattern of data sets, objects, or clusters in the INCLDSN, EXCLDSN, INCLOBJ, EXCLOBJ, INCLCLUS, or EXCLCLUS parameters:

Wildcard	Definition
*	Node specification. If it is the only character in a node, then any character string in the node is accepted. If it is the last character in an otherwise specified string, then any node that begins with the specified string is selected, regardless of any final characters.
?	Character specification. Any character in the position specified by the ? satisfies the selection requirements.
/	Search terminator. Any characters beyond this character are accepted. A pattern without the / implies that the pattern characters correspond exactly to the data set name characters. For example, 2ABC selects all data sets in which the second node is exactly ABC, whereas 2ABC/ selects all data sets in which the second node begins with ABC.
!	String search delimiter. Any string in a data set name that matches the string delimited with ! meets the selection criteria. For example, !string!.
n	Node number indicator. <i>n</i> takes values from one to twenty-two, numbered left to right. The last (rightmost) node in a data set name is the zero node. For example, given node1.node2.node3.node4, then node4 could also be referenced as node0.

Examples with INCLDSN

- All three-node data sets that have a first node of AAA, a second node of BBB, and a third node of <anything> are included.

```
INCLDSN=AAA.BBB.*
```

- All three-node data sets that have a first node of AAA, a second node of <anything>, and a third node of CCC are included.

```
INCLDSN=AAA.*.CCC
```

- All minimum four-node data sets that have the first three nodes of AAA.BBB.CCC are included.

INCLDSN=AAA.BBB.CCC./

- All data sets that have a first node of AAA, a second node of BBB, and a third node starting with C are included.

INCLDSN=AAA.BBB.C/

- All three-node data sets that have a first node of AAA, a second node of three-character length starting with BB, and a third node of CCC are included.

INCLDSN=AAA.BB?.CCC

Examples of INCLDSN used with the GENERIC Parameter

- Limits search results to data set names with an exact match of "ABC" on the 2nd node. Data set names with additional nodes (3 or more) will *not* be considered.

INCLDSN=(2ABC)

- Limits search results to data set names with an exact match of "ABC" on the 2nd node. Data set names with 2 or more additional nodes will be considered.

INCLDSN=(2ABC./)

- Limits search results to data set names with an exact match of "ABC" on the 2nd node. Data set names with 2 or 3 nodes only will be considered.

INCLDSN=(2ABC.*)

- Limits search results to data set names with a 2nd node starting with "ABC". Data set names with 2 or more additional nodes will be considered.

INCLDSN=(2ABC*)
INCLDSN=(2ABC/)

Examples with EXCLDSN

- All data sets (of any number of nodes) with a final (last) node of LINK are excluded.

EXCLDSN=0LINK (i.e. NODE1,NODE2,...LINK)

- All three-node data sets with a third node of LINK are excluded (node1.node2.LINK).

EXCLDSN=3LINK

- All data sets with the third node starting with LINK are excluded (node1.node2.LINKLIB will be excluded).

EXCLDSN=3LINK/

- In a search, all four-node data sets with a fourth node of <anything> are excluded (node1.node2.node3.anything).

EXCLDSN=4*

Examples with INCLCLUS

- All three-node clusters that have a first node of AAA, a second node of BBB, and a third node of <anything> are included.

```
INCLCLUS=AAA.BBB.*
```

- All three-node clusters that have a first node of AAA, a second node of <anything>, and a third node of CCC are included.

```
INCLCLUS=AAA.*.CCC
```

- All minimum four-node clusters that have the first three nodes of AAA.BBB.CCC are included.

```
INCLCLUS=AAA.BBB.CCC./
```

- All clusters that have a first node of AAA, a second node of BBB, and a third node starting with C are included.

```
INCLCLUS=AAA.BBB.C/
```

- All three-node clusters that have a first node of AAA, a second node of three-character length starting with BB, and a third node of CCC are included.

```
INCLCLUS=AAA.BB?.CCC
```

Examples with EXCLCLUS

- All clusters (of any number of nodes) with a final (last) node of MASTER are excluded. The following example is equivalent to NODE1,NODE2,...MASTER.

```
EXCLCLUS=0MASTER
```

- All three-node clusters with a third node of MASTER are excluded. The following example is equivalent to node1.node2.MASTER.

```
EXCLCLUS=3MASTER
```

- All data sets with the third node starting with MASTER are excluded. In the following example, node1.node2.MASTER will be excluded.

```
EXCLCLUS=3MASTER/
```

- In a search, all four-node clusters are excluded (node1.node2.node3.<anything>).

```
EXCLCLUS=4*
```

Examples with INCLOBJ

- All three-node objects that have a first node of AAA, a second node of BBB, and a third node of <anything> are included.

```
INCLOBJ=AAA.BBB.*
```

- All three-node objects that have a first node of AAA, a second node of <anything>, and a third node of CCC are included.

```
INCLOBJ=AAA.*.CCC
```

- All minimum four-node objects that have the first three nodes of AAA.BBB.CCC are included.

INCLOBJ=AAA.BBB.CCC./

- All objects that have a first node of AAA, a second node of BBB, and a third node starting with C are included.

INCLOBJ=AAA.BBB.C/

- All three-node objects that have a first node of AAA, a second three-character node starting with BB, and a third node of CCC are included.

INCLOBJ=AAA.BB?.CCC

Examples with EXCLOBJ

- All objects (of any number of nodes) with a final (last) node of LINK are excluded.

EXCLOBJ=@LINK

- All three-node objects with a third node of LINK (node1.node2.LINK) are excluded.

EXCLOBJ=3LINK

- All objects with the third node starting with LINK (node1.node2.LINKLIB) are excluded.

EXCLOBJ=3LINK/

- In a search, all four-node objects with a fourth node of <anything> (node1.node2.node3.<anything>) are excluded.

EXCLOBJ=4*

Catalog Search

The Catalog Search Interface (CSI) of DFSMS, which is the same search facility used in ISPF Option 3.4, is used in the DSN sub-parameter of the GENERIC parameter and several of the online panels.

NOTE The panel fields that use the CSI are indicated with text that says the pattern name is similar to ISPF 3.4. The commands that support pattern names in the DSN sub-parameter of the GENERIC parameter are noted in those command/parameter descriptions.

SSM supports all of the patterns that ISPF 3.4 supports; however, SSM does not use the option "Include Additional Qualifiers" because it can cause large numbers of catalog entries to be retrieved, which then have to be filtered by the user. If you need additional qualifiers, use * or ** to indicate which nodes you want beyond the nodes explicitly defined in the pattern.

The following table describes how to specify pattern names using wildcards with the CSI. For more information, refer to the IBM DFSMS documentation.

Character	Description
*	A single asterisk by itself indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
**	A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a period or a blank.
%	A single percent sign by itself indicates that exactly one alphanumeric or national character can occupy that position.

Member Name Wildcards

The following special characters can be used to specify a pattern of members:

Wildcard	Definition
/,*	Search terminator. Any characters beyond this character are accepted. A pattern without the / implies that the pattern characters correspond exactly to the data set name characters. For example, ABC* indicates all members starting with ABC. ABC/ is identical to ABC*. Any character in the position specified by the wildcard character satisfies the inclusion or exclusion requirements.
?	Character specification. Any character in the position specified by the ? satisfies the selection requirements. For example, the pattern UT?A refers to those members where the first two characters are UT, the last character is A, and the third character is any legitimate character.
!	String search delimiter. Any string in a data set name that matches !<string>! meets the selection criteria. For example, !HPS! selects all members where the string !HPS! is found anywhere in the name.
X	Hex indicator. X notation allows for the specification of a hex character string in the member name. For example, X'C1C2??E5' indicates all members in which the first two characters are C1C2, the third character could be anything and the last character is E5.

Volume Name Wildcards

The following special characters can be used to specify a pattern of volumes:

Wildcard	Definition
/,*	Search terminator notation. Any character beyond this character in a volume specification is accepted. For example, in the pattern DEV*, all volumes that have DEV as the first three characters will be selected. The pattern DEV/ is identical to DEV*.
?	Wildcard character specification. Any character in the position specified by the ? character satisfies the selection requirements.

Appendix B

Transfer Change Baskets

This appendix describes the steps to package and apply a change basket through email.

Convert and Download Basket

These steps describe how to convert a change basket to a transmittable format suitable for downloading to the desktop.

Allocate an XMIT data set

On the mainframe where the basket resides, allocate an XMIT data set:

```
DSN:your_userid.XMIT  
DCB:DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=6000
```

Unload your Change Basket data set to a transmittable XMIT format

Use the TSO Command:

```
XMIT userid.company DSN(BASKET) OUTDSN(XMIT)
```



NOTE You may receive several screens of informational messages depending on the size of your Change Basket. Press Enter to proceed until your Change Basket is loaded to the XMIT data set.

Define a New Transfer Type Called BIN80

(This example uses IBM Personal Communications.) From the menu bar, select Transfer > Setup > Define Transfer Types. On the pop-up window, specify a new transfer type called: BIN80 (under Transfer Type Names). Make sure the three fields (ascii, clrf, and append) are not checked.

The only required fields to be specified are:

Record Format: Fixed

LRECL: 80

Press the ADD button.



NOTE Use this new transfer file type to download the 'your_userid.XMIT' file from MVS to PC.

Download the 'your_userid.XMIT' file to PC

Download the 'your_userid.XMIT' file from MVS to your desktop, suitable for electronic distribution or as an e-mail attachment. Specify a file name as necessary. For example:

BASKET.BIN

Perform steps similar to the following:

While in ISPF TSO Option 6 (or in READY mode), select the following from the IBM Personal Communications menu bar: Transfer > Receive File From Host. In the pop-up window, specify the following options:

Host File:xmit(Your transmittable formatted data set)

PC File:d:\temp\xmit(The XMIT file on your desktop)

Transfer Type:BIN80(The newly defined transfer type. You must set it.)

Upload and Apply a Basket to a Target MVS System

The following steps describe how to upload and apply a Change Basket to a Target MVS system.

Copy Change Basket File to PC

If you received the Change Basket file as an e-mail attachment, copy the file to a folder on your desktop.



NOTE The Change Basket file is an XMIT file. Do not let the naming convention mislead you.

Allocate a XMIT data set on the Target MVS

DSN: your_userid.XMIT

DCB: DSORG+PS,RECFM=FB,LRECL=80,BLKSIZE=6000

Define or Add a New Transfer Type: BIN80

See "[Define a New Transfer Type Called BIN80](#)" on page 251.

Upload the Change Basket PC file to MVS

While in ISPF Option 6 (TSO), perform a "Send File to Host" operation from the menu bar of your emulator software using the newly created BIN80 transfer type. Proceed to make the following selections, then send:

PC File: BASKET.BIN (Change Basket file)

Host File: XMIT (Prefix your TSO UserID)

Transfer Type: BIN80 (New transfer type)

Issue the TSO RECEIVE command

While in ISPF Option 6 on MVS, issue the TSO RECEIVE command to automatically rebuild the Change Basket from the 'your_userid.XMIT' data set with the proper attributes:

```
RECEIVE INDS (XMIT)
```



NOTE You may receive several screens of informational messages (depending on the size of your Change Basket.) Press Enter until your change Change Basket data set is created. 'your_userid.BASKET'

Use the Change Basket Data Set as Input

Use the Change Basket data set 'your_userid.BASKET' as input to your import process.

Glossary

Alternate Group Administrator	A person assigned by an existing Group Administrator to perform the primary tasks of that Group Administrator in his/her absence.
BICT (Batch Interval Change Tracking)	A batch program (HPSGLOBE) that performs the change-capture process for all tracked Objects at specified intervals. It should be scheduled to run minimally, once every 24 hours. This is an alternative to RTCT (Real Time Change Tracking).
Change Basket (keyed, positional, or text)	<p>A single physical sequential data set that contains the changes between two keyed, positional, or text flat files. This file contains the changed data records. Add, update, and delete information is present, but is not currently used for synchronization by ChangeMan SSM.</p> <p>Definitions for file types:</p> <p>Keyed file: A physical sequential or direct access file that has a field in each record that can be used as a key. This key must be unique for each record in the file.</p> <p>Positional file: A physical sequential or direct access file containing records that are indexed by their physical position. For example, the first record is considered record 1, and the second is record 2.</p> <p>Text file: A physical sequential or direct access file with records that are in no particular order, and do not have a specific key.</p>
Change Basket (non-VSAM)	A single physical sequential data set that contains the changes to an environment over time, or differences between two environments. For example: Test/Prod, SYSRES/ALTRES, and LPAR/LPAR. This file holds the data and information required to synchronize two environments, or propagate change from one system to another, locally or remotely.
Change Basket (VSAM)	A single physical sequential data set that contains the changes/differences between two VSAM clusters. This file contains the data records and required information to synchronize two VSAM KSDS data sets, locally or remotely.
Delta Master	A ChangeMan SSM database that contains the initial base version, plus subsequent changed versions, of a PDS/PDSE member (for data sets defined to ChangeMan SSM with Delta Tracking). Backup members are stored in compressed format.
Department Administrator	A person assigned by a Site Master Administrator to manage the Object Master database.

**DSORG
(Supported
Types)**

DA indicates a direct access data set.

PO indicates a partitioned data set (PDS and PDS/E).

PO type Format-3 data sets provide support for DFSMS Binder, NPO Format-3 load modules, and stored IBM JAVA files.

PS indicates a physical sequential data set.

VS indicates a VSAM data set.

NOTE Data sets with no DSORG specified are not supported.

Extract/Export

Extract is the process that ChangeMan SSM uses to identify changes/differences, utilizing fingerprints and/or active data sets.

Export is the process that packages the changes/differences into a Change Basket data set.

Event

A change event when ChangeMan SSM detects a variance in content or status of a tracked data set or data set member. There are 23 types of events distinguished by the software.

**Fingerprint
Dataset**

A physical sequential data set containing all the data set and member tokens to identify the contents of that environment at the time the Fingerprint data set was generated. A Fingerprint data set can represent a single PDS member, or the contents of several DASD volumes.

**Fingerprint
Dataset (keyed,
positional, or
text)**

A physical sequential file that contains all the fingerprint tokens of all the records in a physical sequential file. This file represents a PS or DA file.

Definitions for file types:

Keyed file: A physical sequential or direct access file that has a field in each record that can be used as a key. This key must be unique for each record in the file.

Positional file: A physical sequential or direct access file containing records that are indexed by their physical position. For example, the first record is considered record 1, and the second is record 2.

Text file: A physical sequential or direct access file with records that are in no particular order, and do not have a specific key.

**Fingerprint
Dataset (VSAM)**

A physical sequential data set containing tokens for every record in a VSAM KSDS cluster. This file represents the contents of a VSAM cluster at the time the Fingerprint data set was generated.

**Fingerprint
Token**

A unique identity attribute for the content of a data set or data set member. The attribute is an eight-byte, hexadecimal value.

Fingerprint Token (keyed, positional, or text)	<p>A unique identity attribute for records of a keyed, positional, or text file. The attribute is a six-byte, hexadecimal value.</p> <p>Definitions for file types:</p> <p>Keyed file: A physical sequential or direct access file that has a field in each record that can be used as a key. This key must be unique for each record in the file.</p> <p>Positional file: A physical sequential or direct access file containing records that are indexed by their physical position. For example, the first record is considered record 1, and the second is record 2.</p> <p>Text file: A physical sequential or direct access file with records that are in no particular order, and do not have a specific key.</p>
Fingerprint Token (VSAM)	<p>A unique identity attribute for records of VSAM clusters. The attribute is a six-byte, hexadecimal value.</p>
Group Administrator	<p>A person assigned by a Department Administrator to manage one or more Object Groups.</p>
HFS Change Basket	<p>A single physical sequential data set that contains the changes/differences between two HFS files. This file contains the data records and required information to synchronize two HFS files, locally or remotely.</p>
HFS Fingerprint Dataset	<p>A physical sequential data set containing tokens for every record in an HFS file. This file represents the contents of an HFS file at the time the Fingerprint data set was generated.</p>
HFS Fingerprint Token	<p>A unique identity attribute for records of HFS files. The attribute is a six-byte, hexadecimal value.</p>
HFS Reverse Change Basket	<p>A single physical sequential data set created during an HFS Import operation. It contains the information and data to undo the effects of updates, as a result of an HFS Import operation. This file has an identical structure to an HFS Change Basket.</p>
Import	<p>Import is the process that ChangeMan SSM utilizes to apply the changes contained in a Change Basket data set, to target environments. During Import processing there are several parameters that can be specified to customize the apply process, as well as a SIMULATE feature, and the creation of a Reverse Change Basket.</p>
MLS (Member Level Security)	<p>This facility allows for the protection of PDS/PDSE members being updated. The software provides the capability for MLS Administrators to identify who is allowed to update specific members/member patterns for specified data sets, by specifying either TSO User IDs or a RACF profile name.</p> <p>If an attempt is made to update a protected member, ChangeMan SSM will either save the changes to an internally-generated member name or cancel the updates, depending on what option you selected for the RTORENAME parameter.</p>
MRT (Member Reference Tracking)	<p>This facility allows you to detect and report references to members of PDSs. Read and update access is tracked on existing members that you define to the MRT system, and where standard I/O routines are used.</p> <p>You can also define members to be excluded from MRT. This allows you to filter out references to high-activity members at the beginning of MRT processing, which greatly enhances performance.</p>

Non-user	A person unknown, or not defined by a Group Administrator to any Object Group, but allowed access to the Synchronization and Detection functions of ChangeMan SSM.
Notifications	A ChangeMan SSM facility that provides change notifications to specified users about changes to tracked data sets and PDS members. There are both TSO User ID and E-mail notification facilities that can be used separately or together.
Object	A data set defined to the Change Tracking component of the tracking software. An object of the tracking function.
Object (Delta Tracked)	<p>A PDS or PDS/E data set defined to the Change Tracking component, where Delta Tracking was specified during the initial define of the object. This specification creates an initial backup of all members into the Delta Master database. Subsequently, only changed members, and additions are backed up in compressed format.</p> <p>NOTE Sequential data sets cannot be delta-tracked; if they are defined as DELTA=Y, the object is defaulted to DELTA=N.</p>
Object Group	A group of conceptually related data sets (Objects) that are associated for tracking, change detection, and reporting. There are no restrictions on mixing data set types in a group.
Object Master	A ChangeMan SSM database connected to the Site Master database, which contains definitions of all the Object Groups, their associated Group Administrators, and users defined to the Change Tracking environment.
RTO (Real Time Option)	<p>RTO was the name used for Real Time Change Tracking (RTCT) for ChangeMan SSM Version 8.1 and earlier.</p> <p>See RTCT.</p>
RTCT (Real Time Change Tracking)	The task of capturing data set or data set member information that allows the software to detect and identify change in real time.
Recall	A ChangeMan SSM facility to allow the recall of a backup version of selected (Delta Tracked) PDS or PDS/E members to a compatible data set. Expansion is performed automatically.
Recover	A ChangeMan SSM facility to allow the recovery of a Delta Tracked PDS or PDS/E data set, as it existed in some previous point in time, to a compatible data set. Expansion is performed automatically.
Redundancy Detection	A ChangeMan SSM facility that identifies duplicate data sets and PDS/PDSE members by content. A comprehensive reporting facility allows for the identification of duplicates to facilitate the deletion of those having duplicate content. This allows for recovery of DASD space and enhanced integrity (only one copy). This facility also assists in library cleanup and consolidation.
Restore	A ChangeMan SSM facility to allow the restore of a backup version of selected (Delta Tracked) PDS or PDS/E members to the original data set. Expansion is performed automatically.
Reverse Change basket	A single physical sequential data set, created during an Import operation, which contains the information and data to undo (entirely or partially) the effects of updates as a result of an Import operation. This file has an identical structure to a Change Basket.

Reverse Change Basket (VSAM)	A single physical sequential data set created during a VSAM Import operation. It contains the information and data to undo the effects of updates, as a result of a VSAM Import operation. This file has an identical structure to a VSAM Change Basket.
Site Master	The main ChangeMan SSM VSAM database, which contains information for all tracked data sets and their detected events.
Site Master Administrator	A person assigned (during installation) to manage the Site Master and define departments.
SMS	System Managed Storage. Enhanced data management software for MVS mainframes from IBM. Provides functions such as automatically allocating data, which prevents most out-of-space errors when disk volumes become full.
Snapshot	<p>A ChangeMan SSM facility that interrogates the VTOC and VVDS to identify data sets existing on one or several volumes, and capture information about these data sets and their attributes. This information is stored in a SNAPSHOT data set, which can be used to compare to another SNAPSHOT data set generated at a different location. This comparison can verify that all data sets are available, and check for differences in any data set attributes or space allocations.</p> <p>The SNAPSHOT function is especially useful as a pre-step to performing DR testing, to verify that all the production data sets are available. In addition, it allows pre-screening to eliminate potential x37 type abends, where a data set was expanded at the source site, but not at the target, when applying maintenance.</p>
Tracking	The function of monitoring a data set or data set member for events in real time (RTCT), or over a period of time (BICT).
User	A person defined to the Change Tracking component as a User in one or more groups by a Group Administrator, and allowed controlled access to its functions.

Index

Symbols

! 249
? 249, 250
* 249, 250
/ 249

A

ACB parameter
 MASK command 175
accessing
 ChangeMan SSM 18
 Online Synchronization Facility 18
 Synchronization Menu 21
Actual Differences Report 61
Adobe Acrobat 11
advanced operations 70
AFTER subparameter
 STRING parameter 182
ALIAS subparameter
 BASE parameter 173
ALL parameter
 EXTRACT command 155
ALL/DELETE/UPDATE/ADD parameters
 HFSEXTRACT command 206
allocating
 XMIT data set 251
 XMIT data set on MVS 252
application control block masking 103
APPLY parameter
 HFSIMPORT command 207
 IMPORT command 165
applying
 change baskets 43
 change baskets to MVS 252
area masking 105
AREA parameter
 MASK command 175, 177
AREA subparameters
 COL 177
 LEN 177
AREA to be masked parameter 24
AREA to search parameter 24
ASSEMBLE subparameter
 LANGUAGE parameter 179
AUTORECALL command 119

B

BASE command 120, 121
base environment
 establishing 66
BASE fingerprints
 comparing 92
base fingerprints 15
BASE parameter
 LIST command 172
BASEDSN subparameter 143, 145
BASKET
 compressing 212
 transferring 212
BASKETDD parameter
 HFSEXPORtcommand 205
 HFSIMPORT command 207
 IMPORT command 165
 VIEWBASKET command 199
BASKETDSN parameter
 HFSEXPORtcommand 205
 HFSIMPORT command 207
 IMPORT command 165
 VIEWBASKET command 199
baskets
 creating 83, 93
 viewing 42, 97
batch 21
Batch Change Detection Facility 99
batch commands
 format of 114
BEFORE subparameter
 STRING parameter 182
bytes before/after
 source and text masking parameters 24

C

CASE parameter
 MASK command 175, 178
cataloged data sets
 comparing 56
CATLG parameter
 IMPORT command 165
change baskets 15, 36, 109, 110
 applying 43
 converting 251
 data set name 37

- downloading 251
 - error conditions 110
 - importing 43, 86, 95
 - optional specification 68
 - reverse 15
 - transferring 110
- Change Detection
- batch 99
- ChangeMan SSM 14
- accessing 18
 - exiting 18
- changes
- detecting 48
 - in packages 49
- CLOCK command 134
- closing
- ChangeMan SSM 18
 - Online Synchronization Facility 18
- COBOL subparameter
- LANGUAGE parameter 179
- COL subparameter
- AREA parameter 177
- commands
- AUTORECALL 119
 - BASE 120
 - CLOCK 134
 - COMPARE 135
 - comparing 100
 - DASDRED 148
 - ENQUEUE (ENQ) 151
 - EXPORT 151
 - EXTENDED SYNC 153
 - EXTRACT 154
 - HFS 201
 - HFSBASE 202
 - HFSCOMPARE 204
 - HFSEXPORT 205
 - HFSEXTRACT 206
 - HFSIMPORT 207
 - HFSMODEL 208
 - HFSTARGET 209
 - IDRLOAD 164
 - IMPORT 164
 - LIST 172
 - MASK 174
 - MODEL command 183
 - NOECHO command 187
 - ONLINE command 188
 - ORPHANCHECK command 188
 - REFTARG 189
 - reserved DDs 118
 - RESTART command 190
 - SETPRINT 191
 - SIMULATE 196
 - SMSMAP 196
 - SNAPSHOT 238
 - stacking 117
 - structure 114
 - synchronization 109, 119
 - TARGET command 198
 - TRACE 198
 - VIEWBASKET 199
 - VIEWFGP 198
 - VOLMAP 200
- COMMENT parameter
- HFSEXPORT command 205
- comments
- in HPSIN DD 118
 - comments in ChangeMan SSM commands 119
- COMPARE command 135
- COMPARE parameter
- COMPARE command 135
- Comparex 19, 55
- generate reports 61
 - print reports 63
 - view reports 62
 - viewing report 62
- COMPAREX subparameter
- FILTER parameter 142
 - MEMTOMEM parameter 145
- Comparex subparameter 138
- comparing
- data set groups 73
 - data sets 71
 - environments 55, 71, 183, 208
 - fingerprint data sets 83
 - fingerprints 92
 - groups 73
 - member-to-member 136
 - PDSs 64
 - results 61
 - snap files 239, 241
 - two environments (batch mode) 100
 - using MASK command 102
 - volumes 79
- compile date/time/CSECT masking 108
- COMPOSITE parameter 137
- compress utility 213
- compressing
- FGP or BASKET 212
- concepts, ChangeMan SSM 13
- converting
- change baskets 251
 - VSAM basket utility 220
- creating
- baskets 83, 93
 - fingerprint data set 28
 - fingerprints (online batch function) 80
 - fingerprints (online function) 28
- CSECT parameter
- MASK command 175
- current environment

specifying 32

D

- DASDRED command 148
- data set
 - comparing 71
 - comparing (JCL) 73
 - fingerprinting 28
 - recalling fingerprint 31
 - verify synchronization 73
 - wildcards 245
- data set groups
 - masking 103
- data set names
 - entering 36
- data set organization
 - supported types 256
- data sets
 - comparing Base and Target 83
 - comparing cataloged 56
 - comparing uncataloged 56
 - excluded data set list 26
 - extracting and exporting 86
 - filtering 26
 - fingerprint name patterns 33
 - package 38
 - packaging 38
 - resolved 30
 - resolving 58, 60
- DATE parameter
 - MASK command 175
- DCB attributes 111, 243
 - Change Basket 111, 243
 - DETECT DD data set 112
 - fingerprint 111, 243
 - HFS Change Basket 243
 - HFS fingerprint 243
 - HFS Reverse Change Basket 243
 - LOG data set 112
 - Reverse Change Basket 111, 243
 - scratch list 243
 - Scratch List data set 112
 - SYNCTRAC DD data set 112
 - token 244
 - token data set 112
 - VSAM Change Basket 111, 243
 - VSAM fingerprint 111, 243
 - VSAM Reverse Change Basket 112, 243
- DD parameter
 - DASDRED command 149
 - EXPORT command 152
 - RESTART command 190
- DD subparameter
 - LOAD parameter 127
 - SAVE parameter 130
- default options 20
- defining
 - new transfer type 252
- DELETE parameter
 - IMPORT command 166
- Demo option 224
 - HFS 226
- DETAIL parameter
 - SETPRINT command 192
- DETECT DD 118
 - reserved DDs 118
- DETECT subparameter 147
 - DSNBYDSN parameter 138
 - FILTER parameter 142
 - MEMTOMEM parameter 145
- DETECT subparameters
 - ADD 147
 - ALL 147
 - COMPLETE 147
 - DELETE 147
 - EQUAL 147
 - NONE 147
 - SBS 147
 - UPDATE 147
- detecting changes 90
 - in fingerprinted data sets 48
- detecting redundancy
 - overview 221
- detection and synchronization
 - for HFS 90
- Detection and Synchronization component
 - overview 14
- differences
 - extracting and exporting 84
- Differences Report
 - printing 63
- DIR parameter
 - SETPRINT command 192
- DIRECTORY
 - exceptional conditions 237
- DISPLAYTOKEN subparameter
 - DSNBYDSN parameter 139
 - DSNTODSN parameter 141
 - FILTER parameter 143
 - MEMTOMEM parameter 145
- documents related to ChangeMan SSM 11
- downloading
 - change baskets 251
- DSN parameter 121
 - DASDRED command 149
 - EXPORT command 152
 - RESTART command 190
- DSN subparameter
 - GENERIC parameter 124

- LOAD parameter 126, 127
- SAVE parameter 130
- DSNAME subparameter
 - DSNTODSN parameter 141
- DSNASSOCIATION parameter
 - SETPRINT command 192
- DSNBYDSN parameter 138
 - COMPARE command 135
 - EXTRACT command 155, 157
- DSNMISMATCH parameter
 - SETPRINT command 192
- DSNOUTSYNC parameter
 - SETPRINT command 192
- DSNSYNC parameter
 - SETPRINT command 192
- DSNTODSN parameter
 - COMPARE command 135, 141
- DSORG
 - supported types 256

E

- Enc keyword 152, 155
- ENQUEUE (ENQ) command 151
- environment
 - comparing 55, 100
 - remote 101
 - specify base 55
 - specifying 29
 - specifying current 32
 - synchronization 21, 109
 - synchronizing 55, 65
- environment comparison
 - AUTORECALL 100
 - BASE 100
 - COMPARE 100
 - LIST 100
 - MASK 101
 - MODEL 101
 - NOECHO 101
 - ONLINE 101
 - SETPRINT 101
 - TARGET 101
 - TRACE 101
- environments
 - compare using MASK command 102
- establishing the base environment 66
- EXCEPTIONONLY parameter
 - SETPRINT command 192
- EXCLDSN parameter 193
 - DASDRED command 148
 - EXPORT command 152
 - IMPORT command 166
- EXCLDSN subparameter
 - EXTRACT DSNBYDSN parameter 157

- GENERIC parameter 124
- IMPORT command 164
- LOAD parameter 127
- SAVE parameter 130
- VOL parameter 132
- EXCLMEM subparameter
 - COMPOSITE parameter 137
 - FILTER parameter 143
- EXCLUDE parameter 203
 - HFSCOMPARE command 204
 - HFSEXPORtcommand 205
 - HFSEXTRACT command 206
 - HFSIMPORT command 208
- EXCLVOL command 238
- EXCLVOL parameter
 - DASDRED command 149
- EXCLVOL subparameter
 - GENERIC parameter 124
 - VOL parameter 132
- exiting
 - ChangeMan SSM 18
 - Online Synchronization Facility 18
- expanding
 - compressed files 216
 - compressed format 212
- EXPORT command 151
 - Enc keyword 152
 - Scr keyword 152
 - Upd keyword 152
- exporting
 - data set differences 84
 - data sets 86
- exporting (online batch function) 36, 83
- EXTENDED SYNC command 153
- EXTRACT command 154
 - DSNBYDSN parameter 157
 - Enc keyword 155
 - NBYONE parameter 160
 - Scr keyword 155
 - Upd keyword 155
- extract mode
 - specifying 85
- extracting
 - data set changes 83
 - data set differences 84
 - data sets 86
- extracting (online batch function) 71, 83, 91

F

- FGP
 - compressing 212
 - transferring 212
- FILE parameter 203
 - HFSCOMPARE command 204

- file transfer utility
 - FTP to remote host 214
- FILTER parameter 142
 - COMPARE command 136
- FILTER parameters
 - BASEDSN command 143
- filtering
 - data sets 26
- fingerprint
 - interpreting results 54
 - results 30
 - token 52
 - token data sets 111
- fingerprint data set
 - creating 28, 81
 - generating JCL 81
 - mask rules 107
 - recalling 31
 - results 30
 - safeguards 101
 - selecting 34
- fingerprint data set name
 - verifying 34
- fingerprint DSN 46
- fingerprint token
 - generating 52
- fingerprinting
 - HFS file 91
 - using MASK command 102
- fingerprints
 - base 15
 - target 15
 - viewing 45
- firewalls 101
- FORCE parameter
 - HFSIMPORT command 208
 - IMPORT command 166
- FORCE subparameter
 - EXTRACT DSNBYDSN parameter 157
- functions 21
 - Synchronization Menu 21

G

- general masking 106
- generating
 - fingerprint tokens 52
- GENERIC command 101
- GENERIC parameter 121, 123
- Glossary 255
- group comparing
 - (online batch function) 73
- group masking 103
- groups
 - comparing 73

H

- help
 - field level 18
 - online 12
- HFS
 - commands 201
 - operations 90
 - pattern names 201
- HFS Demo option 226
- HFS files
 - fingerprinting 91
 - functions 16
- HFSBASE command 202, 203
- HFSCOMPARE command 204
- HFSEXPOR command 205
- HFSEXTRACT command 206
- HFSIMPORT command 207
- HFSMODEL command 208
- HFSTARGET command 209
- HIBOUND subparameter
 - STRING parameter 182
- HPSIN command 238
- HPSIN DD 118
- HPSPLIB DD 118, 238
- HPSSCR DD
 - reserved DDs 118
- HPSSNAPR
 - HPSIN command 240, 241
 - INCLDSN command 240, 241
 - PARM command 240
 - SYSUT1 command 240, 241
 - SYSUT2 command 240, 241
- HPSUOPTN 63
- HPSUREP 19
- HPSWORK DD
 - reserved DDs 118
- HPSWORK2 DD
 - reserved DDs 118

I

- IDRLOAD 164
- IMPORT 164
- importing
 - change baskets 43, 86, 95
- importing (online batch function) 44, 71, 87, 91
- imports
 - undoing 89, 96
- IMS
 - application control block masking 103
 - message format services masking 103
- INCLDSN parameter 193
 - DASDRED command 149
 - EXPORT command 152

- IMPORT command 166
- INCLDSN subparameter
 - EXTRACT DSNBYDSN parameter 158
 - EXTRACT NBYONE parameter 161
 - GENERIC parameter 124
 - IMPORT command 164
 - LOAD parameter 126, 127
 - SAVE parameter 130
 - VOL parameter 132
- INCLMEM subparameter
 - COMPOSITE parameter 137
 - FILTER parameter 142
- INCLMEM subparameter
 - FILTER parameter 143
- INCLUDE parameter 203
 - HFSCOMPARE command 204
 - HFSEXPORtcommand 205
 - HFSEXTRACT command 206
 - HFSIMPORT command 208
- INCLVOL command 238, 240, 241
- INCLVOL parameter
 - DASDRED command 149
- INCLVOL subparameter
 - GENERIC parameter 124
 - VOL parameter 132
- INDIRECT parameter
 - IMPORT command 167
- interpreting fingerprint results 54
- interpreting results 35

J

- JCL
 - comparing
 - data set groups 73
 - creating 43
 - creating fingerprint data set 81
 - defining Mask Rules 23
 - defining Mask rules 25, 26
 - generating to import a change basket 86
 - modifying your library 19
- JCL subparameter
 - LANGUAGE parameter 180
- job cards 19

K

- keywords
 - Enc 152, 155
 - Scr 152, 155
 - Upd 152, 155

L

- language masking 106
 - source and text masking parameters 24
- LANGUAGE parameter
 - MASK command 175, 179
- LEN subparameter
 - AREA parameter 177
- LIST 172
- load library masking
 - set MASK batch mode processing 107
 - set MASK for online session 25, 26
- load module masking 102
- LOAD parameter 121, 203
- Loading a BASE environment 126
- LOBOUND subparameter
 - STRING parameter 182
- LONG
 - DSNBYDSN parameter 139

M

- managing redundancy 111
- MASK 102, 174
 - AREA parameter 177
 - CASE parameter 178
 - LANGUAGE parameter 179
 - SHOW parameter 180
 - SQUEEZE parameter 181
 - STRING parameter 181
- mask
 - comparing environments 102
 - data set filtering 26
 - excluded data set list 26
 - JCL 23, 25, 26
 - rules
 - defining 23, 25, 26
 - viewing 26
- MASK rules
 - defining 21
- masking
 - application control block 103
 - area 105
 - data set groups 103
 - general 106
 - group 107
 - language 106
 - load libraries 107
 - load modules 102
 - message format services 103
 - rules 107
 - search strings 104
 - source data sets 107
 - squeeze 106
 - squeezing 106

masking facility 102
 masking rules
 defining 22
 for load libraries 25
 member
 wildcards 249
 MEMBER parameter
 IMPORT 167
 member reference tracking 224
 MEMBER subparameter
 EXTRACT DSNBYDSN parameter 158
 MEMBERS subparameter 161
 member-to-member compare 136
 MEMTOMEM parameter 144, 145
 COMPARE command 136
 MEMTOMEM subparameters
 DETECT subparamter 145
 message format services masking 103
 MFS parameter
 MASK command 175
 MISMATCH parameter
 SETPRINT command 193
 MIXED or UPPER case
 source and text masking parameters 24
 MIXED subparameter
 CASE parameter 178
 MODEL 183
 MODEL parameter
 HFSCOMPARE command 204
 HFSEXTRACT command 206
 HFSIMPORT command 207
 IMPORT command 167
 MODEL subparameter
 DSNBYDSN parameter 139
 DSNTODSN parameter 141
 EXTRACT DSNBYDSN parameter 158
 modes
 APPLY 87, 90
 import RUN 90
 SIMULATE 87, 90
 MULTIPLE parameter
 SETPRINT command 193

N

NAME parameter
 MODEL command 184, 209
 name patterns
 fingerprint data set 33
 NBYONE parameter
 EXTRACT command 155, 160
 NEWDSN parameter 162
 EXTRACT command 155
 NEWDSNTOVOL parameter
 IMPORT command 168
 NEWVOL parameter 163
 EXTRACT command 155
 NOECHO 187
 NONSTD subparameter
 BASE parameter 173
 NOREPLACE parameter
 IMPORT command 169
 NOTINCLDSN parameter 193

O

object
 wildcards 245
 OLDDSNTOVOL parameter
 IMPORT command 168
 ONLINE 188
 online documentation 11
 online help 12
 Online Synchronization Facility
 accessing 18
 exiting 18
 ORPHAN subparameter
 BASE parameter 173
 ORPHANCHECK 188
 OUTSYNC parameter
 SETPRINT command 193
 overview 17

P

package changes 36, 49
 packaging data sets 38
 parameters
 ALL 156
 COMPOSITE 137
 DSNBYDSN 138
 FILTER 142
 for data set compare 72
 for extracting and exporting 84
 MEMTOMEM 144
 NEWDSN 162
 NEWVOL 163
 SCRDSN 164
 pattern list
 fingerprint data set 30
 pattern names
 for HFS 201
 PDS
 comparing 64
 PRECOMPRESS parameter
 IMPORT command 168
 printing reports 19, 63

R

README file 9
recalling
 fingerprint data set 31
RECURSE parameter 203
redundancy
 level 231
 management 111
 report examples 233
 reporting 233
redundant data 111
REFTARG 189
REJDSN parameter
 SETPRINT command 193
REJECT parameter
 IMPORT command 169
REJECT subparameter 161
 EXTRACT DSNBYDSN parameter 158
related documents, ChangeMan SSM 11
remote environments 101
REPLACE parameter
 IMPORT command 169
REPORT command 238
reports
 Actual Differences 61
 Comparex 61
 comparison 61
 print 63
 SuperC 61
 viewing Comparex 62
reserved DDs
 HPSIN DD 118
 HPSPLIB DD 118, 238
RESET parameter 121, 203
resolving data sets 58, 60
RESTART 190
restarting
 a fingerprint job 231
 redundancy detection 190
results
 interpreting 35
REVERSDD parameter
 IMPORT command 169
REVERSDSN parameter
 IMPORT command 169
reverse change baskets 15, 109
REVERSE parameter
 HFSIMPORT command 208
 IMPORT command 169
REVERSEDD parameter
 HFSIMPORT command 207
REVERSEDSN parameter
 HFSIMPORT command 207
reversing changes 96
RUN mode 87

S

safeguards 101
 fingerprint data set 107
SAVE parameter 121, 203
saving a BASE environment 130
Scr keyword 152, 155
SCRATCH parameter
 IMPORT command 169
SCRDSN parameter 164
 EXTRACT command 155
search strings and masking 104
SEQUENCE subparameter
 DSNTODSN parameter 141
SERCOPY program 165
SETPR 193
SETPRINT command
 EXCLDSN parameter 193
SETPRINT ommand
 INCLDSN parameter 193
SETPRINTcommand 191
SHORT subparameter
 DSNBYDSN parameter 139
SHOW parameter
 MASK command 175, 180
SIMULATE command 196
SIMULATE parameter
 HFSIMPORT command 207
 IMPORT command 169
SINGLE parameter
 SETPRINT command 194
SMSMAP command 196
snap files 238
 comparing 239, 241
SNAPFILE command 238
SNAPSHOT command 238
snapshot comparison utility 219
snapshot facility 217, 237, 238
snapshots 212
 comparing 212
source and text masking parameters
 bytes before/after 24
 language masking 24
 MIXED or UPPER case 24
 squeezing white spaces 24
 STRING search and mask 24
source data set 107
SPACE
 exceptional conditions 237
special characters
 ! (string search delimiter) 245, 249
 ? (wildcard character specification) 245, 249,
 250
 * (node specification) 245, 250
 / (search terminator notation) 245
specifying base environment 55

- specifying target environment 67
- squeeze masking 106
- SQUEEZE parameter
 - MASK command 175, 181
- squeezing white spaces
 - source and text masking parameters 24
- stacking commands 117
- STRING parameter
 - MASK command 176, 181
- STRING search and mask
 - source and text masking parameters 24
- STRING subparameters
 - AFTER 182
 - BEFORE 182
 - HIBOUND 182
 - LOBOUND 182
 - VALUE 182
- subparameters
 - DETECT 147
- super directory 110
 - control area 83
- SuperC 55
 - generate reports 61
- SuperC subparameter
 - DSNBYDSN parameter 138
 - FILTER parameter 142
 - MEMTOMEM parameter 145
- SYMBOLIC subparameter
 - CATLG parameter 165
- symbols
 - n (node identifier) 245
 - X (member hex notation) 249
- sync check 55
 - results 61
- sync check (online function) 55
- SYNC parameter
 - SETPRINT command 194
- Synchronization
 - Menu 21
- synchronization
 - commands 109, 119
 - environments 109
 - functions 18
 - menu 18
 - results 61
- synchronizing
 - environments 65
- synchronizing data 90
- synchronizing environments (online batch function) 65
- SYNCTRAC DD
 - reserved DDs 118
- SYSPLEX DD
 - reserved DDs 118

T

- TARGDSN subparameter
 - FILTER parameter 143
 - MEMTOMEM parameter 145
- TARGET 198
- target environment
 - specifying 59, 67
- TARGET fingerprints
 - comparing 92
- target fingerprints 15
- TARGET parameter
 - LIST command 172
- TIME parameter
 - MASK command 176
- token data sets 111
- TOKEN subparameter
 - DSNTODSN parameter 141
- tokens
 - generating 53
- TONAME subparameter
 - NBYONE parameter 161
- TRACE command 198
- transferring
 - change baskets 110
 - FGP or BASKET 212
 - FGPs 214
 - files 213
- TRANSFORM parameter
 - MODEL command 184, 209
- TSO RECEIVE command 253
- TYPE parameter
 - DASDRED command 149
- TYPE subparameter
 - GENERIC parameter 125
 - LOAD parameter 128
 - VOL parameter 133

U

- uncataloged data sets
 - comparing 56
- undoing imports 89, 96
- Upd keyword 152, 155
- uploading
 - change basket PC file 253
 - change baskets to MVS 252
- UPPER subparameter
 - CASE parameter 178
- user options 19

V

- VALUE subparameter

- STRING parameter 182
- verify
 - synchronization 55
- verify synchronization
 - data set groups 73
 - results 61
- verifying
 - fingerprint data set name 34
- VIEWBASKET command 199
- VIEWFGP command 198
- viewing
 - baskets 42, 97
 - fingerprints 45
- VOL parameter 121
- VOLMAP command 200
- VOLSER command 238
- volser parameter
 - ONLINE command 188
- VOLSER subparameter
 - EXTRACT NEWVOL parameter 163
- volume
 - wildcards 250
- volumes
 - comparing 79
 - tokenizing 230
- VSAM cluster
 - wildcards 245
- VTOC parameter
 - SETPRINT command 194
- VVDS parameter
 - SETPRINT command 194

W

- wildcards 101, 243
 - data set, object and cluster 245
 - member 249
 - volume 250

X

- XMIT data set
 - allocating 251