SERENA®

ChangeMan® ZMF 8.1.3

Installation Guide

Serena Proprietary and Confidential Information

Copyright © 2001-2017 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Trademarks

Serena, TeamTrack, StarTool, PVCS, Comparex, Dimensions, Prototype Composer, Mariner and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo, Version Manager and Mover are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 2345 NW Amberbrook Drive, Suite 200, Hillsboro, OR 97006 USA

Publication date: March 2017

Table of Contents

Welcome to Serena® ChangeMan® ZMF 9
Guide to ChangeMan ZMF Documentation 10
ChangeMan ZMF Documentation Suite 10
Using the Manuals 11
Searching the ChangeMan ZMF Documentation Suite 12
Choosing the Right Installation/Upgrade Manual 12
ChangeMan ZMF Release Notes 13
Online Help 13
Online Tutorial 13
Online Help Screens 13
Online Error Messages 13
Typographical Conventions 14
Notes 14

Chapter 1 **Introduction 15**
ChangeMan ZMF 16
Protecting Production and Development 16
Change Package 16
Change Package Life Cycle 17
SERNET 19
Implementation Strategy 20

Chapter 2 **Installation Requirements 23**
System Requirements 24
Operating System 24
Software Requirements 24
Memory Requirements 24
DASD Space Requirements 25
ISPF Profile Library Allocation 25
REXX Runtime Environment 26
Serena ChangeMan ZMF Product Compatibility 26
CA Panvalet® and CA Librarian® Compatibility 26
Software Conflicts 27
Library Concatenations 27
Abend-AID® (from Compuware Corporation) 27
PDSFAST (from Software Engineering of America) 27
CA PDSMAN® (from Computer Associates) 27
MAINVIEW® SRM StopX37/II™ (from BMC Software) 28
VSAM Performance Tools 28
Load Library APF Authorization 28
IPL 29

Chapter 3 **Installation Considerations** 31
Upgrading From Earlier Versions 32
System Considerations 32
z/OS Subsystem 32
Updating the System Linkage Index 32
Non-Swappable 32
Reusable Address Space Identifiers (ASID) 32
Serena Libraries and LINKLIST 33
VSAM Performance 33
Defining VSAM file characteristics 33
AMP Parameters 33
VSAM I/O Optimization of ZMF Master Files 33
VSAM Linear Data Sets 34
SERNET and ChangeMan ZMF JCL 35
Subsystem ID 35
SERNET Started Task Names 35
Parameters for SERNET and ChangeMan ZMF 36
SER#PARM DD Statement 37
SERLIC DD Statement 38
SYSMDUMP DD Statement 39
SYSTCPD DD Statement 39
Started Procedures for File Tailoring 39
Security Considerations 40
Exclusive Access to Libraries 40
Access to ChangeMan ZMF Functions 40
Security Entities and Required Authority 42
SAF and Your Security System 43
Security for File Tailoring Started Procedures 43
Access to TCP/IP Functions 43
Data Set Considerations 43
Temporary List Data Sets 44
Utility Data Sets 44
Staging Library Model Data Set Name 45
Batch Job Name Considerations 46
User Job Names 46
Install Job Names 47
Viewing Job SYSOUT 47
Staging Versions 48
Customizing ChangeMan ZMF Components 48
Console Log Messages 49

Chapter 4 **Unloading the Software** 51
Introduction 52
ChangeMan ZMF Libraries 52
Documentation 53
Unload Process 53
Requirements 53
Step 1: Unload the ChangeMan ZMF Files 53

Step 2: Transfer Files to the Mainframe 55
Step 3: Expand PDS Libraries with RECEIVE 57

Chapter 5 **Installing ChangeMan ZMF Components 59**
Step 1: Allocate CUSTOM Libraries. 60
Step 2: APF Authorize Delivered and CUSTOM Load Libraries 60
Step 3: Apply ChangeMan ZMF License 61
Step 4: Define ChangeMan ZMF VSAM Files 61
Step 5: Define XMLSPACE VSAM LDS 63
Step 6: Define Staging Versions VSAM Files 65
Step 7: Store SVPARM Control Member 69
Step 8: Allocate the SER#PARM TCP/IP Address Library 70
Step 9: Create ISPF Keylist for EXPAND 70
Step 10: Build SERNET JCL for ChangeMan ZMF 72
Step 11: Build Default File Tailoring Procedure 75
Step 12: Build the Logon CLIST. 76
Step 13: Modify Exit Program CMNEXINS 77
Step 14: APF Authorize Other Load Libraries. 78

Chapter 6 **Configuring Security 79**
Introduction 80
Step 1: Customize SERLCSEC 80
CA Top Secret Security Violations 80
CA ACF2 Batch LOGONID 81
Assembling SERLCSEC. 81
Stubbing ChangeMan ZMF Security 82
Step 2: Add ChangeMan ZMF to Your Security System 82
Add ChangeMan ZMF to CA ACF2 82
Add ChangeMan ZMF to IBM Security Server RACF 85
Add ChangeMan ZMF to CA Top Secret 87
Step 3: Define Security Entities. 88
Step 4: Define Data Set Access 91
Libraries and Data Sets in the SERNET Started Procedure 91
Other Libraries and Data Sets. 92
Step 5: Add OMVS Segment To Use TCP/IP. 93
Step 6: Add PassTicket Support In Sernet 93
RACF Administration Required 94
The SERSET Utility 94
Generating a PassTicket. 94
Failures in PassTicket Generation 94
TOKEN =Operand of SERSET 95
Refreshing SERXPTIK 95
Step 7: Set Security for USS File Systems 96

Chapter 7 **. 97**

Starting, Connecting, and Stopping 97
Start SERNET 98

SERPRINT Messages 98
Subsystem IDs 100
Connect to the ChangeMan ZMF Primary Option Menu 100
Stop SERNET and ChangeMan ZMF 101

Chapter 8 **Modifying Skeletons 103**
Introduction 104
System Library and Data Set Names 104
CMN$$DSN - JOBLIB 108
CMN$$SEX - REXX SYSEXEC 108
CMN$$IAX - Impact Analysis Data Extract 109
CMN$PARM - Build Parameters 110
CA ACF2 LOGONID in JOB Statement Skeletons 111
Software Conflicts 111

Chapter 9 **Installing Other Facilities 113**
Installing XMLSERV 114
Installing CICS NEWCOPY Support 115
Installing NEWCOPY Option OPS 116
Installing NEWCOPY Option OPQ 116
Installing NEWCOPY Option XCI 116
Configure Automated Operations Tools 119

Chapter 10 **Setting Up Housekeeping 121**
Introduction 122
Backup Jobs 122
Restore Jobs 123

Chapter 11 **Installing A ChangeMan ZMF P Instance 125**
Introduction 126
Sharing Libraries and Data Stores 126
Installing P Instance Components 127
Configuring P Instance Security 131
Starting, Connecting, and Stopping 132
SERPRINT Message Example for P Instance 132
Modifying P Instance Skeletons 133
Installing Other P Instance Facilities 133
Setting Up P Instance Housekeeping 134

Appendix A **Manual File Transfer and Expand 135**
Transfer Files To Host Mainframe 136
File Transfer Using FTP 136
File Transfer Using 3270 Emulator 137
Expand Host Target Files 139

Appendix B **JCL and CLIST Examples 141**
Introduction 142
SERNET Started Procedure With ChangeMan ZMF 142

ChangeMan ZMF Logon CLIST 142
CMNxADSP Default File Tailoring Procedure. 142
ChangeMan ZMF P Instance 143

Appendix C **Applying Maintenance Releases 145**
Introduction 146
Updating Delivered Libraries 146
Updating Custom Libraries 147
Managing Copybook Changes 148
Binding DB2 DBRM. 148
Managing XML Services Changes 149
Synchronizing Impact Analysis Data. 149
Updating Housekeeping Jobs. 150

Appendix D **SERNET Keyword Options 151**

Appendix E **Modify Commands 167**
Introduction 168
Standard Modify Commands 168
Restricted Modify Commands 181

Appendix F **Technical Notes 187**
TCP/IP Trouble Shooting. 188
Trusted User ID For ALM Logon To ChangeMan ZMF. 189

Appendix G **ZDDOPTS: ChangeMan ZDD XML Options. 191**
Introduction 192
ZDDOPTS Members 192
XML Syntax 193
Multiple Attribute Values 193
Wildcard Patterns 194
Translating Comments in ZDDOPTS Members 194
AUDIT, DEMOTE, and PROMOTE Options 196
Elements 197
Attributes for <options> Element 197
Attributes for <profile> Element 197
Attributes for <field> Element 197
Name Values for <field> Element 199
Example. 199
BUILD Options. 199
Elements 200
Attributes for <options> Element 200
Attributes for <multiple> Element. 201
Attributes for <profile> Element 201
Attributes for <field> Element 201
Name Values for <field> Element 203
Example. 204
COMMAND Options. 205

Elements 205
Attributes for <options> Element 205
Attributes for <profile> Element 206
Attributes for <command> Element 206
Example 207
LIBTYPE Options 207
Elements 208
Attributes for <options> Element 208
Attributes for <profile> Element 209
Attributes for <libtype> Element 209
Example 209
PKGCREAT Options 209
Elements 210
Attributes for <options> Element 210
Attributes for <profile> Element 210
Attributes for <field> Element 211
Name Values for <field> Element 212
Example 212
PKGPROP Options 213
Elements 213
Attributes for <options> Element 213
Attributes for <profile> Element 214
Attributes for <field> Element 214
Example 214

Index 217

Welcome to Serena® ChangeMan® ZMF

Before you Begin

See the Readme for updates and corrections for this manual. For the latest Readme, go to the Micro Focus SupportLine website at https://supportline.microfocus.com/.

Audience and scope

This manual is intended for system administrators responsible for administering and installing ChangeMan ZMF.

Use this manual to accomplish the following:

- Install ChangeMan ZMF for the first time in your environment
- Build a new environment to run the current release of ChangeMan ZMF

Use this document if you are responsible for any of these tasks:

- Installing ChangeMan ZMF components
- Providing Global or Application Administration for ChangeMan ZMF
- Managing mainframe change control or configuration management
- Providing technical support for mainframe started tasks
- Managing the mainframe security system at your data center

This manual assumes that you are familiar with your mainframe operating system and experienced with your security system.

NOTE If you are upgrading from a previous version of ChangeMan ZMF, you must use the appropriate *ChangeMan ZMF Migration Guide*.

Components

Components for ChangeMan ZMF are delivered in two sets of libraries.

- The libraries with names that include CMNZMF as a node contain the ChangeMan ZMF base product and all selectable options
- The libraries with names that include SERCOMC as a node contain components for the following enabling technologies:
  - SERNET (known previously as Serena/Network), which manages communication and Serena XML Services for Serena mainframe products
  - Serena® License Manager™, which enables licensed Serena mainframe software on authorized CPUs
  - Common components used in multiple Serena products and enabling technologies

Navigating this book

This manual is organized as follows:

- Chapters 1-3 cover things to think about, things to decide, and things to do before you install ChangeMan ZMF.
- Chapters 4-7 provide step-by-step instructions for Installing ChangeMan ZMF components, configuring security, starting the SERNET started task where your ChangeMan instance runs, and connecting to ChangeMan ZMF in an ISPF environment.
- Chapters 8-10 provide instructions for making required modifications to ISPF skeletons, installing other tools, and setting up housekeeping jobs like VSAM file backup.

- The Appendixes provide examples, reference material, and other supplemental information.

Change Bars

Change bars in the left margin of a page identify changes that have been made to this manual in this release.

Guide to ChangeMan ZMF Documentation

The following sections provide basic information about ChangeMan ZMF documentation.

ChangeMan ZMF Documentation Suite

The ChangeMan ZMF documentation set includes the following manuals in PDF format.

Manual	Description
Administrator's Guide	Describes ChangeMan ZMF features and functions with instructions for choosing options and configuring global and application administration parameters.
ChangeMan ZMF Quick Reference	Provides a summary of the commands you use to perform the major functions in the ChangeMan ZMF package life cycle.
Customization Guide	Provides information about ChangeMan ZMF skeletons, exits, and utility programs that will help you to customize the base product to fit your needs.
DB2 Option Getting Started Guide	Describes how to install and use the DB2 Option of ChangeMan ZMF to manage changes to DB2 components.
ERO Concepts	Discusses the concepts of the ERO Option of ChangeMan ZMF for managing releases containing change packages.
ERO Getting Started Guide	Explains how to install and use the ERO Option of ChangeMan ZMF to manage releases containing change packages.
High Level Language Functional Exits Getting Started Guide	Provides instructions for implementing and using High Level Language (Cobol, PL/1, and REXX) exits, driven consistently by all clients to enforce local business rules in ZMF functions.
IMS Option Getting Started Guide	Provides instructions for implementing and using the IMS Option of ChangeMan ZMF to manage changes to IMS components.
INFO Option Getting Started Guide	Describes two methods by which ChangeMan ZMF can communicate with other applications: ■ Through a VSAM interface file. ■ Through the Tivoli Information Management for z/OS product from IBM.

Manual	Description
Installation Guide	Provides step-by-step instructions for initial installation of ChangeMan ZMF. Assumes that no prior version is installed or that the installation will overlay the existing version.
Java / HFS Getting Started Guide	Provides information about using ZMF to manage application components stored in USS file systems, especially Java application components.
Load Balancing Option Getting Started Guide	Explains how to install and use the Load Balancing Option of ChangeMan ZMF to connect to a ZMF instance from another CPU or MVS image.
M+R Getting Started Guide	Explains how to install and use the M+R Option of ChangeMan ZMF to consolidate multiple versions of source code and other text components.
M+R Quick Reference	Provides a summary of M+R Option commands in a handy pamphlet format.
Messages	Explains messages issued by ChangeMan ZMF, SERNET, and System Software Manager (SSM) used for the Staging Versions feature of ZMF.
Migration Guides	Provides guidance for upgrading ChangeMan ZMF. Refer to the Readme for more information.
OFM Getting Started Guide	Explains how to install and use the Online Forms Manager (OFM) option of ChangeMan ZMF.
SER10TY User's Guide	Gives instructions for applying licenses to enable ChangeMan ZMF and its selectable options.
User's Guide	Describes how to use ChangeMan ZMF features and functions to manage changes to application components.
XML Services User's Guide	Documents the most commonly used features of the XML Services application programming interface to ChangeMan ZMF.
ZMF Web Services User's Guide	Documents the Web Services application programming interface to ChangeMan ZMF.

Using the Manuals

Use Adobe® Reader® to view ChangeMan ZMF PDF files. Download the Reader for free at get.adobe.com/reader/.

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 with a single mouse click. These links appear in blue.

- **Comments.** All PDF documentation files that Serena delivers with ChangeMan ZMF have enabled commenting with Adobe Reader. Adobe Reader version 7 and higher has commenting features that enable you to post comments to and modify the contents of PDF documents. You access these features through the Comments item on the menu bar of the Adobe Reader.
- **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.

Searching the ChangeMan ZMF Documentation Suite

There is no cross-book index for the ChangeMan ZMF documentation suite. You can use the Advanced Search facility in Adobe Acrobat Reader to search the entire ZMF book set for information that you want. The following steps require Adobe Reader 6 or higher.

1. Download the ZMF All Documents Bundle ZIP file and the *ChangeMan ZMF Readme* to your workstation from the My Downloads tab on the Serena Support website.
2. Unzip the PDF files in the ZMF All Documents Bundle into an empty folder. Add the *ChangeMan ZMF Readme* to the folder.
3. In Adobe Reader, select **Edit** | **Advanced Search** (or press **Shift+Ctrl+F**).
4. Select the **All PDF Documents in** option and use **Browse for Location** in the drop down menu to select the folder containing the ZMF documentation suite.
5. In the text box, enter the word or phrase that you want to find.
6. Optionally, select one or more of the additional search options, such as **Whole words only** and **Case-Sensitive**.
7. Click **Search**.
8. In the **Results**, expand a listed document to see all occurrences of the search argument in that PDF.
9. Click on any listed occurrence to open the PDF document to the found word or phrase.

Choosing the Right Installation/Upgrade Manual

Choose the manual that fits your situation when installing or upgrading ChangeMan ZMF.

Your task	Manual to use
Installing ChangeMan ZMF for the first time,	*ChangeMan ZMF Installation Guide*
Building a new ChangeMan ZMF instance from scratch	*ChangeMan ZMF Installation Guide*
Upgrading from a prior ChangeMan ZMF	*ChangeMan ZMF Migration Guides as listed in the Readme*

IMPORTANT! Always see the most current *Readme* for your ChangeMan ZMF release in case it contains documentation updates for the installation/upgrade manual you use.

ChangeMan ZMF Release Notes

High-level descriptions of the enhancements that are delivered in ChangeMan ZMF 8.1 and later releases are included in the "Features and Fixes" section of the latest *ChangeMan ZMF Readme*.

Online Help

Online help is the primary source of information about ChangeMan ZMF. Online help is available as a tutorial, through Help screens, and in ISPF error messages.

Online Tutorial

ChangeMan ZMF includes an online tutorial that provides information about features and operations, from high-level descriptions of concepts to detailed descriptions of screen fields.

To view the tutorial table of contents, select option T from the Primary Option Menu, or jump to it from anywhere in ChangeMan ZMF by typing =T and pressing ENTER.

Press PF1 from anywhere in the Tutorial for a complete list of Tutorial navigation commands and PF keys.

Online Help Screens

If you have questions about how a ChangeMan ZMF screen works, you can view a help panel by pressing PF1 from anywhere on the screen.

Online Error Messages

If you make an invalid entry on a ChangeMan ZMF screen, or if you make an invalid request for a function, a short error message is displayed in the upper right corner of the screen. Press PF1 to display a longer error message that provides details about the error condition.

Remember that the long message does not display automatically. Request the long message by pressing PF1.

Typographical Conventions

The following typographical conventions are used in the online manuals and online help. These typographical conventions are used to assist you when using the documentation; they are not meant to contradict or change any standard use of typographical conventions in the various product components or the host operating system.

Convention	Explanation
italics	Introduces new terms that you may not be familiar with and occasionally indicates emphasis.
bold	Emphasizes important information and field names.
UPPERCASE	Indicates keys or key combinations that you can use. For example, press the ENTER key.
`monospace`	Indicates syntax examples, values that you specify, or results that you receive.
`monospaced italics`	Indicates names that are placeholders for values you specify; for example, *`filename`*.
vertical rule |	Separates menus and their associated commands. For example, select File | Copy means to select Copy from the File menu. Also, indicates mutually exclusive choices in a command syntax line.

Notes

Sterling Connect:Direct® is an IBM® point-to-point file transfer software product that can be used to transfer files between two ChangeMan ZMF instances. The original name of the product was Network Data Mover (NDM). The "NDM" mnemonic persists, embedded in Connect:Direct and ChangeMan ZMF component names, options, and JCL examples.

Chapter 1
Introduction

This chapter provides information about Serena ChangeMan ZMF.

ChangeMan ZMF	16
Protecting Production and Development	16
Change Package	16
Change Package Life Cycle	17
SERNET	19
Implementation Strategy	20

ChangeMan ZMF

Serena® ChangeMan® ZMF protects corporate software assets by automating the development process, resulting in higher application quality and increased application availability. It's a comprehensive solution that provides reliable and streamlined implementation of software changes in a z/OS environment.

Since ChangeMan ZMF controls every code change, it guarantees source-to-load integrity and makes sure that only successfully tested programs make it into production. By moving code through an automated life cycle with strict accountability every step of the way, ChangeMan ZMF reduces regression errors and maintenance costs.

Protecting Production and Development

ChangeMan ZMF answers the question, "What is production?" by including all development and test environments within the controls traditionally applied only to production execution libraries.

ChangeMan ZMF is granted exclusive update access by your security system to baseline and production libraries, and exclusive update access to libraries in controlled test environments. ChangeMan ZMF creates and manages the staging libraries where developers check out and modify application components.

The components in all of these libraries are protected by restricting access to the ChangeMan ZMF functions that can change the contents of these libraries. Developers, approvers, testing coordinators, ChangeMan ZMF administrators, and others can execute only those functions in ChangeMan ZMF for which they have been granted authorization by your security system.

All functions in ChangeMan ZMF are subject to rules defined in ChangeMan ZMF administration. Processes are consistent and repeatable to the extent that your ChangeMan administrators define rules that constrain user choices and options.

Change Package

A ChangeMan ZMF change package is a unit of work for application component changes being made for a project.

A change package consists of descriptive information, control parameters, and history information stored in ChangeMan ZMF VSAM master files. A set of staging libraries that belong exclusively to the change package contain application source, load, and other components that are being changed.

A change package is a secure development environment for project components. Access to a package and to package components is managed by ChangeMan ZMF using rules stored in your security system.

Rules in ChangeMan ZMF administration set constraints on options for managing components in change packages. Other administration rules restrict options for managing change packages through the change package life cycle.

Package components remain in package staging libraries throughout the ChangeMan ZMF package life cycle. Components are copied from package staging libraries to test libraries for testing, and components are copied from staging libraries to production libraries for installation.

Change Package Life Cycle

The change package life cycle is rules-based process consisting of:

- Actions that developers, approvers, testing coordinators, and others perform on change packages and change package components
- Processes that are automatically initiated by ChangeMan ZMF when certain conditions are met

Authority to perform actions in the package life cycle is controlled by ChangeMan ZMF and defined in your security system. Rules in ChangeMan ZMF administration define what options are available in the change package life cycle.

The steps in the change package life cycle include the following:

- **Create Package** is the first step in the change package life cycle. Using a series of input panels, screens, or windows, you enter information that describes the change package, and you set control parameters that determine how the package behaves during the rest of the package life cycle.
- **Checkout Component** copies components from a baseline or promotion library into a staging library allocated exclusively to your package. You can also check out components to a personal library, which is tracked by ChangeMan ZMF.
- **Stage Component** is where you edit and build package components to meet project requirements. Source components are processed through predefined build processes to create executables and build listings. You can also stage components into your package from libraries outside of ChangeMan ZMF to bring those components under the control of ChangeMan ZMF.
- **Package Audit** detects problems that will occur in production if you install your package in its current condition. Audit detects synchronization problems between components in your change package, and it detects synchronization problems between package components, components in participating packages, components in promotion, and baseline components.
- **Freeze Package** locks package information and package components to prevent further changes and to ensure that the components you install into production are the same as the components you tested. You can selectively unfreeze, change, audit, and refreeze components to fix problems found in testing.
- **Promote Package** copies package components from staging libraries into test libraries. As a package is promoted from one testing level to the next, package components are removed from libraries in the prior level and copied from staging libraries into test libraries for the next level.
- **Demote Package** removes package components from test libraries.
- In **Approve Package**, predefined approvers review package information, components, and test results and approve or reject the package for install. If an approver rejects the package, they enter text Reject Reasons.

- **Revert Package** removes all previously entered approvals, unlocks package information and components, and opens the package back up to development.
- **Distribute Package** occurs if you are installing a package through another ChangeMan ZMF instance at a remote site. It starts automatically when all package approvals are entered for a package. The package is transmitted to the remote site where package records are added to the ChangeMan ZMF instance running there, package staging libraries are allocated and populated, and the package is added to the internal scheduler.
- **Install Package** starts automatically, either when the package install date and time arrive, when the last approval is entered, or when an external job scheduling system triggers the first installation job. If the application has production libraries that are separate from baseline libraries, current production modules are backed up and new versions are copied from package staging libraries into the production libraries.
- **Baseline Package** starts automatically after a package is installed. This process ripples current and prior versions of package components down in a stack of prior baseline versions, and then copies package components into the baseline libraries as the new current version.
- **Backout Package** removes package components from production libraries and restores the backups made when the package was installed. Package components that are the current version in baseline libraries are removed, and components are reverse rippled up the stack of prior baseline versions to restore the old current version.

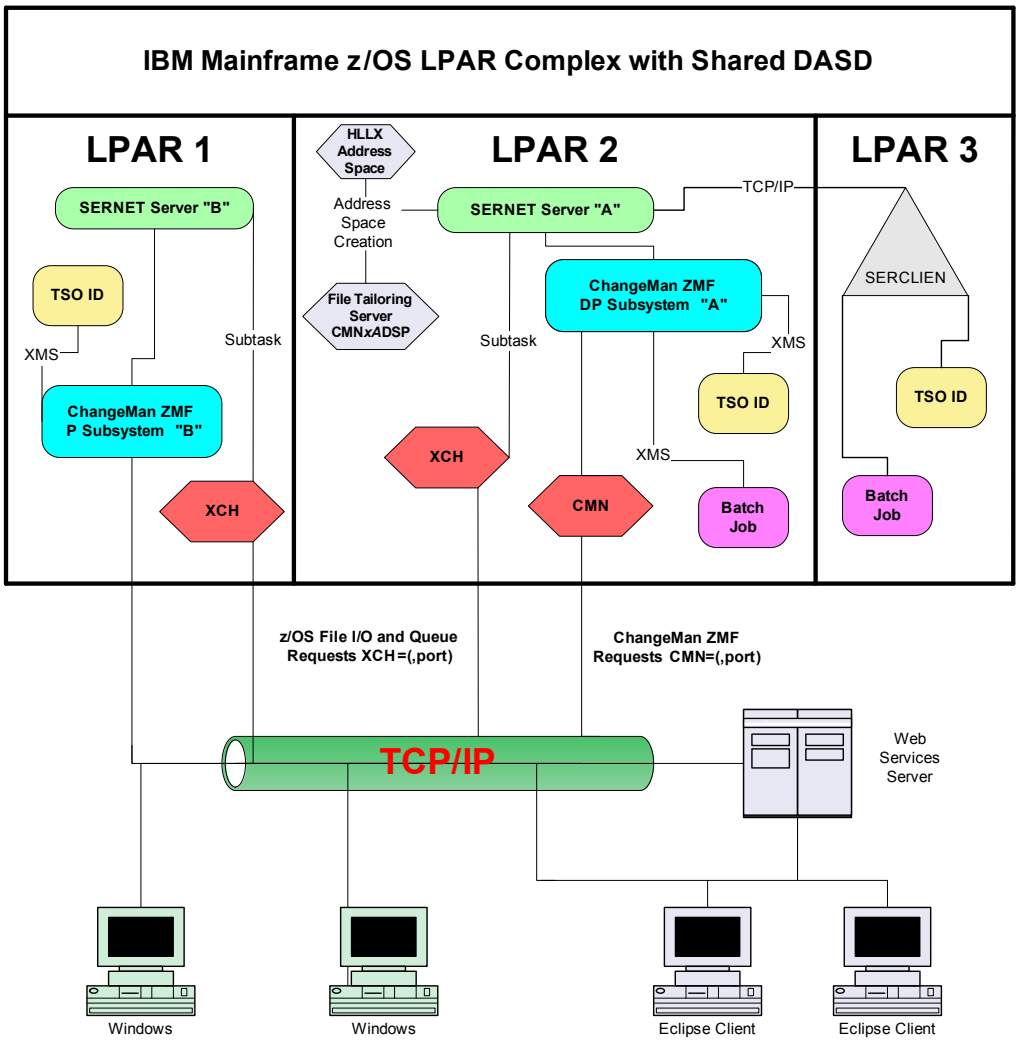
SERNET

SERNET (previously called SERENA/Network) provides communication and other services on the enterprise server for Serena products. SERNET runs as a started task on an LPAR.

Some Serena products, such as ChangeMan ZMF, run as applications under a SERNET instance. Other products, such as Serena® ChangeMan® ZDD, run on other platforms and use a SERNET instance to gain access to mainframe files and services.

The diagram below is a logical view of the Serena SERNET architecture. It shows two SERNET instances on separate LPARs, each managing a ChangeMan ZMF instance. Users access these ChangeMan ZMF instances from TSO sessions in the z/OS environment, ChangeMan ZDD running on a Windows workstation, or from ChangeMan ZMF for Eclipse running on an Eclipse workstation. The TSO user in the last LPAR is using the Load Balancing Option of ChangeMan ZMF to work from a mainframe environment where there is no SERNET or ChangeMan ZMF instance.

Serena Software SERNET Architecture



Serena ChangeMan ZDD	Serena ChangeMan ZMF for Eclipse
Direct access to ChangeMan ZMF components in a change package from a workstation running Windows .	Direct access to ChangeMan ZMF components in a change package from a workstation running Eclipse .

Implementation Strategy

ChangeMan ZMF is designed to be flexible so that it can serve customers who have a broad range of data center standards and change management requirements. A by-product of that flexibility is that there are choices you must make before you configure ChangeMan ZMF to manage components in your development and production environments.

These choices include:

- How you will define applications in ChangeMan ZMF
- How you will configure baseline and production libraries in relation to those applications
- What kind of package approval process you need
- What roles you want people to play, and what authority they will require to access ChangeMan ZMF functions
- Whether you will create multiple ChangeMan ZMF instances to manage production application components across your enterprise

These issues are discussed in the first few chapters of the *ChangeMan ZMF Administrator's Guide*, and additional details are found in the *Administrator's Guide* chapters that tell you how to configure Global and Application Administration. Your organization needs to examine these issues and find consensus among developers, development managers, testing groups, data center operations, security, EDP audit, IT management, and business partners.

You can help these groups understand the issues and the alternatives available by installing ChangeMan ZMF and bringing up a test or demonstration ChangeMan ZMF instance where you can show them how ChangeMan ZMF works. As you reach tentative decisions about the configuration want to use, you can change your initial settings in administration, create new applications, or build new ChangeMan ZMF instances to test your design.

You can also use information in the *ChangeMan ZMF Customization Guide* to explore exit programs and skeleton modifications to satisfy your requirements.

This *Installation Guide* tells you how to install ChangeMan ZMF components and make entries in your security system so that you can bring up a ChangeMan ZMF instance.

When you get a ChangeMan ZMF instance running, Serena recommends that you use the instructions in chapter "Setting Up Global Administration" and chapter "Setting Up Application Administration" in the *ChangeMan ZMF Administrator's Guide* to set up the simplest ChangeMan ZMF configuration possible, accepting system defaults where they are available and using the most liberal parameter settings possible.

When you have configured global and application administration in ChangeMan ZMF, create some simple JCL and source components in the new baseline libraries. These components do not need to be executable, but the source code must compile successfully.

Use the instructions in the *ChangeMan ZMF User's Guide* to create a change package with no components, and process this empty package through the change package life cycle. Next, create a package and check out components into the package. Stage some simple source components so you can see compile and link edit processing. Process the package through the life cycle.

Use this incremental approach to exercise more and more features of ChangeMan ZMF as you gain knowledge of how you can implement ChangeMan ZMF to meet your requirements.

If you license one or more ChangeMan ZMF selectable options, use the installation instructions in the *Getting Started Guide* for each of those options to add those facilities to your test or demonstration instance of ChangeMan ZMF. Experiment with those options to decide how you want to implement them.

Chapter 2
Installation Requirements

This chapter describes requirements for installing Serena® ChangeMan® ZMF in your mainframe environment.

System Requirements	24
Serena ChangeMan ZMF Product Compatibility	26
CA Panvalet® and CA Librarian® Compatibility	26
Software Conflicts	27
Load Library APF Authorization	28
IPL	29

System Requirements

This section defines operating system and software requirements, memory requirements, and disk storage requirements.

Operating System

ChangeMan ZMF runs without modification on all IBM and plug-compatible mainframes running z/OS®. See the Readme for a link to current information about OEM software compatibility testing.

Software Requirements

Refer to the current platform matrix on the support website to determine what system software to run ChangeMan ZMF in your mainframe environment.

- You will need one of the following external security systems:
  - IBM Security Server RACF®
  - CA ACF2®
  - CA Top Secret®
- The MVS SAF (System Authorization Facility) is used to interface to your security system.

If you intend to use ChangeMan ZMF to distribute components to remote CPUs for testing or installation, you need one of the following data transport systems:

- Bulk Data Transfer (BDT)
- Connect:Direct® (formerly NDM)
- NetMaster
- Netview® FTP
- XCOM 6.2

Memory Requirements

SERNET runs as a started task on each CPU that employs SERNET services. You may run multiple SERNET instances on a single processor to test SERNET or its applications, to run different Serena products at different release levels of SERNET, or to achieve separation of control.

The recommended region size for a SERNET instance is zero megabytes (REGION=0M), which provides the absolute maximum memory above and below the line. SERNET does virtually all work in memory above the line, without spilling to DASD. To support hundreds

of concurrent users, a large amount of virtual memory is used. The symptom of insufficient memory is abend S878.

NOTE If there are system exits that disallow REGION=0M, then set the SERNET region size to at least 48 megabytes (REGION=48M). Consider removing exits that limit memory acquisition, such as IEALIMIT, which limits an address space from acquiring more than 32 megabytes above the line, or increase the maximum allowed to at least 48 megabytes

Each SERNET instance obtains 64 bytes of CSA (Common System Area subpool 245) to hold a SubSystem Communications Vector Table (SSCVT) control block. One SSCVT is required for each SERNET started task on a processor. To avoid fragmenting memory, this short piece of memory is never released (FREEMAIN). When SERNET is shut down and restarted, it reuses the same anchor block until the next IPL clears it out.

Almost all SERNET mainframe programs are written in reentrant assembly language. Most load modules have been given the RENT attribute at link-edit time as well as AMODE=31. Some load modules reside below the line using RMODE=24, but most reside above the line using RMODE=ANY. All memory that can be managed above the line is acquired there for memory constraint relief.

DASD Space Requirements

Eight VSAM KSDS files and two VSAM LDS files are created during the installation process. The recommended minimum total primary allocation is 94 cylinders of 3390 disk.

ISPF Profile Library Allocation

When you use the ISPF interface to ChangeMan ZMF, it creates up to three members in your ISPF profile library (ISPPROF).

Name	Function	Description
xxxxPROF	ZMF ISPF profile	Used to store ZMF user variables from the ISPF profile variable pool between ZMF sessions. Panel variables like job card images and package create fields are saved to the profile at the end of a ZMF session (VPUT) and retrieved at the start of a new session (VGET) so that panel fields can be pre-filled.
xxxxEDIT	ISPF edit	Transparent to ZMF
xxxxEDRT	ISPF edit recovery	Transparent to ZMF

If you license the ChangeMan ZMF M+R Option, it creates an additional ISPF profile library member.

Name	Function	Description
CDFPROF	M+R ISPF profile	Used to store M+R user variables from the ISPF profile variable pool between ZMF sessions.

The profile name prefix xxxx is set by the NEWAPPL parameter in the ZMF logon CLIST. If prefixes are made unique for each ZMF instance, your profile library will contain up to four members for the first ZMF instance that you use, and up to three more members for each additional ZMF instance you use.

ISPF profile libraries for users of ChangeMan ZMF must be allocated with sufficient primary, secondary, and directory space to allow ZMF to create and update the profile members discussed here. This applies to all ISPF applications that run with ISPF NEWAPPL(xxxx).

REXX Runtime Environment

XMLSERV is an XML page prototype tool delivered with ChangeMan ZMF that you can use to experiment with Serena XML Services.

A REXX runtime environment is required to execute this program. If you have not installed a REXX runtime environment in your z/OS system, and you attempt to execute program XMLSERV, the program will fail with this message:

```
IRX0159E The run time processor EAGRTPRC is not available
```

See the IBM *TSO/E Customization* manual for information about setting up a REXX runtime environment.

Serena ChangeMan ZMF Product Compatibility

The Software Compatibility topic in the ChangeMan ZMF Readme tells you how to identify what versions of Serena and OEM software products are compatible with the base ChangeMan ZMF product.

You can avoid many compatibility problems between ChangeMan ZMF and other Serena products by using JOBLIB and STEPLIB instead of adding Serena load libraries to the LINKLIST.

If you want to run ChangeMan ZMF with other Serena products under one SERNET instance, examine the compatibility chart to see whether there is a potential conflict. Contact Serena Technical Support if you have questions about how to resolve compatibility issues.

CA Panvalet® and CA Librarian® Compatibility

SERNET includes program SERIFACE as an interface to CA Panvalet and CA Librarian. The source for SERIFACE is delivered in the SERCOMC ASMSRC library, but you do not need to customize SERIFACE to use CA Panvalet or CA Librarian with ChangeMan ZMF.

CA Panvalet must be at Release 12 or 14. Missing PTFs can cause problems. If the PAM module has been link-edited with the RENT option, SERIFACE abends with an S0C4. PAM is not reentrant and must be link-edited with the NORENT parameter.

The CA Panvalet or CA Librarian product library must be available for calls from SERNET. If the product library is not in the LINKLIST, then add the library to the STEPLIB in the SERNET started task JCL, the STEPLIB in ChangeMan ZMF cataloged procedures, and your custom JOBLIB skeleton, CMN$$DSN.

At startup, SERNET tries to load modules LIBR for CA Librarian and PAN#1 for CA Panvalet. If the load is successful, the interface to these products is enabled. If you have

renamed program LIBR, use SERNET keyword parameter LIB to tell SERNET the new name. See "LIB" on page 159.

Software Conflicts

If you use any of the software products described in this section, read the information provided to determine if you must take action to prevent conflicts with ChangeMan ZMF or SERNET.

Library Concatenations

SER* modules are delivered with all Serena mainframe products. Use STEPLIB and JOBLIB with Serena mainframe products instead of LINKLIST concatenations to avoid problems with software release compatibility.

Abend-AID® (from Compuware Corporation)

If an abend occurs, Abend-AID generates a formatted dump. However, some SERNET problems cannot be solved without a full dump, so Serena accepts only full dumps when helping you with abends.

If your installation uses Abend-AID and you always want full dumps from SERNET, code the following DD statement in the SERNET started task procedure and any JCL that executes SERNET programs.

```
//ABNLIGNR    DD DUMMY
```

PDSFAST (from Software Engineering of America)

At one time, PDSFAST would enqueue the temporary data set created during a SERCOPY reallocate function, preventing SERCOPY from renaming the data set and finishing the recovery. This problem has been fixed by the vendor.

PDSFAST and SERCOPY contain compression facilities that interfere with each other. Since you cannot suppress the SERCOPY compression facility when SERCOPY is executed by the SERNET started task (parameters are generated internally), you must suppress the compression facility in PDSFAST for copy tasks initiated by SERNET.

Suppress PDSFAST compression for ChangeMan ZMF by adding this DD statement to the SERNET started task procedure:

```
//NPDSFAST  DD  DUMMY
```

CA PDSMAN® (from Computer Associates)

If ChangeMan ZMF manages a very active production load library as a baseline, and CA PDSMAN option ISPF STATS is set to NO, you may receive the following messages from Audit:

```
+IKJ56225I DATA SET PRODT.CMNT.BASE.ACTP.LOD ALREADY IN USE, TRY LATER+
+IKJ56225I DATA SET IS ALLOCATED TO ANOTHER JOB OR USER
```

These messages may indicate that Audit is unable to get a shared enqueue on a baseline library because the CA PDSMAN puts a very brief exclusive enqueue on a library to update directory statistics when a member is accessed. If there is no evidence that another job has an exclusive enqueue on the library, the problem may be resolved by setting CA PDSMAN option ISPF STATS to YES. You may also define a separate library to use as a baseline and use the active library as a ChangeMan ZMF production library.

MAINVIEW® SRM StopX37/II™ (from BMC Software)

StopX37/II intercepts DASD space abends and reallocates data sets. SERCOPY executed from the SERNET started task compresses and reallocates PDS libraries. If both StopX37/II and SERCOPY attempt to fix a space problem, the results are unpredictable.

Suppress StopX37/II for ChangeMan ZMF by coding the following DD statement in the SERNET started task procedure:

```
//X37IGN   DD  DUMMY
```

VSAM Performance Tools

There are several software products that enhance VSAM performance. These products may improve VSAM performance under SERNET and ChangeMan ZMF. However, there may be conditions where SERNET is unable to coexist with these tools.

If program CMNVINIT in install job INITIAL2 hangs in a DETECTED WAIT (DW) state, and there are no messages indicating the cause of the problem, try suppressing any VSAM performance tools that you may have running. These DD statements can be used in step JCL to suppress two popular VSAM performance products:

SST-HiperLoad Plus for VSAM (Performance Essential from Softworks)

```
//PSPOFF  DD  DUMMY
```

HIPER-CACHE™ (from BMC Software)

```
//VVHC$OFF  DD  DUMMY
```

If these DD statements solve the problem, code them in the following places:

- In the SERNET started task procedure.
- In the default file tailoring started procedure CMNxADSP and in any clones that you create and specify in ChangeMan ZMF administration.

Load Library APF Authorization

Some SERNET and ChangeMan ZMF load modules have an Authorization Code of 1 (AC=1). Load libraries containing these modules must be APF authorized. APF authorize all libraries in a JOBLIB or STEPLIB concatenation when one library in the concatenation is APF authorized.

IPL

When you install SERNET and ChangeMan ZMF, IPL your system if you need to accomplish the following:

- APF authorize ChangeMan ZMF load libraries if you only use a static APF list.
- Permanently APF authorize ChangeMan ZMF load libraries if you use a dynamic APF list.
- Pick up new entries in your security system, such as the following in IBM Security Server RACF:
  - Static Class Descriptor Table
  - Started Procedure Table.

NOTE The Serena License Manager gives you a choice of storing licenses for Serena mainframe products in CSA or in a PDS. If you choose to store licenses in CSA, reload the licenses to CSA when you IPL. See the *Serena SER10TY 4.2 User Guide* for information about applying licenses.

Chapter 3
Installation Considerations

This chapter describes issues you must consider before installing ChangeMan ZMF.

NOTE The information in this chapter is provided to help you plan for your installation. Do not execute any installation or configuration tasks until you get to Chapter 4, "Unloading the Software" on page 51 and Chapter 5, "Installing ChangeMan ZMF Components" on page 59.

Upgrading From Earlier Versions	32
System Considerations	32
VSAM Performance	33
VSAM Linear Data Sets	34
SERNET and ChangeMan ZMF JCL	35
Started Procedures for File Tailoring	39
Security Considerations	40
Data Set Considerations	43
Batch Job Name Considerations	46
Staging Versions	48
Customizing ChangeMan ZMF Components	48
Console Log Messages	49

Upgrading From Earlier Versions

See "Choosing the Right Installation/Upgrade Manual" on page 12 to ensure that this *Installation Guide* is the correct manual for your installation or upgrade of ChangeMan ZMF.

System Considerations

This section describes system issues that you must consider before you start the ChangeMan ZMF installation process.

z/OS Subsystem

While each SERNET instance is identified by a "subsystem ID," SERNET is not a formal z/OS subsystem like JES or DB2®; do not define SERNET in the subsystem name table in SYS1.PARMLIB(IEFSSNxx). If you define it in the subsystem name table, SERNET may abend with an S0C4 when it tries to update the subsystem communication vector table with the identifying address space (ASID).

Updating the System Linkage Index

Each SERNET instance uses a system linkage index (a z/OS resource). The system linkage index is not released when a SERNET started task is shut down. However, the next time the same subsystem ID (a SERNET instance identifier) is initialized, it uses the same system linkage index.

The NSYSLX parameter in IEASYSxx defines the number of linkage indexes (in addition to those in the system function table) to be reserved as system linkages. The default number is 55. If your environment has a number of subsystems defined that use system linkage indexes (for example, DB2 and IMS V5), you might need to increase the value of NSYSLX if you define multiple SERNET instances on the same LPAR.

Non-Swappable

The SERNET address space must be available at all times for asynchronous requests coming from client desktops and from other z/OS address spaces. Each SERNET instance makes itself non-swappable by internally issuing the following:

```
SYSEVENT TRANSWAP
```

TRANSWAP is IBM's preferred method of making an address space non-swappable for long periods of time.

Reusable Address Space Identifiers (ASID)

SERNET, ChangeMan ZMF, and all selectable options for ChangeMan ZMF except the Enterprise Release Option (ERO) are compatible with reusable ASIDs. Ensure that reusable ASIDs are enabled on your z/OS system.

For information about ERO and reusable ASIDs, see the *ChangeMan ZMF ERO Getting Started Guide*.

Serena Libraries and LINKLIST

Serena recommends that you do not add load libraries for its mainframe products to the LINKLIST. Instead, include a STEPLIB statement in the JCL for each SERNET instance, and include a JOBLIB or STEPLIB statement in the JCL for each batch job submitted by a Serena product.

STEPLIB and JOBLIB are preferred because:

- If you license more than one Serena product and you do not keep the products at compatible release levels, common Serena load modules in a LINKLIST library might interfere with the proper function of some of these products.
- You should segregate delivered (vendor) versions of load modules in libraries separate from customized programs such as exits. It is easier to maintain the proper concatenation of custom and delivered load libraries if they are in STEPLIB or JOBLIB statements in started procedures and batch JCL.

VSAM Performance

ChangeMan ZMF uses VSAM files to store information about application components and change packages. The ChangeMan ZMF package master, component master, long name component master, activity log, and recovery files are active KSDS VSAM files. This section provides information about enhancing VSAM and ChangeMan ZMF performance.

Defining VSAM file characteristics

Model JCL to define VSAM KSDS files for ChangeMan ZMF includes DEFINE CLUSTER parameter settings that are intended to optimize performance. Serena recommends using the delivered settings for CISZ, FREESPACE, and SHAREOPTIONS.

AMP Parameters

Model JCL for ChangeMan ZMF files in a SERNET started procedure is delivered with AMP parameters to optimize VSAM performance. Serena recommends using the delivered AMP subparameter values for STRNO, BUFND, and BUFNI. These subparameters can be adjusted when you have performance data that suggests different settings.

As delivered, ChangeMan ZMF obtains all VSAM buffers and control blocks above the line (RMODE31=ALL). No GSR or LSR buffer pools are used.

VSAM I/O Optimization of ZMF Master Files

Customers have the option of using BLSR (Batch Local Shared Resources), SMB (System Managed Buffers), or no buffering on their started task JCL. There are sample JCL cards which can be commented in/out to pick the desired method.

SMB only applies to extended VSAM datasets, it has no effect otherwise.

The SMB parameter SMBVSP, can be tweaked to improve performance, ACCBIAS and SMBDFR should not be changed. For more information on tuning SMB, see the IBM manual 'Z/OS DFSMS Using Datasets'.

BLSR is a subsystem of MVS that improves performance by allowing programs that use VSAM non-shared resources (NSR) to use VSAM local shared resources (LSR) without changing application source code or link-editing application load modules.

BLSR can be used with ChangeMan ZMF Version 5 and above. The required JCL changes are included as comments in the model JCL for a ChangeMan ZMF instance.

See the *MVS Programming Batch Local Shared Resources Subsystem Guide* for complete information on the application of BLSR in your environment. Serena strongly recommends that one of the supported VSAM I/O optimization methods is strongly recommended to be enabled as it has shown to dramatically improve VSAM IO performance with the ZMF instance.

VSAM Linear Data Sets

ChangeMan ZMF stores some data in VSAM Linear Data Sets (LDS). An LDS contains data but no control information. IDCAMS is used to define a linear data set. An LDS has only a data component. An LDS is just a physical sequential VSAM data set comprised of 4 KB physical records.

When a ChangeMan ZMF program opens an LDS, the entire file is read into a data space. The data is accessed by the program as a byte-addressable string in virtual storage. When a program updates the data space, the LDS is updated asynchronously by data-in-virtual services.

To a ChangeMan ZMF program, an LDS looks like a table in virtual storage that requires no physical I/O for processing. ChangeMan ZMF employs VSAM LDS files to improve the performance of files that are accessed non-sequentially in an unpredictable pattern. Only programs running under the SERNET/ ChangeMan ZMF started tasks are allowed to dynamically update VSAM LDS files.

In ChangeMan ZMF 8.1, there are two sets of data that are stored in VSAM LDS files:

- An impact analysis LDS contains multiple tables, such as baseline unique number (BUN) data, component identification data, and component relationship data. Impact analysis data is updated as component relationships are added, updated, and deleted, and as library types, applications, and baseline libraries are added and deleted.
- Relationships between XML schemas and DSECTS used for fixed-format control blocks and copybooks are stored in the XMLSPACE LDS. These relationships are static between updates to ChangeMan ZMF software.

LDS Space Utilization

The ChangeMan ZMF impact analysis LDS is defined with extents. The physical space is logically formatted as it is filled with data. You can determine the space utilization of the impact analysis LDS by running an IDCAMS LISTCAT.

The contents of the XMLSPACE LDS do not change between software releases. Space utilization is managed by Serena with changes to the delivered JCL that creates and populates the LDS.

SERNET and ChangeMan ZMF JCL

Expect to run at least two instances of ChangeMan ZMF:

- A production ChangeMan ZMF instance that manages application components in production libraries.
- A test ChangeMan ZMF instance that the ChangeMan ZMF Administrator uses to test upgrades and modifications before they are installed into the libraries running the production ChangeMan ZMF.

Each instance of ChangeMan ZMF runs as an application under a separate SERNET instance. Before building SERNET and ChangeMan ZMF JCL, consider the issues described in the following subsections.

Subsystem ID

Each instance of SERNET is identified by a unique one-character subsystem ID. Valid values for a subsystem ID are:

- Blank (space)
- Numeric 0-9
- Alphabetic A-Z
- Special characters @, #, and $.

NOTE Although a null (blank) subsystem ID is valid, Serena strongly recommends that you avoid using a null subsystem ID.

A subsystem ID is assigned through SERNET keyword option SUBSYS=*subsysID*, which is input to program SERVER.

SERNET Started Task Names

As stated previously, you will have at least two ChangeMan ZMF instances: a test instance and a production instance. You may also have multiple ChangeMan ZMF instances running on other LPARs to manage production libraries.

Each ChangeMan ZMF instance runs under a separate SERNET started task. Each SERNET started task must be assigned a unique identity in z/OS for console commands, automated data center management tools, and SMF. There are three ways to establish a unique z/OS identity for a SERNET started task:

- Member name - Build a separate procedure (member) for each started task. Use only the member name in the START command.

```
S SERPROC1
```

The SERNET started task *jobname* and *identifier* is SERPROC1.

- Identifier - Append an identifier to the procedure member name in the START command.

```
S SERPROC.SERTASK2,ID=2
```

The SERNET started task *jobname* is SERPROC and the *identifier* is SERTASK2.

- Jobname - Use the JOBNAME parameter in the START command.

```
S SERPROC,JOBNAME=SERTASK3,ID=3
```

The SERNET started task *jobname* and *identifier* are both SERTASK3.

If you use a common procedure for several SERNET instances, then you must use an identifier or a JOBNAME parameter in the START command.

NOTE When you assign a started task identity that is different from the started procedure member name, IBM recommends that you use the JOBNAME parameter because it provides an identity that is available to the most z/OS services.

Parameters for SERNET and ChangeMan ZMF

SERNET behavior, and the behavior of applications like ChangeMan ZMF that run under SERNET are controlled by keyword options input to program SERVER.

Passing Parameters to SERNET

Keyword options may be passed to SERNET in two ways:

- In the EXEC statement for program SERVER, as subparameters in the PARM= parameter.

  Example 1:

```
//SERVER  PROC ID=1,OPT='XCH=1234'
//SERVER  EXEC PGM=SERVER,                    *Started Task
//             REGION=0M,                     *Maximum Region
//             DYNAMNBR=200,                  *High allocations
//             PARM='SUBSYS=&ID,&OPT'         *Execution Parms
```

  Example 2:

  Override the SERVER parameters in Example 1 by setting symbolic parameters in the START command.

```
S SERPROC,ID=2,XCH=2345
```

- In a data set read by program SERVER at a DD statement referred to by the keyword option DDNAME=*ddname* coded as a PARM= subparameter.

  Example:

```
//SERVER  PROC
//SERVER  EXEC PGM=SERVER,                    *Started Task
//             REGION=0M,                     *Maximum Region
//             DYNAMNBR=200,                  *High allocations
//             PARM='DDNAME=ANYNAME'          *Execution Parms
. . .
//ANYNAME   DD DSN=SERCOMC.PARMS(SERPARM)
```

  PDS member SERPARM contains:

```
SUBSYS=3        /* SERNET SUBSYS ID
XCH=3456        /* TCP/IP PORT #
```

Keyword Options For ChangeMan ZMF

There are many SERNET keyword options. Options listed in this section are required or are commonly used with ChangeMan ZMF.

See Appendix D, "SERNET Keyword Options" on page 151 for detailed descriptions of the options listed here.

To find other SERNET keyword options that can be used with ChangeMan ZMF, look for "CMN" in the "Application(s)" row of the description tables in Appendix D.

Required Options

These options must be specified for a SERNET instance running ChangeMan ZMF.

Option	Description
SUBSYS	SUBSYS on page 163
CMN=*port* or CMN	apl on page 152

Common Options

You may code these options for a SERNET instance running ChangeMan ZMF depending on the functions you use.

Option	Description
DDNAME	DDNAME on page 156
SDNOTIFY	SDNOTIFY on page 162
TIMEOUT	TIMEOUT on page 163
STAX	STAX on page 162

Special Case Options

These options are used only in special situations.

Option	Description
TCPIP	TCPIP on page 163
XML	XML on page 165
LIB	LIB on page 159

SER#PARM DD Statement

Each SERNET started task creates a reference table of application TCP/IP addresses and port numbers for Serena applications. This reference table is stored in a PDS member named #SERx, where x is the subsystem ID of the SERNET started task.

The information in a #SERx member is used to communicate with a ChangeMan ZMF instance using TCP/IP rather than cross memory services from the following:

- ChangeMan ZMF ISPF client

- ZMF programs in batch jobs
- ZMF File tailoring started procedures

The library containing a #SERx member usually has a DSN low level node of TCPIPORT and is referenced by:

JCL	CLIST
//SER#PARM DD DSN=...	ALLOC DD (SER#PARM) DSN(...

This example shows the format of member #SER1 in a SER#PARM library:

```
* SMFI.SUBS APP DOT.TED.DEC.MAL PORT# TCPIPROC --Update-Time-Stamp--
  BH3A.SER1 XCH DNSA            11111          2008/08/15 @ 10:30:59
  BH3A.SER1 CMN DNSA            22222          2008/10/19 @ 13:31:42
* END OF DATA
```

SERNET creates a #SERx member from the location of the started task (SMFID, DNS name) and from SERNET keyword parameters (SUBSYS=x, apl=port). SERNET cannot update an existing #SERx member. If you want to refresh the information in a #SERx member, SHUTDOWN the started task where the member was created, delete the member from the SER#PARM library, and START the procedure to recreate the member.

NOTE Prior to ChangeMan ZMF 7.1.3, #SERx members were generated with dotted decimal IP addressees. Starting with ZMF 7.1.3, they are generated with DNS names of 16 characters or less. The maximum length of a DNS name will be extended in future ZMF versions.

The SER#PARM DD statement is required in the JCL for any started task that runs ChangeMan ZMF, even if you do not intend to use TCP/IP for ZMF communication.

If there is a TCP/IP address space on the LPAR where SERNET runs, SERNET automatically creates a #SERx member in the SER#PARM file. If there is no TCP/IP address space on the same image, you must manually create a member in this file and manually code a dummy table entry. Instructions for manually creating this member are provided in Chapter 5, "Installing ChangeMan ZMF Components".

CAUTION! A library containing #SERx members can be shared by multiple SERNET started tasks as long as those started tasks have unique subsystem IDs. However, do not use a SER#PARM library for any other purpose, such as passing SERNET keyword parameters to a started task. SERNET opens this library for output, which can interfere with other uses of the file.

SERLIC DD Statement

The SER10TY Serena License Manager gives you a choice of storing licenses for Serena mainframe products in CSA or in a PDS.

If you store licenses in a PDS, that library must be named in DD statement SERLIC included in:

- Started procedures that run ChangeMan ZMF
- CLISTs you use to connect to ChangeMan ZMF instances

See the *Serena SER10TY 4.2 User Guide* for information about applying licenses.

SYSMDUMP DD Statement

The preferred means of gathering diagnostic information for a program interrupt in a SERNET started task is through a data set allocated to a SYSMDUMP DD statement. The data set should have these attributes:

```
//SYSMDUMP  DD DISP=(MOD,CATLG,CATLG),             * SYSMDUMP
//             DSN=somnode.SERCOMC.SYSMDUMP(+1),
//             UNIT=SYSDA,SPACE=(CYL,(200,100),RLSE),
//             DCB=(DSORG=PS,RECFM=FBS,LRECL=4160,BLKSIZE=4160)
```

Serena recommends that you define a GDG index for the SYSMDUMP data set to prevent diagnostic information in the data set from being overwritten when the SERNET instance is restarted after an abend.

SYSTCPD DD Statement

If you use TCP/IP to communicate with ChangeMan ZMF and there are multiple TCP/IP started tasks running on the same LPAR, you may need to code DD name SYSTCPD in the SERNET started task JCL. See topic "Considerations for Multiple Instances of TCP/IP" in the IBM publication *z/OS Communications Server IP Configuration Guide*.

Started Procedures for File Tailoring

A SERNET started task running ChangeMan ZMF initiates started procedures to perform ISPF file tailoring to create JCL for the following functions:

- Package installation
- Stage
- Promotion
- Other batch activities

These procedures run like started tasks in their own address space. They are not regular batch jobs and they do not require an initiator. They run only as long as it takes to perform file tailoring for the ChangeMan ZMF function that started them, and then they terminate. Multiple copies of these procedures may be started, and multiple started tasks may execute at the same time.

The default name for these started procedures is CMNxADSP, where x is the subsystem ID of the of the SERNET started task where ChangeMan runs. However, up to four separate file tailoring started procedure names may be specified in ChangeMan ZMF administration, one for each of the four file tailoring functions listed above.

Instructions for creating and implementing the default file tailoring started procedure CMNxADSP are provided in subsequent topics and chapters in this manual. As your requirements become clear later in your implementation of ChangeMan ZMF, you can clone procedure CMNxADSP to create additional started procedures, repeating the installation steps executed for CMNxADSP, and then enter the name of the new procedures in ChangeMan ZMF administration.

Security Considerations

ChangeMan ZMF maintains the integrity of production components by managing production libraries and all libraries that contain components in any stage of development.

ChangeMan ZMF answers the questions, "What is production?" by including all development and test environments within the controls traditionally applied only to production execution libraries and source libraries. These controls include:

- Physical access to development and production libraries.
- Access to the automated change management functions that change development and production libraries.

Exclusive Access to Libraries

To exercise such broad control, ChangeMan ZMF must have:

- Exclusive update access to baseline (production) libraries
- Exclusive update access to libraries in controlled test (promotion) environments
- Exclusive create, update, and scratch access to ChangeMan ZMF staging libraries that are dynamically allocated for each change package
- Read access to all system libraries involved in the development process.

All of this data set access is granted to the SERNET started task where ChangeMan ZMF runs. ChangeMan ZMF performs on-line functions using this security access, and it submits batch jobs that inherit the security access of the started task, even when the jobs run with the user's TSO ID as the job name.

Access to ChangeMan ZMF Functions

Once exclusive update access to production and development libraries has been granted to ChangeMan ZMF, the components in these libraries are protected by limiting access to the ChangeMan ZMF functions that can change these libraries.

These restrictions are defined in your security system.

- You define a new resource class for ChangeMan ZMF.
- You define security entities under that resource class that represent ChangeMan ZMF functions.
- You define security entities that represent applications in ChangeMan ZMF.
- You grant access for each security entity to individual TSO IDs or to groups.

When someone attempts to use a ChangeMan ZMF function, ChangeMan ZMF queries the security system with the name of the security entity for that function and the TSO ID of the user.

- If your security system determines that the TSO ID has sufficient access to the security entity, ChangeMan ZMF permits the user to execute the function.
- If the TSO ID does not have sufficient access, ChangeMan prohibits execution of the function.

Administrator and Change Manager Security Entities

Five security entities have fixed format names and control administrator and change manager functions. These security entities are always required for a ChangeMan ZMF implementation.

NOTE Some ChangeMan ZMF selectable options require additional security entities. If you license a selectable option, see the *Getting Started Guide* for that option.

Security Entity	ChangeMan ZMF Function
CMNGBADM or CMNxGBAD	Configure/change global administration
CMNLCADM or CMNxLCAD	Configure/change application administration
CMNREVRT or CMNxREVR	Revert packages to DEV status after the approval process has started
CMNBKOUT or CMNxBKOU	Back out packages in BAS or INS status
CMNMON or CMNxMON	Monitor failed package installs or packages in the ChangeMan ZMF scheduler

The x imbedded in the security entity name represents the one-character subsystem ID of the SERNET started task where ChangeMan runs. A security entity with an embedded subsystem ID controls authority only for the ChangeMan ZMF instance with that subsystem ID.

Default Security Entities

Normally, ChangeMan ZMF searches for the subsystem-specific entity with the embedded subsystem ID, and if that entity is not found, the search looks for the entity without the subsystem ID.

For example, the search for the global administrator entity for ChangeMan ZMF running under subsystem ID 3 is:

1 CMN3GBAD, and if not found, then...

2 CMNGBADM

Implicitly, the security entities without the embedded subsystem ID become the default set that are used when you have not defined subsystem-specific entities.

CAUTION! If you use a null (blank) subsystem ID, the security entities for that ChangeMan ZMF instance become the default entities for all other instances. That is one reason why Serena strongly recommends that you avoid using a null subsystem ID.

Mandatory System-Specific Security Entities

If your security administrator uses one of the following parameters in your security system, and the ChangeMan ZMF subsystem-specific security entity is not found, there is no search for the default entity, and authorization is denied.

Security System	Parameter
RACF	PROTECTALL
CA ACF2	MODE(ABORT)
CA Top Secret	MODE(FAIL)

If one of these parameters is used, you must create a set of system-specific security entities with embedded subsystem ID for each ChangeMan ZMF instance.

Security Entities and Required Authority

This table lists the security entities that control execution of ChangeMan ZMF functions, and it shows the level of security access that is required to execute each function.

NOTE Some ChangeMan ZMF selectable options require additional security entities. If you license a selectable option, see the *Getting Started Guide* for that option.

The first five rows in the table describe five security entities with fixed format names. To grant authority to execute other functions, you define your own security entity name in your security system, and then associate that entity name with a ChangeMan ZMF function by making an entry in ChangeMan ZMF administration.

ChangeMan ZMF Function	Primary Authorization	Additional Authorization
Configure/change global administration	UPDATE access to entity CMNGBADM or CMNxGBAD	None
Browse global administration	READ access to entity CMNGBADM or CMNxGBAD	None
Configure/change application administration	UPDATE access to entity CMNLCADM or CMNxLCAD	UPDATE access to the application entity you define
Browse application administration	READ access to entity CMNLCADM or CMNxLCAD	READ access to the application entity you browse
Revert packages to DEV status after the approval process has started	UPDATE access to entity CMNREVRT or CMNxREVR	READ access to the application entity you define
Back out packages in BAS or INS status	UPDATE access to entity CMNBKOUT or CMNxBKOU	READ access to the application entity you define
Monitor failed package installs or packages in the Change Man ZMF scheduler	UPDATE access to entity CMNMON or CMNxMON	READ access to the application entity you define
Query packages, browse components in baseline and promotion	READ access to the application entity you define	None
Create packages, work with package components	UPDATE access to the application entity you define	None

ChangeMan ZMF Function	Primary Authorization	Additional Authorization
Promote/demote packages to a site/level	UPDATE access to the promotion entity you define for the site/level	READ access to the application entity you define
Approve/reject packages	UPDATE access to an approval entity you define	READ access to the application entity you define
Selectively unfreeze/ refreeze components in frozen package	UPDATE access to the next approval entity	UPDATE access to the application entity you define

When you define a ChangeMan ZMF security entity in your security system, you define it with no universal access. When you grant a TSO ID or group access to the security entity, you give them READ or UPDATE authority.

SAF and Your Security System

You define security rules and authorizations for ChangeMan ZMF users in your security system. ChangeMan ZMF is compatible with IBM Security Server RACF, CA ACF2, and CA Top Secret.

SAF is an acronym for System Authorization Facility, an interface defined by z/OS that enables programs to use system authorization services to protect access to resources such as data sets and system commands. SAF provides a common interface for IBM Security Server RACF, CA ACF2, and CA Top Secret where you define the security rules for an LPAR.

Security for File Tailoring Started Procedures

Special security definitions are required for file tailoring started procedure CMNxADSP and any clones you may create and specify in ChangeMan ZMF administration.

These definitions are detailed in Chapter 6, "Configuring Security".

Access to TCP/IP Functions

Access to TCP/IP Services in z/OS Communications Server requires a z/OS UNIX security context, referred to as an OMVS segment, for the user ID associated with a SERNET instance.

See the section “Requirement for an OMVS Segment” in the IBM publication *z/OS Communications Server: IP Configuration Guide*.

Additionally, RACF PassTickets are a requirement for mainframe clients (not ChangeMan ZDD or ChangeMan ZMF for Eclipse) connecting via TCP/IP. Instructions for generating RACF PassTickets are detailed in Chapter 6, “Configuring Security”.

Data Set Considerations

ChangeMan ZMF uses several data set types that impose unique requirements on your DASD management tools and/or your data set security rules.

Temporary List Data Sets

ChangeMan ZMF stage jobs write SYSOUT data sets to z/OS temporary files named &&LIST*. At the end of the job, utility program SERPRINT reads these temporary files, combines them into a single file, writes the file in compressed format to a package staging library member, and then writes this same information to SYSOUT at DD name PRINT2 in eye-readable format.

Program SERPRINT cannot find &&LIST* files that are written to virtual I/O (VIO) files because these files exist only in paging storage.

In ChangeMan ZMF administration, you can specify an esoteric device group or generic device type that applies only to &&LIST* data sets. Consult with your storage administrator or system programmer to identify an esoteric device group or generic type that is ineligible for VIO and would be appropriate for &&LIST* data sets.

Utility Data Sets

ChangeMan ZMF uses "temporary" cataloged utility data sets when performing functions such as:

- Expanding compressed listings for the Browse Listing function
- ISPF file tailoring to create job JCL from ChangeMan ZMF skeletons
- Opening package components for edit (You edit the component in a utility data set rather than in the staging library member.)

ChangeMan ZMF utility data sets are created by:

- SERNET started tasks running ChangeMan ZMF
- File tailoring started procedures initiated by ChangeMan ZMF
- ChangeMan ZMF users (automatically) as they connect to ChangeMan ZMF through the ISPF interface

ChangeMan ZMF deletes the utility data set when the function is completed.

The same data set name format is used for all ChangeMan ZMF utility data sets. The DSN format for ChangeMan ZMF utility data sets is defined in exit program CMNEXINS. Comments in the source code for CMNEXINS describe the default DSN format.

Utility Data Set Security

When you first install ChangeMan ZMF, you must:

- Choose a data set naming convention for ChangeMan ZMF utility data sets.
- Code that DSN format in exit program CMNEXINS.
- Define rules in your security system that grant ALTER/CREATE access to:
  - SERNET started tasks running ChangeMan ZMF
  - File tailoring started procedures initiated by ChangeMan ZMF
  - Users who access ChangeMan ZMF though the ISPF interface

In some data centers, there is an existing high-level qualifier like TEST that has universal ALTER/CREATE security access.

If you do not have such a high-level qualifier already defined, work with your DASD manager and security administrator to set up a high-level qualifier that ChangeMan ZMF can use for utility data sets.

TIP Comments in the source for exit program CMNEXINS describe several examples that can help you choose a high-level qualifier.

DASD Management for Utility Data Sets

ChangeMan ZMF deletes utility data sets when users exit the function that created the data set. When a ChangeMan ZMF function or session does not end normally, utility data sets are not deleted. For example, if your TSO session times out when you are editing a package component, the utility data set allocated for your edit session is not deleted.

You can create an ACS routine for DFHSMS to delete old utility data sets based on the naming convention code in exit program CMNEXINS. However, you should not let DFHSMS delete utility data sets while they can still be used to recover edit changes from an interrupted edit-in-stage session.

For example, if your TSO session times out while you are editing a ChangeMan ZMF package component on a Friday afternoon, you can recover your edit changes after the weekend if the utility data set you were editing is still cataloged. (See topics "Automatic Edit Recovery" and "Manual Edit Recovery" in the *ChangeMan ZMF User's Guide*.)

Staging Library Model Data Set Name

Your global administrator defines a Staging Library Model Data Set Name in Global Administration Parameters. This model is used to create data set names when ChangeMan ZMF allocates these data sets for a change package:

- Package staging libraries
- Dot X libraries for package installation job JCL
- PACKAGE data set for package installation processing at ZMF production instances
- Dot B data set for package blackout processing in a ZMF DP/P environment
- Dot R data set for package revert processing in a ZMF DP/P environment

The Staging Library Model Data Set Name must include:

- A three- or four-character node for application mnemonic
- A seven-character node for the package number preceded by #
- A node to distinguish development staging libraries from production staging libraries that are used to install a package at a remote site

The longest library name that is constructed using the minimal staging library model data set name is:

```
...aaaa.#nnnnnn.e.X.ssssssss
```

where:

aaaa is the 3-4 character application mnemonic

nnnnnn is the six digit package number

e indicates whether the data set is a development or production staging library

X is a literal that denotes a library containing package installation JCL

ssssssss is a 1-8 byte site name

ChangeMan ZMF allocates these data sets under the authority of the user ID for the SERNET started task that runs ChangeMan ZMF. ChangeMan ZMF automatically reallocates these data sets if they run out of extents or directory space, and the data sets are deleted by a housekeeping job submitted by the started task.

You can add a high level qualifier and other nodes to the Staging Library Model Data Set Name as long as the longest name does not exceed the 44 character maximum. Define a Staging Model Data Set Name that allows you to satisfy these security access requirements and DASD management requirements:

- The SERNET started task must have exclusive ALTER/CREATE, UPDATE, and DELETE access to data sets created with the Staging Model Data Set Name.
- DFHSM can migrate these data sets to a secondary storage format or media. SERNET will issue an HRECALL command when it needs a data set that has been migrated, or it will issue an HDELETE if it wants to delete a migrated data set.
- Data sets created with the Staging Model Data Set Name should not be compressed unless the development and production instances of ChangeMan ZMF are down and no ChangeMan ZMF installation or promotion jobs are running.

Batch Job Name Considerations

ChangeMan ZMF submits batch jobs to perform functions like component stage (build), package promote, and package install. These jobs may manipulate components in libraries under ChangeMan ZMF control and update ChangeMan ZMF master files.

For these batch jobs submitted by ChangeMan ZMF:

- The job owner is the user ID of the SERNET started task where ChangeMan ZMF runs.
- The job runs with the security authority of the started task user ID.

Job names and other JOB statement information depend on whose behalf the job is run.

User Job Names

ChangeMan ZMF submits batch jobs on behalf of ChangeMan ZMF users who work with packages and components.

For example, when a user stages a like-source component, ChangeMan ZMF builds stage job JCL from ISPF skeletons and submits a job to compile and link edit the component, update the component status and component history, and put newly generated components into package staging libraries.

The user enters JOB Statement Information, including a job name, on an ISPF panel in the online process that initiates batch processing.

You can validate or change the JOB statement information entered by users for jobs submitted by ChangeMan ZMF on their behalf by customizing exit program CMNEX008.

Install Job Names

ChangeMan ZMF submits a series of batch jobs to install and baseline ripple a package. Even if you use an external job scheduler like CA 7® Workload Automation or CA Scheduler® Job Management to submit the first install job for a change package, ChangeMan ZMF submits all following install jobs for the package.

ChangeMan ZMF builds JCL for install jobs from ISPF skeletons. JOB statement information for these jobs, except for job name, is taken from an ISPF panel in Application Administration. The job name is constructed as follows:

For an application with a four-character application mnemonic, the install job name is:

```
//aaaattnn
```

where

```
aaaa = application name
tt   = transaction code
nn   = last 2 characters of package number
```

For an application with a three-character application mnemonic, the install job name is:

```
//aaattnnn
```

where

```
aaa  = application name
tt   = transaction code
nnn  = last 3 characters of package number
```

You can validate or change the job name for install jobs submitted by ChangeMan ZMF by customizing exit program CMNEX008.

Viewing Job SYSOUT

ChangeMan ZMF users will want to view the sysout from jobs submitted on their behalf by ChangeMan ZMF, and they need authority to purge such job output. Developers, project managers, change control staff, and operations staff may need to view sysout from ChangeMan ZMF install jobs.

You may have to make adjustments to the following to provide the necessary access to sysout from jobs submitted by ChangeMan ZMF:

- Job access rules in your sysout management tool
- Job access rules in your security system
- JES rules for access to job output
- Job name overrides in ChangeMan ZMF exit program CMNEX008

Staging Versions

The staging versions facility can save an unlimited number of versions of a package component that a developer might create in a staging library between the time the component is first added to the package and the time the package is baselined.

Staging versions are stored as full copies in a compressed format in a VSAM file. Each version may be labeled with an optional 35-character description.

Some features of staging versions are available if the staging versions facility is installed by defining three VSAM files and by including the HPSPLIB and HPSIN DD statements in the SERNET started procedure.

All features of staging versions are available for package components in a library type if the staging version facility is installed, and if the application administrator enables staging versions for the library type.

All staging versions processing is bypassed if the HPSPLIB and HPSIN DD statements are deleted from the SERNET started procedure.

Issues to consider when deciding whether to install the staging versions facility:

- If the staging versions facility is installed, ChangeMan ZMF reads one of the staging version VSAM files whenever a user performs a function that replaces a member in a staging library. This VSAM read might have an adverse effect on ChangeMan ZMF response time.
- If the staging versions facility is installed *and enabled* for a library type, ChangeMan ZMF writes to the staging version VSAM files whenever a user saves a staging version. This VSAM write might have an adverse effect on ChangeMan ZMF response time.

For a full description of the Staging Versions facility, see topic “Staging Versions” in Chapter 3 "Pre-Implementation Decisions" in the *ChangeMan ZMF Administrator Guide*.

Customizing ChangeMan ZMF Components

ChangeMan ZMF is designed to be flexible so that it can serve customers with different change management processes and different data center standards. Within the constraints of best practices for change management, you can adapt ChangeMan ZMF to your environment using:

- Configuration in Global and Application Administration
- Modifications to ISPF skeletons that are file tailored into JCL for batch processes
- Code in exit programs that are provided by Serena

Even if you do not need to customize ChangeMan ZMF components to fit your change management practices, you must modify some components to fit your local data set naming conventions.

When you modify ChangeMan ZMF components, preserve the original components delivered from Serena. Never edit components in the libraries you unloaded from the ZMF installer, and never link edit load modules into the delivered libraries.

Allocate a separate set of custom libraries for ChangeMan ZMF components that you change. Copy the component you want to modify from the delivered library into your custom library, then make you changes in the custom library.

"Step 1: Allocate CUSTOM Libraries" on page 60 contains a sample list of delivered and custom libraries.

When you modify a component delivered from Serena, keep the name of the custom component the same as the delivered component so that you can use a file compare tool or code merge tool to help you reapply your modifications for a ChangeMan ZMF upgrade.

When you create an original custom component for ChangeMan ZMF, use an abbreviation or acronym for your company or organization in the first three characters of the component name to differentiate your component from components delivered from Serena.

Console Log Messages

SERNET and ChangeMan ZMF write certain messages to the console log (WTO) to make them available to your automated operations tools. You can configure your automated operations tools to recognize the messages and issue notifications or execute remedial tasks.

Chapter 4

Unloading the Software

This chapter lists tasks you perform to unload ChangeMan ZMF components from the ZMF installer and create libraries on the mainframe.

Introduction	52
Step 1: Unload the ChangeMan ZMF Files	53
Step 2: Transfer Files to the Mainframe	55
Step 3: Expand PDS Libraries with RECEIVE	57

Introduction

Components for ChangeMan ZMF are delivered in compressed binary format in a ZMF installer. This chapter tells you how to unload files from the installer to your workstation or network drive, copy component files to the mainframe, and expand those files into PDS libraries.

In this chapter:

v = Version number for ChangeMan ZMF
r = Release number
m = Maintenance release number

ChangeMan ZMF Libraries

The libraries installed by the ZMF installer contain all of the components necessary to install and run the ChangeMan ZMF base product and all selectable options.

This table shows the libraries that are unloaded from the ZMF installer and created on the host.

Libraries	Library Contents
CMNZMF.VvRrMm.ASMCPY	ChangeMan ZMF copybooks for exit programs
CMNZMF.VvRrMm.ASMSRC	ChangeMan ZMF source for exit programs
CMNZMF.VvRrMm.CLIST	CLIST to connect to ChangeMan ZMF
CMNZMF.VvRrMm.CNTL	ChangeMan ZMF JCL
CMNZMF.VvRrMm.DBRMLIB	ChangeMan ZMF DB2 Option and ERO DBRM
CMNZMF.VvRrMm.ERR	ChangeMan ZMF ERO Release Audit error descriptions
CMNZMF.VvRrMm.LOAD	ChangeMan ZMF load modules
CMNZMF.VvRrMm.MESSAGES	ChangeMan ZMF ISPF messages
CMNZMF.VvRrMm.PANELS	ChangeMan ZMF ISPF panels
CMNZMF.VvRrMm.REX	ChangeMan ZMF REXX program source
CMNZMF.VvRrMm.SAMPLES	ChangeMan ZMF samples
CMNZMF.VvRrMm.SKELS	ChangeMan ZMF ISPF skeletons
SERCOMC.VvRrMm.ASMCPY	SERNET/Common/SER10TY exit copybooks
SERCOMC.VvRrMm.ASMSRC	SERNET/Common/SER10TY exit source
SERCOMC.VvRrMm.CEXEC	SERNET/Common/SER10TY compiled REXX modules
SERCOMC.VvRrMm.CLIST	SERNET/Common/SER10TY CLIST
SERCOMC.VvRrMm.CNTL	SERNET/Common/SER10TY JCL
SERCOMC.VvRrMm.LOAD	SERNET/Common/SER10TY load modules
SERCOMC.VvRrMm.PANELS	SERNET/Common/SER10TY ISPF panels
SERCOMC.VvRrMm.SAMPXML	ChangeMan ZDD XML parameters for ChangeMan ZMF
SERCOMC.VvRrMm.SKELS	SERNET/Common/SER10TY ISPF XML page models
SERCOMC.VvRrMm.XML	SER10TY NAMES XML page

Documentation

ChangeMan ZMF documentation is available for download at the Serena Customer Support website. See "Accessing the Documentation" in the *ChangeMan ZMF Readme*.

Unload Process

The automated and manual steps in the unload procedure perform the following actions:

- Copy and decompress files from the ZMF installer to a work station or network drive.
- Build an FTP command file.
- Build JCL on the work station or network drive for a mainframe job to execute TSO RECEIVE commands.
- Edit the FTP command file.
- Execute FTP with the command file to transfer files from the work station or network drive to the host mainframe computer.
- Edit the RECEIVE job JCL on the mainframe.
- Submit the RECEIVE job on the mainframe to expand the binary XMIT files into PDS libraries.

Requirements

To execute the unload process you need:

- TCP/IP FTP connectivity from your work station to the mainframe computer
- The dotted decimal IP address or domain name of the LPAR on the mainframe
- A mainframe user ID and password
- Security authorization to allocate the required files on the mainframe

Step 1: Unload the ChangeMan ZMF Files

Execute this procedure to copy files from the ZMF installer to a work station or network drive, build an FTP command file, and create JCL for a mainframe job to execute TSO RECEIVE commands.

1 Download **ZMFvrmSetup.exe** from the Support tab on the Serena Software web site to any directory on your workstation. Double-click **ZMFvrmSetup.exe**.

2 If this version of ChangeMan ZMF is already installed on this workstation, the Serena **ChangeMan ZMF v.r.m Removal** dialog box is displayed.

   a Click **Remove** on this dialog box and the next dialog box.

   b Click **Finish** on the **Completed the Serena ChangeMan ZMF v.r.m Setup Wizard** dialog box.

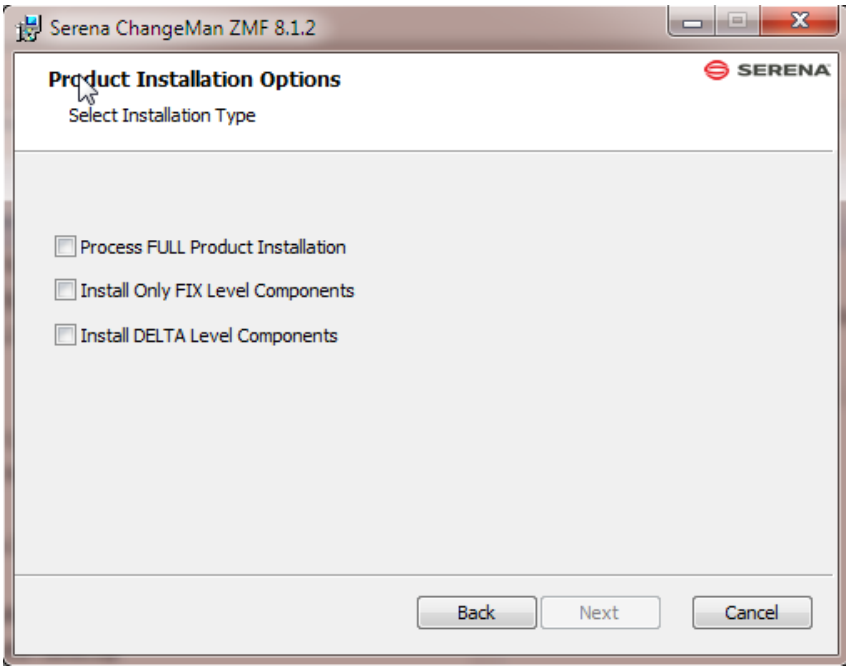
   c Restart the unload procedure from #1 on page 53.

3 Read the text in the **End-User License Agreement** scroll box. If you agree to the terms stated in the End-User License Agreement, check "I accept the terms in the License Agreement" and click **Next**.

NOTE Two or three installation options, (FULL, FIX and DELTA) may be provided when executing the installer. Refer to the Readme section entitled 'installation' to decide which best suits your specific requirements. See "Updating Delivered Libraries" on page 146 to help you decide which option to choose.

The example continuing below is from the 8.1.2 installer.

After checking your choice, click **Next.**



4 On the Installation Options **- Now Confirming Installation...** dialog, click **Next** to accept the default folder, or click **Change Installation Target...** to select a different folder, then click **OK** and **Next**.

5 On the **Host FTP and Transmission Information** dialog box, type the following information:

- Host Name or IP - The dotted decimal IP address or DNS name of the mainframe host
- Host Logon ID - Your host logon (TSO) ID
- Transmitted File HLQ - High level qualifiers for the host data set names of the binary XMIT files transmitted from your workstation
- PDS Library HLQ - High level qualifiers for the host data set names of the PDS libraries expanded via RECEIVE from the binary XMIT files
- HFS Directory - Directory for the Jython script that ZMF uses to deploy Java applications to WebSphere on z/OS. If you do not intend to use the Jython script, you can leave the default directory your/hfs/directory.

NOTE Follow these rules when setting high level qualifiers:

- Your mainframe user ID must have authority to allocate files with the HLQ you specify.
- Do not enter leading or trailing periods.
- Do not enter parenthesis or quotes (single or double).
- Do not specify an HLQ longer than 17 characters, including embedded periods.
- An HLQ entered in this dialog box is not case sensitive, even though it will be embedded in the RECEIVE job that is run on the host.

CAUTION! Do not choose high level qualifiers that will create data set names that already exist on the mainframe. The transmit process and the RECEIVE job will overlay existing data sets.

Click **Next**.

6 On the **Ready to install Serena ChangeMan ZMF v.r.m** dialog box, click **Install**.

7 The **Setup Status** dialog box is displayed as files are decompressed and copied to your desktop or network destination folder, the FTP command file is built, and JCL for the mainframe RECEIVE job is created.

8 When the unload process is completed, the **Completed the Serena ChangeMan ZMF v.r.m Setup Wizard** dialog box is displayed. Check the "Open Read Me for ChangeMan ZMF v.r.m", click **Finish**.

Step 2: Transfer Files to the Mainframe

Your work station must be connected to the network to transfer files to the host

Execute this procedure to transfer ChangeMan ZMF files from your work station or network drive to the mainframe using FTP.

NOTE If you want to use 3270 emulator software instead of FTP to transfer the files to the host, go to Appendix A, "Manual File Transfer and Expand" on page 135.

1 On the **Start** menu on your work station, go to **All Programs > Serena > ChangeMan ZMF v.r.m > Edit FTP Input**. This will open a Notepad window and display the FTP command file used to transfer the ChangeMan ZMF XMIT format files to the mainframe.

2 Overtype **<PASSWORD>** in the third line of the file with your mainframe password, making sure that the case matches what z/OS expects.

3 If you do not intend to use the Jython script to deploy Java applications to WebSphere on z/OS, you can delete the **put** command for the **JYTHON** file at the bottom.

4 Save the file under the same name.

5 Initiate the FTP transfer from the **Start** menu at **Programs > Serena > ChangeMan ZMF v.r.m > FTP files to host.**
(this executes FTP.bat).

When the transfer starts, a command prompt window opens and shows:

```
FTP process has begun.
```

When the transfer is completed, the command prompt window shows:

```
FTP process has completed.
FTP Execution Log: ...
Press any key to continue . . .
```

6 View the **Ftp.log** file using the path displayed in the command prompt window.

Verify that the files were transferred to the mainframe host.

If you see the following errors in the FTP Log, make the recommended corrections to the **FTP.Input** file, then restart the process at #4 above.

Message	Corrective Action
Unknown host...	Edit the FTP.Input file and verify the IP address or domain name of your mainframe host.
530 PASS command failed	Edit the FTP.Input file and verify that you entered the correct user ID and password.

Block size specified for the host [binary] data set is invalid for the record length
Record length... invalid.
	Edit the FTP.Input file:
	a Delete these four command lines:
	`literal site cylinders`
	`literal site blocksi=3120`
	`literal site lr=80`
	`literal site recfm=fb`
	b Replace the deleted lines with this command (on a single line):
	`literal site recfm=fb lr=80 blocksi=3120 cylinders`

Not connected	The FTP connection was lost so files were not transferred. Reconnect and restart the FTP process.

7 Remove your mainframe logon password in the FTP command file:

a On the **Start** menu, go to **Programs > Serena > ChangeMan ZMF v.r.m > FTP.Input**.

b Overtype your password in the third line with **<PASSWORD>**.

c Save the file under the same name.

NOTE If you have difficulty with this automated FTP process, you can transfer ChangeMan ZMF files manually from your work station to the mainframe and expand the files into PDS libraries with the manual procedure in Appendix A, "Manual File Transfer and Expand" on page 135.

Step 3: Expand PDS Libraries with RECEIVE

The Setup Wizard process creates a batch mainframe job to execute the RECEIVE command on binary XMIT files transferred to the host. The RECEIVE job is transferred to the mainframe along with the binary XMIT files.

Execute this procedure to submit the RECEIVE job on the mainframe to expand the ChangeMan ZMF component files into PDS libraries.

1. Log on to the mainframe host.
2. Edit the file named *node*.RECEIVE.ZMF*vrm*.JCL, where *node* is the Transmitted File HLQ you entered in the Setup Wizard.
3. Replace the four //JOBCARD lines at the top of the file with your JOB statements.
4. Make global changes to the DATASET parameter of the RECEIVE commands to change the low-level nodes in the library data set names to match your local requirements. Retain the last node name. For example:

   Original

   ```
   //SYSTSIN  DD    *
     RECEIVE INDATASET('hlq1.CMNZMF.V8R1M3.ASMCPY')
     DATASET('hlq2.CMNZMF.V8R1M3.ASMCPY')
   ```

   After change

   ```
   //SYSTSIN  DD    *
     RECEIVE INDATASET('hlq1.CMNZMF.V8R1M3.ASMCPY')
     DATASET('hlq2.CMN813.ASMCPY')
   ```

5. If you did not transfer the JYTHON file from your workstation to the mainframe because you do not intend to use the Jython script, delete job step JYTHON.
6. Submit the job.
7. Examine each of the output PDS libraries created with the host high level qualifier to ensure that they are PDS libraries.

NOTE The RECEIVE job can give a Return Code 00, and the job SYSOUT can show the message "Restore successful to data set...", even when the RECEIVE process did not complete successfully.

8. Save the RECEIVE job JCL.

Chapter 5

Installing ChangeMan ZMF Components

This chapter lists tasks you perform to install Serena® ChangeMan® ZMF.

CAUTION! Do not execute the steps in this chapter to upgrade an existing ChangeMan ZMF instance to the current version. If you run the jobs described in this chapter and use existing ChangeMan ZMF data set names, you will destroy the data in those files. Use the appropriate *ChangeMan ZMF Migration Guide* to upgrade a ChangeMan ZMF instance to the current version.

Step 1: Allocate CUSTOM Libraries	60
Step 2: APF Authorize Delivered and CUSTOM Load Libraries	60
Step 3: Apply ChangeMan ZMF License	61
Step 4: Define ChangeMan ZMF VSAM Files	61
Step 5: Define XMLSPACE VSAM LDS	63
Step 6: Define Staging Versions VSAM Files	65
Step 7: Store SVPARM Control Member	69
Step 8: Allocate the SER#PARM TCP/IP Address Library	70
Step 9: Create ISPF Keylist for EXPAND	70
Step 10: Build SERNET JCL for ChangeMan ZMF	72
Step 11: Build Default File Tailoring Procedure	75
Step 12: Build the Logon CLIST	76
Step 13: Modify Exit Program CMNEXINS	77
Step 14: APF Authorize Other Load Libraries	78

Step 1: Allocate CUSTOM Libraries

Allocate a set of libraries for customized ChangeMan ZMF components so you can preserved delivered components. See "Customizing ChangeMan ZMF Components" on page 48.

Recommendations for allocating CUSTOM libraries:

- Choose data set names that comply with your data center standards.
- Use the same RECFM and LRECL as the corresponding delivered library.
- For simplicity, allocate SPACE=(CYL,(1,1,50)).
- You may optimize the BLKSIZE parameter.
- It is not necessary to allocate a SERCOMC CUSTOM CNTL library. You can keep all customized JCL in the CMNZMF CUSTOM CNTL library.
- It is not necessary to allocate a SERCOMC CUSTOM CLIST library. You can keep all customized CLISTs in the CMNZMF CUSTOM CLIST library.

1 Allocate CUSTOM libraries for the delivered libraries shown in this table.

Delivered Libraries	Custom Libraries
...CMNZMF.VvRrMm.ASMSRC	...CMNZMF.VvRrMm.CUSTOM.ASMSRC
...CMNZMF.VvRrMm.CLIST	...CMNZMF.VvRrMm.CUSTOM.CLIST
...CMNZMF.VvRrMm.CNTL	...CMNZMF.VvRrMm.CUSTOM.CNTL
...CMNZMF.VvRrMm.LOAD	...CMNZMF.VvRrMm.CUSTOM.LOAD
...CMNZMF.VvRrMm.MESSAGES	...CMNZMF.VvRrMm.CUSTOM.MESSAGES
...CMNZMF.VvRrMm.PANELS	...CMNZMF.VvRrMm.CUSTOM.PANELS
...CMNZMF.VvRrMm.REX	...CMNZMF.VvRrMm.CUSTOM.REX
...CMNZMF.VvRrMm.SKELS	...CMNZMF.VvRrMm.CUSTOM.SKELS
...SERCOMC.VvRrMm.ASMSRC	...SERCOMC.VvRrMm.CUSTOM.ASMSRC
...SERCOMC.VvRrMm.LOAD	...SERCOMC.VvRrMm.CUSTOM.LOAD

2 Allocate a "delivered" TABLES library and a CUSTOM TABLES library. Allocate these PDS libraries with minimal primary and secondary space, minimal directory space, and DCB attributes RECFM=FB, LRECL=80, and any BLKSIZE.

```
...CMNZMF.VvRrMm.TABLES
...CMNZMF.VvRrMm.CUSTOM.TABLES
```

Step 2: APF Authorize Delivered and CUSTOM Load Libraries

Add these libraries to the APF list:

- Delivered CMNZMF and SERCOMC LOAD libraries that you unloaded from the ChangeMan ZMF installer.
- Custom LOAD libraries that you allocated in the previous step.

You can use the SETPROG system command to add these libraries to a dynamic APF list until the next IPL. If you use a static APF list, you must IPL before you can execute some of the subsequent steps in the installation process.

Step 3: Apply ChangeMan ZMF License

Apply the license you received from Serena to enable ChangeMan ZMF and any of the selectable options you licensed.

NOTE ChangeMan ZMF 8 requires a new license key.

See the *Serena SER10TY User's Guide* for instructions on how to apply the license. The load modules, JCL, and other components that run SER10TY are included in the SERCOMC libraries unloaded by the ZMF installer.

Step 4: Define ChangeMan ZMF VSAM Files

ChangeMan ZMF base system data is at least seven VSAM files and two QSAM files.

- Package Master VSAM KSDS File
- Component Master VSAM KSDS File
- Long Name Component Master VSAM KSDS File
- Package and Component Master Forward Recovery VSAM KSDS File
- Activity Log VSAM KSDS File
- Delay VSAM KSDS File
- Impact Analysis VSAM LDS File
- Impact Analysis Forward Recovery QSAM Log File

All of these data sets are referenced by DD statements in the ChangeMan ZMF started procedure.

Execute this procedure to define and initialize ChangeMan ZMF VSAM files and to allocate sequential log recovery files.

1. Copy member **INITIAL1** and **INITIAL2** from the delivered CMNZMF CNTL library to your custom CMNZMF CNTL library.
2. Edit member INITIAL1 in your custom CNTL library.
   - **a** Code your JOB statement at the top.
   - **b** Change all occurrences of DSN node placeholder *somnode* to create data set names that comply with your local naming conventions. Remember to change DSN names on the IDCAMS DELETE as well as DEFINE commands for VSAM files.
   - **c** Code primary and secondary space allocations in the CYLINDERS and MEGABYTES space parameters.

The amount of space you need depends on the number of components you manage with ChangeMan ZMF, how often they are changed, how many change packages you create, how many relationships between components are tracked, and aging and archiving criteria.

Serena recommends that you start with these allocations.

VSAM File	Unit	Pri	Sec
CMNPMAST (Package Master)	CYL	20	5
CMNCMPNT (Component Master)	CYL	20	5
CMNCMPNL (Long Name Component Master)	CYL	20	5
IADSP (Impact Analysis)	MB	10	5
CMNLOG (Log File)	CYL	10	5
CMNDELAY (Delay File)	CYL	1	1
CMNRECV (Recovery File)	CYL	10	5

NOTE If you do not anticipate managing any components with names longer than 8 characters, you can use a minimum space allocation for the CMNCMPNL long name component master.

3 Submit INITIAL1.

4 Examine the job output for problems.

5 Edit member INITIAL2 in your custom CNTL library.

a Code your JOB statement at the top.

b Change the JOBLIB statement to concatenate your delivered CMNZMF and SERCOMC LOAD libraries.

c Change all occurrences of DSN node placeholder *somnode* to match the files that you defined with INITIAL1.

d Use the following table to determine the execution parameter values for program CMNINTPM in job step INITPM.

Keyword Parameter	Purpose and Valid Values	
RSRCCL	Defines the name of the resource class under which ChangeMan ZMF security entities are defined in your security system.	
	$CHGMAN	Suggested if your security system is IBM Security Server RACF.
	CMN	Suggested if your security system is CA ACF2.
	UR1	Suggested if your security system is CA Top Secret.

Keyword Parameter	Purpose and Valid Values
ENVTYP	Defines the ChangeMan ZMF environment you are building. **Note:** Serena recommends that you define your first ChangeMan ZMF test or demonstration instance as an A or All environment. A An *all* environment where a single ChangeMan ZMF instance manages development, maintains baseline libraries, and maintains production libraries that are on shared DASD and shared catalogs. D A *development* environment where a ChangeMan ZMF instance manages development and maintains baseline libraries, but it does not maintain production libraries. A D environment sends package information to a separate P environment for installation into production libraries. P A *production* environment where a stripped-down ChangeMan ZMF instance receives change package information from a D or DP environment and installs package components into production libraries. No development is performed in a P environment. A P environment is required where production libraries reside on an LPAR that is separate from development and does not share DASD and catalogs with development. A P environment is also required if DB2 binds or IMS gens for install are performed on an LPAR that is separate from development. DP A *development and production* environment where a ChangeMan ZMF instance manages development, maintains baseline libraries, manages production libraries on shared DASD and shared catalogs, and can send package information to a separate P environment for installation into production libraries that are not on shared DASD with shared catalogs.

6 Submit INITIAL2.

7 Examine the job output for problems.

a Example of SYSPRINT from program CMNINTPM in job step INITPM:

```
         CMNINTPM - Resource Class=$CHGMAN
         CMNINTPM - CMN Environment=DP
         CMNINTPM - Package Master Initialized
```

b Example of CMNPRINT from program CMNIAIN0 in job step CMNIAIN0:

```
                  ChangeMan(R) ZMF Impact Analysis Dataspace Analysis and Initialization
                               -- CMNIAIN0 - 8.1.2 20150910 16.48
20161030 22432815  Execution parameter is: INITIAL
20161030 22432828 I/A dataspace initialization complete
```

Step 5: Define XMLSPACE VSAM LDS

Serena XML Services uses a VSAM Linear Data Set (LDS) to map the relationships between XML schemas and DSECTS used for fixed-format control blocks and copybooks. This data set is referenced by the XMLSPACE DD statement in the ChangeMan ZMF started procedure.

One XMLSPACE LDS can be used for all ChangeMan ZMF instances running at the same maintenance release level on the same LPAR.

Execute this procedure to define and populate the XMLSPACE VSAM LDS.

NOTE This step also creates a sequential MAPDATA data set used by the XML prototype tool XMLSERV.

1. Copy member **XMLLOAD** from the delivered SERCOMC CNTL library to your custom CMNZMF CNTL library.
2. Edit member XMLLOAD in your custom CNTL library.
   a. Code your JOB statement at the top.
   b. In step RECEIVE, change DSN node placeholder *somnode* in the RECEIVE command INDS parameter to point to your delivered SERCOMC CNTL library.
   c. Change all other occurrences of DSN node placeholder *somnode* to create data set names that comply with your local naming conventions. Keep the last node names MAPDATA and XMLSPACE.
3. Submit JCL member XMLLOAD.
4. Examine the job output for problems:
   a. Example of SYSTSPRT from job step RECEIVE:

```
READY
DELETE 'CMNTP.S6.SERCOMC.MAPDATA'
IDC3012I ENTRY CMNTP.S6.SERCOMC.MAPDATA NOT FOUND+
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLEG-42
IDC0551I ** ENTRY CMNTP.S6.SERCOMC.MAPDATA NOT DELETED
IDC0014I LASTCC=8
READY
RECEIVE INDS('CMNTP.S6.V813.SERCOMC.CNTL(XMLDATA)')
INMR901I Dataset DDNAME.MAPDATA from BATCHTSO on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +
INMR908A The input file attributes are: DSORG=SEQUENTIAL, RECFM=F, BLKSIZE=4096,
INMR909A You may enter DSNAME, SPACE, UNIT, VOL, OLD/NEW, or RESTORE/COPY/DELETE
INMR001I Restore successful to dataset 'CMNTP.S6.SERCOMC.MAPDATA'
READY
END
```

   b. Browse member XMLDATA in your SERCOMC CNTL library. Note the date and time in this line of text near the top of the data set:

```
SERNET XML Dsect Cross Reference. Created: dd mmm yyyy hh:mm:ss
```

   Browse the new MAPDATA file and look for the same text at the top of that file. The date and time should be the same as in the XMLDATA member.

   c. Examine the job JESMSGLG to verify that job step LOAD completed with RC=00.

Step 6: Define Staging Versions VSAM Files

The staging versions facility of ChangeMan ZMF retains multiple versions of edited components. This facility is supported by three VSAM files, which are included in the ChangeMan ZMF started procedure.

NOTE If you do not intend to use staging versions to keep prior versions of staged package components, then skip this step.

Execute this procedure to define and initialize three staging versions VSAM files.

1 Copy members **INITIAL3** and **SVPARM** from the delivered CMNZMF CNTL library to your custom CMNZMF CNTL library.

2 Edit member INITIAL3 in your custom CNTL library.

CAUTION! Follow these instructions for editing INITIAL3 exactly as they are written. The staging versions facility is based on Serena® ChangeMan® SSM technology. Some control statements and security definitions are included in this job, even though they serve no function in staging versions under ChangeMan ZMF. If you change these control statements and security definitions, this job will not process successfully.

a Code your JOB statement at the top.

b Change the JOBLIB statement to concatenate your delivered CMNZMF and SERCOMC LOAD libraries.

c Change all occurrences of *somnode*.CMNZMF.CNTL to point to your custom CMNZMF CNTL library.

d Change all remaining occurrences of DSN node placeholder *somnode* to create data set names that comply with your local naming conventions. Remember to change DSN names on the IDCAMS DELETE as well as DEFINE commands for VSAM files.

e Code primary and secondary space allocations in the CYLINDERS space parameter in the IDCAMS DEFINE statements. The amount of space you need depends on the number and size of components for which staging versions is enabled, and the level of development activity at your site. Serena recommends that you start with these allocations:

VSAM File	Cylinders Pri	Sec
CMNVSAM (Staging Versions Archive Master)	10	10
CMNSVDM (Staging Versions Delta Master)	10	50
CMNSVOM (Staging Versions Object Master)	10	10
CMNSVSM (Staging Versions Site Master)	10	10

f If you use a PDS to store licenses applied with Serena License Manager, add DD statement SERLIC to the four job steps that execute PGM=HPSTRACK, job step five that executes PGM=HPSMAINT and code the DSN= with your license PDS.

If you use CSA to store licenses, do not add DD statement SERLIC to the job JCL. See "Step 3: Apply ChangeMan ZMF License" on page 61 and "SERLIC DD Statement" on page 38.

g In STEP1, change placeholder *installer* to your TSO ID in the following HPSIN control statement for program HPSTRACK:

```
OMADMADD=(ID=installer,NAME='Installer')
```

h In STEP4, change placeholder *your.source.library* to the data set name of any source code library that contains members that are typical of source that will be managed by ChangeMan ZMF.

This step generates a compression table to optimize text compression in the CMNSVDM VSAM file. The library you specify is used only to generate the table. No data from the library is loaded to the compression table.

3 Edit SVPARM in your custom CNTL library.

a In the SITEMASTER, OBJECTMASTER, and DELTAMASTER parameters, code the VSAM data set names that you used in CNTL member INITIAL3. (See the DELETE statements in job step STEP0.)

b In the SITEADMINS parameter, change placeholder *installer* to your TSO ID.

c Save member SVPARM.

4 Submit INITIAL3.

5 Examine the job output for problems.

a Review JESMSGLG for messages from program HPSTRACK.

b Example of CMNPRINT from program HPSTRACK in job step STEP1:

```
*****************************************************************************************************************************
*                                                                                                                           *
*                                  Serena ChangeMan SSM   (V8.5.1.00)                                    PAGE 0000001 *
*                                  SERENA SOFTWARE, INC                                                                     *
*                       PROPRIETARY SOFTWARE PRODUCT OF Serena Software, Inc.  ALL RIGHTS RESERVED.                         *
*   Mon  2016/10/31_19:03:38    JOBNAME = USER0153    STEPNAME = STEP1       PGMNAME = HPSTRACK    JULDATE = 2016305   *
*****************************************************************************************************************************


HPS0022I  HPSINIT   19:03:38 Parmlib in effect: CMNTP.CMN813.C6.CNTL(SVPARM)
                             Security Userid  : USER015

HPS0037I  HPSINIT   19:03:38 M-lrecl = 20,200
                             M-datsi = 20,145

HPS0029I  HPSINIT   19:03:39 Successful open of Site Master:   CMNTP.CMN813.CMNSVSM

HPS0334I  HPSOINIT  19:03:39 Max-lrecl = 4,084
                             Max-data  = 4,009

HPS0329I  HPSOINIT  19:03:40 Successful open of Object Master: CMNTP.CMN813.CMNSVOM

Command    Column  10        20        30        40        50        60        70        80                    PAGE 0000002
Count-seg  ....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
    1     "TRACE=OFF                                                                                          "
    2     "OMADMADD=(ID=********,NAME='Universal Admin')                                                      "
    3     "OMADMADD=(ID=USER015,NAME='Installer')                                                             "

HPS1300I  HPSTRACK  19:03:40 EOF HPSIN:  Cards read=4   with 3 Commands.

HPS1361I  HPSTRACK  19:03:40 ADMADD (********        ) function successful.

HPS1361I  HPSTRACK  19:03:40 ADMADD (USER015         ) function successful.

HPS0000I  HPSTRACK  19:03:40 Job Started: 2016/10/31_19:03:38
                             Job Ended  : 2016/10/31_19:03:40
                             Warning cnt: 0
                             Severe msgs: 0
```

c Example of CMNPRINT from program HPSTRACK in job step STEP2:

```
*****************************************************************************************************************************
*                                                                                                                           *
*                                  Serena ChangeMan SSM   (V8.5.1.00)                                    PAGE 0000001 *
*                                  SERENA SOFTWARE, INC                                                                     *
*                       PROPRIETARY SOFTWARE PRODUCT OF Serena Software, Inc.  ALL RIGHTS RESERVED.                         *
*   Mon  2016/10/31_19:03:40    JOBNAME = USER0153    STEPNAME = STEP2       PGMNAME = HPSTRACK    JULDATE = 2016305   *
*****************************************************************************************************************************


HPS0022I  HPSINIT   19:03:40 Parmlib in effect: CMNTP.CMN813.C6.CNTL(SVPARM)
                             Security Userid  : USER015

HPS0029I  HPSINIT   19:03:40 Successful open of Site Master:   CMNTP.CMN813.CMNSVSM

HPS0329I  HPSOINIT  19:03:41 Successful open of Object Master: CMNTP.CMN813.CMNSVOM

Command    Column  10        20        30        40        50        60        70        80                    PAGE 0000002
Count-seg  ....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
    1     "TRACE=OFF                                                                                          "
    2     "GRPADMADD=(ID=*******,NAME='Universal Admin') for CMN SV                                           "

HPS1300I  HPSTRACK  19:03:41 EOF HPSIN:  Cards read=3   with 2 Commands.

HPS1336I  HPSTRACK  19:03:41 DEPTADD/GRPADMADD - Department added: *******

HPS0000I  HPSTRACK  19:03:41 Job Started: 2016/10/31_19:03:40
                             Job Ended  : 2016/10/31_19:03:41
                             Warning cnt: 0
                             Severe msgs: 0
```

d Example of CMNPRINT from program HPSTRACK in job step STEP3:

```
*********************************************************************************************************************************
*                                                                                                                               *
*                                  Serena ChangeMan SSM   (V8.5.1.00)                                    PAGE 0000001 *
*                                  SERENA SOFTWARE, INC                                                                         *
*                      PROPRIETARY SOFTWARE PRODUCT OF Serena Software, Inc.  ALL RIGHTS RESERVED.                              *
*   Mon  2016/10/31_19:03:42    JOBNAME = USER0153    STEPNAME = STEP3       PGMNAME = HPSTRACK    JULDATE = 2016305   *
*********************************************************************************************************************************


HPS0022I  HPSINIT   19:03:42 Parmlib in effect: CMNTP.CMN813.C6.CNTL(SVPARM)
                             Security Userid  : USER015

HPS0029I  HPSINIT   19:03:42 Successful open of Site Master:   CMNTP.CMN813.CMNSVSM

HPS0329I  HPSOINIT  19:03:42 Successful open of Object Master: CMNTP.CMN813.CMNSVOM

Command    Column  10        20        30        40        50        60        70        80                    PAGE 0000002
Count-seg  ....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
    1     "TRACE=OFF                                                                      "
    2     "GRPADD=(GROUP=CMNGROUP)                                                        "

HPS1300I  HPSTRACK  19:03:42 EOF HPSIN:  Cards read=2   with 2 Commands.

HPS0329I  HPSOINIT  19:03:43 Successful open of Object Master: CMNTP.CMN813.CMNSVOM

HPS1338I  HPSTRACK  19:03:43 GRPADD - Group Added: *******         .CMNGROUP

HPS0000I  HPSTRACK  19:03:43 Job Started: 2016/10/31_19:03:42
                             Job Ended  : 2016/10/31_19:03:43
                             Warning cnt: 0
                             Severe msgs: 0
```

e Example of CMNPRINT from program HPSTRACK in job step STEP4:

```
*********************************************************************************************************************************
*                                                                                                                               *
*                                  Serena ChangeMan SSM   (V8.5.1.00)                                    PAGE 0000001 *
*                                  SERENA SOFTWARE, INC                                                                         *
*                      PROPRIETARY SOFTWARE PRODUCT OF Serena Software, Inc.  ALL RIGHTS RESERVED.                              *
*   Mon  2016/10/31_19:03:43    JOBNAME = USER0153    STEPNAME = STEP4       PGMNAME = HPSTRACK    JULDATE = 2016305   *
*********************************************************************************************************************************


HPS0022I  HPSINIT   19:03:43 Parmlib in effect: CMNTP.CMN813.C6.CNTL(SVPARM)
                             Security Userid  : USER015

HPS0329I  HPSOINIT  19:03:44 Successful open of Object Master: CMNTP.CMN813.CMNSVOM

HPS0029I  HPSINIT   19:03:44 Successful open of Site Master:   CMNTP.CMN813.CMNSVSM

Command    Column  10        20        30        40        50        60        70        80                    PAGE 0000002
Count-seg  ....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+....+
    1-1   "  IDVTABLE=(TABLENAME=TEXT,                                                    "
     -2   "            REFDSN=USER015.JCL.CNTL)                                         "

HPS1300I  HPSTRACK  19:03:44 EOF HPSIN:  Cards read=2   with 1 Commands.

HPS0502I  HPSTRACK  19:03:44 Delta Master cluster has been initialized.Max-lrecl = 22,520
                             Max-data size = 22,506

HPS0504I  HPSTRACK  19:03:45 Successful open of Delta Master:  CMNTP.CMN813.CMNSVDM

HPS0329I  HPSOINIT  19:03:45 Successful open of Object Master: CMNTP.CMN813.CMNSVOM

HPS1384I  HPSTRACK  19:03:48 Standard Compression-Expansion table defined: TEXT

HPS0000I  HPSTRACK  19:03:48 Job Started: 2016/10/31_19:03:43
                             Job Ended  : 2016/10/31_19:03:48
                             Warning cnt: 0
                             Severe msgs: 0
```

f Example of CMNPRINT from program HPSTRACK in job step STEP5

```
*****************************************************************************************************************************
*                                                                                                                           *
*                                  Serena ChangeMan SSM   (V8.5.1.00)                                    PAGE 0000001 *
*                                  SERENA SOFTWARE, INC                                                                     *
*                       PROPRIETARY SOFTWARE PRODUCT OF Serena Software, Inc.  ALL RIGHTS RESERVED.                         *
*   Mon  2016/10/31_19:03:49    JOBNAME = USER0153    STEPNAME = STEP5       PGMNAME = HPSMAINT    JULDATE = 2016305   *
*****************************************************************************************************************************


HPS0022I  HPSINIT   19:03:49 Parmlib in effect: CMNTP.CMN813.C6.CNTL(SVPARM)
                             Security Userid  : USER015

Execute parm = "LIC=SSV,ARCHIVE"


Options in effect: Purge=None
                   Level=None
                   Trace=OFF
                   Archive

HPS1722I  HPSMAINT  19:03:50 SUMMARY REPORT:
                                       0  Site Master records updated
                                       0  Site Master records erased
                                       0  Object Master records erased
                                       0  Delta Master records updated
                                       0  Delta Master records erased
                                       0  Archive Master records added
                                       0  Archive Master bytes written

                                       1  Archive Master file size (records)
                                     100  Archive Master file size (bytes)
                                       0  Files being tracked

HPS0000I  HPSMAINT  19:03:50 Job Started: 2016/10/31_19:03:49
                             Job Ended  : 2016/10/31_19:03:50
                             Warning cnt: 0
                             Severe msgs: 0
```

NOTE If your INITIAL3 job fails, compare your JCL to the original job in the delivered CNTL library. If you have made a change to any JCL statement or control statement that was not explicitly described in this section, restore the statement to its delivered state and submit INITIAL3 again.

Step 7: Store SVPARM Control Member

NOTE If you do not intend to use staging versions to keep prior versions of staged package components, then skip this step.

Copy member **SVPARM** from your custom CNTL library into a production or systems control member library. You created member SVPARM in your custom CNTL library in "Step 6: Define Staging Versions VSAM Files" on page 65.

Step 8: Allocate the SER#PARM TCP/IP Address Library

SERNET uses a PDS or PDSE library to store TCP/IP addresses and port numbers for Serena applications. See "SER#PARM DD Statement" on page 37. Allocate this library even if you do not intend to use TCP/IP for ChangeMan ZMF communication.

CAUTION! Do not use the SER#PARM library for any other purpose. SERNET opens this library for output, which can interfere with other uses of the file.

Allocate the PDS(E) with these characteristics:

```
DSN=somnode.SERCOMC.TCPIPORT      * Recommended last node
DCB=(RECFM=FB,LRECL=80,BLKSIZE=0) * Let SMS set BLKSIZE
SPACE=(TRK,(2,5,30))
```

If there is a TCP/IP address space on the LPAR where SERNET and ChangeMan ZMF run, SERNET automatically creates a #SERx member, where "x" is the subsystem ID of the SERNET started task, to store TCP/IP addresses and port numbers.

No TCP/IP Address Space

If there is no TCP/IP address space on the image where the SERNET started task will run, manually create a member in the SER#PARM file and code a dummy TCP/IP address and port number:

1. Create member #SERx, where "x" is the subsystem ID of the SERNET started task.
2. Edit member #SERx and code these lines to set the TCP/IP address and port number to zero. (You may omit the scale.)

```
* --+----1----+----2----+----3----+----4----+----5----+----6----+
* SMFI S APP TCPIPROC PORT# ADDR
  smfi x XCH          00000 D001
  smfi x CMN          00000 D001
```

Where:

smfi	Four character SMFID of the system.
x	SERNET started task subsystem ID.

If you do not know the SMFID of the system, execute the following REXX:

```
/* REXX */ say mvsvar(syssmfid)
```

You can also browse SYS1.PARMLIB(SMFPRMxx) and look for the SID= parameter.

Step 9: Create ISPF Keylist for EXPAND

ChangeMan ZMF panels reference a special keylist that includes a PF key assignment for **EXPAND**.

Execute this procedure to generate keylist member CMNKEYS.

1. Copy member **CMNDTL** from the delivered CMNZMF CNTL library to your custom CMNZMF CNTL library. Note this now has three ISPSTART commands.
2. Edit member CMNDTL in your custom CNTL library.
   a. Code your JOB statement at the top.
   b. Code DD name DTLGML with the delivered CMNZMF SAMPLES library.
   c. Code DD name DTLTAB with the "delivered" TABLES library you allocated in "Step 1: Allocate CUSTOM Libraries" on page 60.
   d. Change all remaining occurrences of placeholder *somnode* to the high level qualifier you use for ISPF system libraries.
3. Submit JCL member CMNDTL.
4. Examine the job output for problems.
   a. Example of ISPLOG from job step DTL:

```
Time               *** ISPF transaction log ***                      Userid: USER015   Date: 16/10/31   Page: 1

21:45   Start of ISPF Log - - -  - Session # 1 ------------------------------------------------------
21:45      TSO     - Command  -  - ISPDTLC CMNELSK (PROFDDN=* CUASUPP REPLACE)
21:45                            - ISPC028AI: Total of 2 message(s) have been suppressed.
21:45                            - ISPC029AI: Total of 2 warning(s) and 1 error(s) found.
21:45    End of ISPF Log - - - - - Session # 1 ------------------------------------------------------
Time               *** ISPF transaction log ***                      Userid: USER015   Date: 16/10/31   Page: 1

21:45   Start of ISPF Log - - -  - Session # 2 ------------------------------------------------------
21:45      TSO     - Command  -  - ISPDTLC CMNEROK (PROFDDN=* CUASUPP REPLACE)
21:45                            - ISPC028AI: Total of 2 message(s) have been suppressed.
21:45                            - ISPC029AI: Total of 2 warning(s) and 1 error(s) found.
21:45    End of ISPF Log - - - - - Session # 2 ------------------------------------------------------
Time               *** ISPF transaction log ***                      Userid: USER015   Date: 16/10/31   Page: 1

21:45   Start of ISPF Log - - -  - Session # 3 ------------------------------------------------------
21:45      TSO     - Command  -  - ISPDTLC CMNMREK (PROFDDN=* CUASUPP REPLACE)
21:45                            - ISPC028AI: Total of 2 message(s) have been suppressed.
21:45                            - ISPC029AI: Total of 2 warning(s) and 1 error(s) found.
21:45    End of ISPF Log - - - - - Session # 3 ------------------------------------------------------
```

b Example of SYSTSPRT from job step DTL (ensure it runs on an SMS managed dataset or there will be LIBDEF RC=20 and a failure):

```
READY
 ISPSTART CMD(ISPDTLC CMNELSK (PROFDDN=* CUASUPP REPLACE))
ISPC021I: ISPDTLC ISPF 7.1 Dialog Tag Language Conversion Utility
ISPC020AI: ISPF Dialog Tag Language Conversion Utility for ISPF 7.1. Current
   APAR level: PID. Current PTF number: 11 Dec 2012.
ISPC022I: Converting source file "CMNTP.CMN813.SAMPLES(CMNELSK)"...
ISPC028I: 2 message(s) have been suppressed.
ISPC029I: 2 warning(s) and 0 error(s) found.
ISPD117
The initially invoked CLIST ended with a return code = 1
 USER015.USER015A.J0391638.D0000103.? was preallocated (no free was done).
READY
 ISPSTART CMD(ISPDTLC CMNEROK (PROFDDN=* CUASUPP REPLACE))
ISPC021I: ISPDTLC ISPF 7.1 Dialog Tag Language Conversion Utility
ISPC020AI: ISPF Dialog Tag Language Conversion Utility for ISPF 7.1. Current
   APAR level: PID. Current PTF number: 11 Dec 2012.
ISPC022I: Converting source file "CMNTP.CMN813.SAMPLES(CMNEROK)"...
ISPC028I: 2 message(s) have been suppressed.
ISPC029I: 2 warning(s) and 0 error(s) found.
ISPD117
The initially invoked CLIST ended with a return code = 1
 USER015.USER015A.J0391638.D0000103.? was preallocated (no free was done).
READY
 ISPSTART CMD(ISPDTLC CMNMREK (PROFDDN=* CUASUPP REPLACE))
ISPC021I: ISPDTLC ISPF 7.1 Dialog Tag Language Conversion Utility
ISPC020AI: ISPF Dialog Tag Language Conversion Utility for ISPF 7.1. Current
   APAR level: PID. Current PTF number: 11 Dec 2012.
ISPC022I: Converting source file "CMNTP.CMN813.SAMPLES(CMNMREK)"...
ISPC028I: 2 message(s) have been suppressed.
ISPC029I: 2 warning(s) and 0 error(s) found.
ISPD117
The initially invoked CLIST ended with a return code = 1
 USER015.USER015A.J0391638.D0000103.? was preallocated (no free was done).
READY
END
```

Step 10: Build SERNET JCL for ChangeMan ZMF

The procedure for a SERNET started task that runs ChangeMan ZMF is built from two JCL members delivered on the ZMF installer.

See "SERNET Started Procedure With ChangeMan ZMF" on page 142 for an example of a finished procedure.

Execute this procedure to build JCL for a SERNET instance that runs ChangeMan ZMF.

1 In your custom CNTL library, create a member for the SERNET started task procedure. Name the member according to the convention that you established in "SERNET Started Task Names" on page 35.

2 Copy in member **SERVER** from the delivered SERCOMC CNTL library. This is the model JCL for a SERNET started task.

3 After the SERVER model JCL, copy in member **STARTJCL** from the delivered CMNZMF CNTL library. This is the model for ChangeMan ZMF DD statements.

4 Code the STEPLIB DD statement to concatenate these libraries in the stated order:

a Custom LOAD libraries that you allocated for customized CMNZMF and SERCOMC load modules. See "Step 1: Allocate CUSTOM Libraries" on page 60.

b Delivered CMNZMF LOAD library and delivered SERCOMC LOAD library that you unloaded from the ZMF installer. See Chapter 4, "Unloading the Software" on page 51.

c CA Librarian product library or CA Panvalet product library if the library is not in the LINKLIST and you intend to use these products for ChangeMan ZMF baseline libraries. See "CA Panvalet® and CA Librarian® Compatibility" on page 26.

5 Code the SER#PARM DD statement to point the PDS(E) that you created in "Step 8: Allocate the SER#PARM TCP/IP Address Library" on page 70.

6 Code the XMLSPACE DD statement to point to the VSAM LDS that you created and populated with JCL member XMLLOAD in "Step 5: Define XMLSPACE VSAM LDS" on page 63.

7 If you use a PDS to store licenses applied with Serena License Manager, add DD name SERLIC to point to your license PDS.

If you use CSA to store licenses, do not add DD name SERLIC to the started task JCL.

See "Step 3: Apply ChangeMan ZMF License" on page 61 and "SERLIC DD Statement" on page 38.

8 If you will store SERNET keyword options in a PDS or PDSE library member (see "Passing Parameters to SERNET" on page 36), do the following:

a Code keyword option DDNAME=*ddname* in the PARM parameter for program SERVER.

b Allocate the PDS or PDSE library with LRECL=80

c Code a corresponding *ddname* DD statement that points to the library that you allocated and the member where you will store SERNET keyword options.

9 Create a GDG index for SYSMDUMP data sets. (See "SYSMDUMP DD Statement" on page 39.) Code the GDG data set name at the SYSMDUMP DD statement.

10 Code the following DD names to point to the VSAM clusters that you created with JCL member INITIAL1 in "Step 4: Define ChangeMan ZMF VSAM Files" on page 61. The last node in the VSAM cluster name that you created should be the same as the DD name.

CMNPMAST
CMNCMPNT
CMNCMPNL

NOTE If you will use Batch Local Shared Resources (BLSR) for ChangeMan ZMF VSAM files (see "VSAM I/O Optimization of ZMF Master Files" on page 33), follow the instructions in the JCL comments to use an alternate DD statement for CMNPMAST, CMNCMPNT, and CMNCMPNL.

11 Code the CMNIMPCT and CMNIALOG DD statements to point to the VSAM LDS and the sequential log file that you created with JCL member INITIAL1 in "Step 4: Define

ChangeMan ZMF VSAM Files" on page 61. These DD names do not match the last node of the data set names that you created:.

DD Name	Last DSN Node
CMNIMPCT	IADSP
CMNIALOG	IALOG

12 Code the following DD names to point to the VSAM clusters that you created with JCL member INITIAL1 in "Step 4: Define ChangeMan ZMF VSAM Files" on page 61. The last node in the VSAM cluster name that you created should be the same as the DD name.

CMNLOG
CMNRECV
CMNDELAY

13 If you intend to use the staging versions facility of ChangeMan ZMF, code the HPSPLIB DD statement to point to the SVPARM control member that you edited and saved in "Step 7: Store SVPARM Control Member" on page 69. SERLIC DD statement must be added to this job and any procedure using DD statement HSPSLIB..

NOTE If you do not intend to use staging versions to keep prior versions of staged package components, then delete the HPSPLIB and HPSIN DD statements. Do not use DD DUMMY or DSN=NULLFILE.

14 Change placeholder *somnode* in the following ISPF system libraries to conform to your standards:

```
somnode.SISPMENU
somnode.SISPTENU
```

15 Change placeholder *somnode* in the ISPF libraries at the following ISPF DD statements to match the library names for delivered libraries you created in "Step 3: Expand PDS Libraries with RECEIVE" on page 57 and CUSTOM library names you created in "Step 1: Allocate CUSTOM Libraries" on page 60:

```
ISPPLIB
ISPMLIB
ISPSLIB
ISPLLIB
ISPTLIB
```

16 Code required SERNET keyword options in the PARM parameter for program SERVER, or code them in the library member referred to by the DD name specified in the DDNAME=*ddname* keyword option. See "Parameters for SERNET and ChangeMan ZMF" on page 36.

NOTE If you code keyword option SUBSYS=*subsysID* as a record in the library member referred to by DDNAME=*ddname*, then remove all references to the symbolic parameter &ID in the started procedure.

17 If you do not use the symbolic parameter OPT, delete it from the PROC statement at the top of the JCL, and remove it from the PARM statement for program SERVER.

18 Add DD statements that are required to resolve conflicts between ChangeMan ZMF and other software tools in use at your site. See "Software Conflicts" on page 27.

19 Install the SERNET procedure in a system PROCLIB.

NOTE If you intend to enforce local business rules in ChangeMan ZMF functions, you must establish a separate HLLX started task which will be controlled by ZMF only. Follow the instructions detailed in the *ChangeMan ZMF 8.1 High Level Language Functional Exits Getting Started Guide* for this feature.

Step 11: Build Default File Tailoring Procedure

ChangeMan ZMF starts a file tailoring procedure to run in a separate address space to create JCL for package installation jobs, stage build jobs, promotion jobs, and other batch processes.

See "CMNxADSP Default File Tailoring Procedure" on page 142 for an example of a finished procedure.

Execute this procedure to build a default file tailoring started procedure for the ChangeMan ZMF instance that you built in "Step 10: Build SERNET JCL for ChangeMan ZMF" on page 72.

1 In your custom CNTL library, create a new member named CMN*x*ADSP, where *x* is the subsystem ID of the SERNET started task.

2 Copy in member **CMNADSP** from the delivered CNTL library. Edit CMNxADSP in your CUSTOM CNTL library.

3 Copy the library concatenations for following DD statements from the SERNET started procedure you created in "Step 10: Build SERNET JCL for ChangeMan ZMF" on page 72 into the corresponding DD statements in CMNxADSP:

```
STEPLIB
SER#PARM
ISPPLIB
ISPMLIB
ISPSLIB
ISPTABL
```

4 At the SYSEXEC DD statement, change placeholder *somnode* in the REX and CEXEC libraries to match the library names for delivered libraries you created in "Step 3: Expand PDS Libraries with RECEIVE" on page 57 and CUSTOM library names you created in "Step 1: Allocate CUSTOM Libraries" on page 60.

5 In the ISPTLIB DD statement, concatenate the same libraries you coded in the SERNET started procedure at the ISPTLIB DD statement, leaving the first library as a temporary library without a DSN.

6 Change the high level qualifier on the library in the ISPTABL DD statement to conform to your standards for system ISPF libraries.

7 Add the same DD statements that you added to the SERNET started task JCL to resolve conflicts between ChangeMan ZMF and other software tools in use at your site.

CAUTION!

- Do not code any values for symbolic parameters. The calling programs resolve symbolic parameters.
- Do not change any other JCL statements such as the PROC name, step name, or DD names.

8 Install started procedure CMNxADSP in a system PROCLIB.

Step 12: Build the Logon CLIST

Users and administrators execute a CLIST to connect to SERNET and ChangeMan ZMF.

Execute this procedure to build a CLIST to connect to the ChangeMan ZMF instance that you built in "Step 10: Build SERNET JCL for ChangeMan ZMF" on page 72.

1 In your custom CMNZMF CLIST library, create a member for the logon CLIST. For your first ChangeMan ZMF instance, name the CLIST the same as the started procedure.

2 Copy in member **CMN** from the delivered CMNZMF CLIST library.

3 Set the default value for parameter SS to the subsystem ID that you specified for the SERNET started procedure. Example for subsystem ID 0:

```
SS(0)
```

4 A default value for the CON parameter is not recommended. Set parameter CON to default value T for TCP/IP or X for Cross Memory Services **only** if communication to the SERNET started task is exclusively one or the other.

5 In the ALLOC DD (SER#PARM) statement, code the library that you coded in the SER#PARM DD statement in the SERNET started procedure.

6 If you coded a SERLIC DD statement in the SERNET started procedure, code an ALLOC DD (SERLIC) statement with the same library name. Example:

```
ALLOC DD (SERLIC) DSN(   +
  'somnode.LICENSE.DATA' +
  ) SHR REU
```

7 In the ALLOC DD (SERPRINT) statement, code a SYSOUT class.

8 Change all remaining occurrences of placeholder *somnode* to create data set names that match the library names for delivered libraries you created in "Step 3: Expand PDS Libraries with RECEIVE" on page 57 and CUSTOM library names you created in "Step 1: Allocate CUSTOM Libraries" on page 60.

9 In the LIBDEF ISPTLIB statement, ensure that the TABLES library that you populated in "Step 9: Create ISPF Keylist for EXPAND" on page 70 is included in the concatenation.

10 Modify the ISPEXEC SELECT statement to add the subsystem ID to two parameters:

a NEWAPPL - This parameter specifies a 1- to 4-character application code that is prefixed to the user and edit profile names. By including the subsystem ID in NEWAPPL, separate profiles are maintained for each ZMF instance that you use. Example:

```
NEWAPPL(CMN&SS)
```

b SCRNAME - This ISPF command sets the 2- to 8-character screen name that is displayed in the upper left corner of ChangeMan ZMF panels when you manually issue the command SCRNAME ON. Adding the subsystem ID to the screen name makes it easy to identify which ZMF instance you are working on from any ZMF panel. Example:

```
SCRNAME(CMN&SS)
```

11 Test the CLIST syntax in the ISPF Command Shell panel:

```
EX 'somnode.CMNZMF.CUSTOM.CLIST(CMNx)'
```

The **ChangeMan Down** panel (CMNSTDWN) is displayed if you have no errors.

12 Copy the CLIST into a SYSTEM or user CLIST library.

Step 13: Modify Exit Program CMNEXINS

Follow these steps to modify exit program CMNEXINS to define the DSN that ChangeMan ZMF uses for utility data sets.

1 Copy the source for program **CMNEXINS** from the delivered CMNZMF ASMSRC library to your custom ASMSRC library.

2 Edit CMNEXINS to set the high level qualifiers for ChangeMan ZMF utility data sets. See "Utility Data Sets" on page 44.

TIP The exit program source contains comments and several samples that can help you code your DSN naming and allocation conventions.

3 Copy member **ASSEMBLE** from the delivered SERCOMC CNTL library to your custom CMNZMF CNTL library.

4 Edit the assemble and link JCL.

a Code your JOB statement at the top.

b Change the assembler program name and the linkage editor program name to comply with your local standards.

c Change the SYSLIB concatenation for the ASM step to include the delivered CMNZMF and SERCOMC copybook and source libraries.

d Change the SYSLIB concatenation for the LKED step to include the delivered CMNZMF and SERCOMC LOAD libraries.

e Change the SYSLMOD data set name in the link step to point to your custom LOAD library. Do not link-edit into a delivered LOAD library.

f Code the symbolic parameters on the EXEC statement, which follows the PEND statement at the bottom of the in-line procedure. Set MBR to the program name CMNEXINS. Code AC=0 for the authorization code symbolic parameter.

5 Submit the ASSEMBLE job JCL.

6 If the return code is zero, compare the directory entries for CMNEXINS in your custom LOAD library to the directory entries in the delivered CMNZMF LOAD library. If they are not identical, adjust the ASSEMBLE job JCL and submit the job again.

7 If you are changing exit CMNEXINS after you have brought up the SERNET started task, stop and start the started task to enable your changes.

Step 14: APF Authorize Other Load Libraries

All load libraries in the STEPLIB concatenation in the SERNET started task JCL must be APF authorized.

In "Step 2: APF Authorize Delivered and CUSTOM Load Libraries" on page 60, you added the ChangeMan ZMF delivered and custom LOAD libraries to the APF list.

If there are other libraries in the STEPLIB concatenation, such as custom promotion load libraries in a test ChangeMan ZMF instance, add those libraries to the APF list now.

You can use the SETPROG system command to add libraries to a dynamic APF list until the next IPL. If you use a static APF list, an IPL is required before you can bring up ChangeMan ZMF and before you can test some of the housekeeping jobs you build in Chapter 10, "Setting Up Housekeeping" on page 121.

Chapter 6

Configuring Security

This chapter tells you how to define a SERNET started task running ChangeMan ZMF to your security system and define security entities that grant authorization to execute ChangeMan ZMF functions.

Introduction	80
Step 1: Customize SERLCSEC	80
Step 2: Add ChangeMan ZMF to Your Security System	82
Step 3: Define Security Entities	88
Step 4: Define Data Set Access	91
Step 5: Add OMVS Segment To Use TCP/IP	93
Step 6: Add PassTicket Support In Sernet	93
Step 7: Set Security for USS File Systems	96

Introduction

See "Security Considerations" on page 40 for a description of how ChangeMan ZMF works with your security system.

This chapter includes instructions for configuring security in the three security systems that are compatible with ChangeMan ZMF: IBM Security Server RACF, CA ACF2, and CA Top Secret. Regardless of the security system you use, these are the tasks you perform.

- Set switches in security interface program SERLCSEC.
- Identify each SERNET instance to your security system as a started task.
- Assign a User ID to each SERNET started task. One User ID may be assigned to several SERNET started tasks if their data set access needs are identical.
- Set up a security resource class under which ChangeMan ZMF security entities can be defined.
- Define security entities to grant authorization to execute ChangeMan ZMF functions.
- Permit data set access to the SERNET started tasks.
- Add an OMVS segment for access to TCP/IP facilities in Unix System Services.

Step 1: Customize SERLCSEC

Program SERLCSEC establishes the security environment for the user's subtask in SERNET so that the subtask runs with the authority of the logged on user.

If your security system is IBM Security Server RACF, no changes to SERLCSEC are required.

If you use CA Top Secret or CA ACF2 as you security system, see the topics below for changes you might make to SERLCSEC.

Before you modify the source code for SERLCSEC, copy the source for program SERLCSEC from the delivered SERCOMC ASMSRC library to your custom ASMSRC library, and make your changes in the custom library.

To assemble and link edit SERLCSEC, See "Assembling SERLCSEC" on page 81.

CA Top Secret Security Violations

Many CA Top Secret customers can run with the delivered switch settings in SERLCSEC. However, if you are using CA Top Secret and there are too many security violations for

users to connect to ChangeMan ZMF, set the bit switch for &LOGSVIO to zero, as shown in the SERLCSEC code fragment below:

```
     *-------------------------------------------------------------------
     * Below is where the user can tailor the source code
     *-------------------------------------------------------------------
     &VERFYID SETB  1       (YES)       SAF user ID verification
   I - &LOGSVIO SETB  0       (YES)       security violation logging
   D - &LOGSVIO SETB  1       (YES)       security violation logging
     &LOGSEL  SETB  1       (NO)        selective logging (by calling pgm)
     &STUB    SETB  0       (NO)        nullfy security checking
     *-------------------------------------------------------------------
     *ACF2 batch ids are specified in a user modifiable table
     *     'F ACF2BAT WORD 1' will find the start of this table.
     *     Please modify it as necessary.
     *-------------------------------------------------------------------
     * Notes:
     *    #1 If &VERFYID is set to '1' in a Top Secret (TSS) shop,
     *       the started task must be set up as a MULTIUSER FACILITY.
     *    #2 &LOGSVIO must be set to '1' for &LOGSEL to have any affect.
     *       If selective logging is selected, the logging is determined
     *       by the calling program from products like XCH, CMN and CMW.
     *-------------------------------------------------------------------
```

CA ACF2 Batch LOGONID

Customers who use CA ACF2 for their security system may need to define alternate user IDs for the LOGONID in batch JCL submitted by ChangeMan ZMF through the SERNET started task. Batch LOGONIDs are included but commented out in ZMF JOB statement skeletons:

```
)CM //*LOGONID CMNBATCH   <--- UNCOMMENT FOR ACF2
```

Alternate user IDs are specified in a table in SERLCSEC. Alternate user IDs may be assigned globally across all SERNET instances, or an alternate user ID can be assigned to a specific SERNET instance by specifying a subsystem ID.

See the comments at label ACF2BAT in program SERLCSEC for details about coding entries in the LOGONID table.

NOTE The table at ACF2BAT was created to provide alternate user IDs in CA ACF2. However, user IDs coded in this table are also authorized in IBM Security Server RACF and CA Top Secret.

Assembling SERLCSEC

If you modified the source for program SERLCSEC, follow these steps to assemble the custom source to create a custom load module.

1. Copy member **ASSEMBLE** from the delivered SERCOMC CNTL library to your custom CMNZMF CNTL library.
2. Edit the assemble and link JCL.
   - **a** Code your JOB statement at the top.
   - **b** Change the assembler program name and the linkage editor program name to comply with your local standards.
   - **c** Change the SYSLIB concatenation for the ASM step to include the delivered CMNZMF and SERCOMC copybook and source libraries.

d Change the SYSLIB concatenation for the LKED step to include the delivered CMNZMF and SERCOMC LOAD libraries.

e Change the SYSLMOD data set name in the link step to point to your ChangeMan ZMF custom LOAD library. Do not link-edit into a delivered LOAD library.

f Code the symbolic parameters on the EXEC statement, which follows the PEND statement at the bottom of the in-line procedure. Set MBR to the program name SERLCSEC. Code AC=0 for the authorization code symbolic parameter.

3 Submit the ASSEMBLE job JCL.

4 If the return code is zero, compare the directory entries for SERLCSEC in your custom LOAD library to the directory entries in the delivered LOAD library. If they are not identical, adjust the ASSEMBLE job JCL and submit the job again.

5 If you are changing SERLCSEC after you have brought up the SERNET started task, shut SERNET down and restart it to enable your changes.

Stubbing ChangeMan ZMF Security

If you have difficulty gaining access to ChangeMan ZMF administration because of security problems, you can temporarily code SERLCSEC to disable security checking. Set the `&STUB` switch to 1 and set `&VERFYID` switch to 0.

CAUTION! Only use the STUB switch in SERLCSEC temporarily and with extreme caution because it allows all users access to all ChangeMan ZMF functions. This gives them the authority to update all libraries managed by ChangeMan ZMF. Stubbing security also allows anyone to connect to ChangeMan ZMF and make changes to global and application administration.

Step 2: Add ChangeMan ZMF to Your Security System

Set up entities and grant privileges in your security system to restrict access to ChangeMan ZMF functions. For a general discussion of security and ChangeMan ZMF, see "Security Considerations" on page 40.

Add ChangeMan ZMF to CA ACF2

The steps listed here provide examples specific to CA ACF2 for accomplishing the security setup tasks listed in the "Introduction" on page 80. The following conventions are used in these examples:

- **CMN** is the CA ACF2 resource type.
- **SERUSER** is the SERNET logon ID.
- **SERPROC** is the started procedure name.
- **CMNBATCH** is the batch LOGONID. ChangeMan ZMF JOB statement skeletons contain CMNBATCH for the LOGONID.

- **CMNxADSP** is the started procedure name for the default file tailoring started task, where x is the subsystem ID of the SERNET started task that initiates CMNxADSP.

IMPORTANT! Resource type CMN was suggested when you initialized the package master VSAM file in "Step 4: Define ChangeMan ZMF VSAM Files" on page 61. If you use a different resource type here to satisfy local requirements, code the same resource type in initialization subparameter "RSRCCL" on page 62.

This section is not intended to be an authoritative reference for CA ACF2 command syntax. Your security administrator should be aware of the intent of each step and should adjust the sample command syntax if necessary.

1 Change the ACF2 Global System Options (GSO) to associate a logon ID with started tasks. Set up a default logon ID to allow all started tasks to come up successfully. From the TSO command processing option, enter the following commands:

```
ACF2
SET CONTROL(GSO)
INSERT ACFSTCID (Set up default logon ID for started tasks)
OPTS STC (To have privilege of started task)
```

2 Set up SERUSER as unique logon ID for SERNET. From the TSO command processing option enter the following commands:

```
ACF2
SET CONTROL(GSO)
INSERT SERUSER (To insert a new logon ID)
OPTS STC (To have privilege of started task)
```

To give ChangeMan ZMF enough data set access to perform its functions, add non-cancel authority NONCNCL or PREFIX(********).

3 Add two TSO IDs to enable SERNET to submit batch jobs:

a Add one TSO ID with the started task option:

```
ACF
SET LID
INSERT SERPROC JOB STC ACC-SRCE(STCINRDR) MUSASS
```

MUSASS means multiple-user single address space system.

b Add a second ID with the TSO and batch option. Define this ID so that it can be invoked with //*LOGONID by SERNET programs SERVMDUH and SERUSER:

```
ACF
SET LID
INSERT CMNBATCH JOB TSO PROGRAM(SER-)
RESTRICT SUBAUTH
```

CAUTION! Verify that neither of these IDs is assigned to users as a TSO logon ID.

4 Create an ACF2 GSO record type STC for the default file tailoring started procedure CMNxADSP, where x is the subsystem ID of the SERNET started task:

```
ACF2
SET CONTROL(GSO)
INSERT STC.CMN LOGONID(SERUSER) STCID(CMNxADSP)
```

Assign the same user ID as the SERNET started task that initiates the file tailoring started task.

Make additional entries when you create other file tailoring started procedures and enter them in global administration.

5 Add the following version-dependent definitions to the environment:

a For ACF2 Version 4:

```
SAFMAPS      MAPS(CMN/CMN)
SAFPROT      SUBSYS(SVC109)
CNTLPTS(SER-)
CLASSES(CMN,DATA SET)
```

b For ACF2 Version 6.*x*:

```
SET C(GSO)
INSERT CLASMAP.CMN
RESOURCE(CMN)
RSRCTYPE(CMN)
INSERT SAFDEF.CMN001
ID(CMN001)
PROGRAM(SER-)
RB(SVC109)
RACROUTE(REQUEST=AUTH CLASS=CMN)
```

6 Copy member **#ACF2** from the delivered CMNZMF CNTL library to your custom CMNZMF CNTL library. This member contains model code to create a CA ACF2 rule base for ChangeMan ZMF security entities under the CMN resource class.

7 Edit PDS member #ACF2 to create CA ACF2 rules for the five administrative security entities. You will define more entities and rules later, but these definitions allow you to get ChangeMan ZMF running.

a Code your JOB statement at the top.

b Code a valid output data set name for the rule base PDS.

c Change IEBUPDTE control characters from "$/" to "./"

d Code CA ACF2 rules for the five administrative security entities listed in "Administrator and Change Manager Security Entities" on page 41. Remember that if you want rules specific to one subsystem ID, imbed the subsystem ID in the fixed format security entity name.

e Move the code for all other security entities outside of the IEBUPDTE JCL, then submit the job to create the *somnode*.CMNZMF.ACF2 PDS.

8 Activate the resource type CMN and compile the members you just created in the custom CNTL member #ACF2. Enter the following commands from the TSO command processing option:

```
ACF2
SET RESOURCE(CMN)
COMPILE 'somnode.CMNZMF.ACF2' ALL STORE
```

9 Check for any super IDs to verify that there are no other logon ID's that have authority to update libraries managed by ChangeMan ZMF. If an ID has NONCNCL authority, it can access any data set or authority level secured for ChangeMan ZMF.

10 If you use the ACF2 TSO Command Limiting feature to restrict execution access to TSO commands, add the following [Product Name Short No Marks] programs to the ACF2 Command Limiting Table:

```
CMNINIT
CMNCISPF
```

This is the format of the table entries:

```
TSOtable CSECT *      REGIONAL ACF2 TABLE
TSOCST ,              TSO RESTRICTED COMMANDS LIST
$TSOCMD CLS           CLEAR SCREEN
$TSOCMD CMNINIT       CHAMGEMAN ZMF INITIALIZATION
$TSOCMD CMNCISPF      SECURE PROGRAM NAMES TABLE
$TSOCEND ,            INDICATE END OF LIST
END
```

If you use the ACF2 Command Limiting Feature and do not make the table entries above, and you attempt to use ZMF, the following message results:

```
IKJ56500I COMMAND XXXXXXXX NOT FOUND
```

CAUTION! Do not add these modules to the TSO Command Table IKJTSOnn in SYS1.PARMLIB. If you do and you attempt to use ZMF, your session will freeze and this message is displayed:

ISPS118L SERVICE NOT INVOKED. A VALID ISPF ENVIRONMENT DOES NOT EXIST.

Add ChangeMan ZMF to IBM Security Server RACF

The steps listed here provide examples specific to IBM Security Server RACF for accomplishing the security setup tasks listed in the "Introduction" on page 80. The following conventions are used in these examples:

- **$CHGMAN** is the general resource class.
- **SERPROC** is the member name of the started procedure.
- **SERTASK** is the jobname assigned to the started task when procedure member **SERPROC** is started. See "SERNET Started Task Names" on page 35.
- **SERUSER** is the RACF user ID for all SERNET instances. If you want different RACF authority for different SERNET instances, assign a unique user ID to each.
- **CMNxADSP** is the procedure member name of the default file tailoring started task, where x is the subsystem ID of the SERNET started task that initiates CMNxADSP. See "Step 11: Build Default File Tailoring Procedure" on page 75.

IMPORTANT! Resource class name $CHGMAN was suggested when you initialized the package master VSAM file in "Step 4: Define ChangeMan ZMF VSAM Files" on page 61. If you use a different resource class name here to satisfy local requirements, do the following:

- Follow the rules for class name syntax in the *z/OS Security Server RACF Security Administrator's Guide*.
- Code the same resource class name in initialization subparameter "RSRCCL" on page 62.

This section is not intended to be an authoritative reference for RACF command syntax. Your security administrator should be aware of the intent of each step and should adjust the sample command syntax if necessary.

1 Define $CHGMAN as a RACF resource class by adding an entry to the dynamic class descriptor table (CDT).

Use this job to execute commands to define the class to the CDT. You can also enter the commands under TSO or through RACF administrative panels.

```
//jobname  JOB (account),'CHGMAN/RACF',
//             CLASS=?,NOTIFY=?,
//             MSGCLASS=?
//* RACF class descriptor table - sample assembly *
//CDTDEF   EXEC  PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD    SYSOUT=*
//SYSTSIN  DD    *
 RDEFINE CDT $CHGMAN UACC(NONE) CDTINFO(DEFAULTUACC(NONE) +
   FIRST(ALPHA) OTHER(ALPHA NUMERIC NATIONAL              +
   SPECIAL) MAXLENGTH(39) POSIT(25)                       +
   RACLIST(REQUIRED) OPERATIONS(NO))
 SETROPTS CLASSACT($CHGMAN)
 SETROPTS GENERIC($CHGMAN)
 SETROPTS RACLIST($CHGMAN)
 SETROPTS RACLIST(CDT) REFRESH
```

In this example:

- The RDEFINE statement declares $CHGMAN as class to RACF and specifies its characteristics.
- The first SETROPTS statement activates the class.
- The second SETROPTS statement allows the specification of generic profiles in this class.
- The third SETROPTS statement activates these definitions immediately.

2 Define the user ID that will be assigned to the SERNET started task:

```
ADDUSER SERUSER NAME('SERNET') OWNER(owner userid) +
    DFLTGRP(group name) DATA('SERNET STARTED TASK')
```

3 Add each SERNET instance to the STARTED class to associate the started task with the user ID.

NOTE With RACF 2.1 and higher, you may define started procedures to the STARTED class rather than adding them to the RACF Started Procedure Table, which requires an IPL. The STARTED class must be active at your site.

Use the following command:

```
RDEF STARTED SERPROC.SERTASK STDATA(USER(SERUSER) +
    GROUP(groupname))
SETROPTS RACLIST(STARTED) REFRESH
```

In the STARTED class, you specify both the started procedure member name and the jobname assigned in the START command. (See "SERNET Started Task Names" on page 35.) You can use a wild card for the jobname:

```
RDEF STARTED SERPROC.** STDATA(USER(SERUSER) GROUP(groupname))
```

or

```
RDEF STARTED SERPROC.SER* STDATA(USER(SERUSER) GROUP(groupname))
```

4 Make an additional entry in the STARTED class for the default file tailoring started procedure CMNxADSP, where x is the subsystem ID of the SERNET started task. Issue the following command:

```
RDEF STARTED CMNxADSP.** STDATA(USER(SERUSER) +
     GROUP(groupname))
```

Assign the same user ID as the SERNET started task that initiates the file tailoring started task.

NOTE Make additional entries in the STARTED class when you create other file tailoring started procedures besides CMNxADSP and enter their names in global administration.

Add ChangeMan ZMF to CA Top Secret

The steps listed here provide examples specific to CA Top Secret for accomplishing the security setup tasks listed in the "Introduction" on page 80. The following conventions are used in these examples:

- **SERPROC** is the member name of the started procedure.
- **SERACID** is the name of ACID associated with the SERNET instance.
- **SERFAC** is the name of the Multiuser Facility created for the SERNET instance.
- **CMNxADSP** is the procedure member name of the default file tailoring started task, where x is the subsystem ID of the SERNET started task that initiates CMNxADSP.

This section is not intended to be an authoritative reference for CA Top Secret command syntax. Your security administrator should be aware of the intent of each step and should adjust the sample command syntax if necessary.

1 Define the SERNET server as a Multiuser Facility, using one of the dummy facility entries in the Facilities Matrix Table. See the CA Top Secret USER GUIDE for information on how to add a new facility.

Example:

```
FACILITY(USERxx=NAME=SERFAC)
FACILITY(SERFAC=PGM=SER)
FACILITY(SERFAC=NOASUBM)
FACILITY(SERFAC=LCFCMD)
FACILITY(SERFAC=UIDACID=7)
```

2 Execute a TSS refresh to implement this parameter change, or perform a temporary change until the next refresh or IPL by executing this command:

```
TSS MODIFY FACILITY(USERxx=NAME=SERFAC)
```

3 After the above TSS command has been completed, query Top Secret to verify that the definition is correct.

```
TSS MODIFY FAC(SERFAC)
```

The following messages are from a successful installation.

```
TSS9550I FACILITY DISPLAY FOR SERFAC
TSS9551I INITPGM=SER ID=T TYPE=013
TSS9552I
    ATTRIBUTES=INUSE,ACTIVE,SHRPRF,NOASUBM,NOABEND,MULTIUSER,NOXD
    EF
TSS9552I ATTRIBUTES=LUMSG,STMSG,SIGN(M),INSTDATA,RNDPW,AUTHINIT
TSS9552I ATTRIBUTES=NOPROMPT,NOAUDIT,RES,WARNPW,NOTSOC,LCFCMD
TSS9552I
    ATTRIBUTES=MSGLC,NOTRACE,NOEODINIT,IJU,NODORMPW,NONPWR,NOIMSX
    TND
TSS9553I MODE=FAIL DOWN=GLOBAL LOGGING=INIT,MSG
TSS9554I UIDACID=8 LOCKTIME=000 DEFACID=*NONE* KEY=8
TSS9556I MAXUSER=03000 PRFT=003
TSS0300I MODIFY FUNCTION SUCCESSFUL
```

4 Create a CA Top Secret started task (STC):

```
TSS CREATE(SERACID) TYPE(USER) NAME('SERACID STC FOR CHGMAN')
FACILITY(STC,BATCH,SERFAC)  DEPT(NAME) PASS(NOPW,0)
```

5 Add the new STC to the Started Task Table:

```
TSS ADD(STC) ACID(SERACID) PROCNAME(SERPROC)
```

6 Make an additional entry in the Started Task Table for the default file tailoring started procedure CMNxADSP, where x is the subsystem ID of the SERNET started task:

```
TSS ADD(STC) ACID(SERACID) PROCNAME(CMNxADSP)
```

Assign the same user ID as the SERNET started task that initiates the file tailoring started task.

Make additional entries when you create other file tailoring started procedures and enter them in global administration.

7 Add access to the facility (SERFAC) for the started task user ID (SERACID) and anyone who will use ChangeMan ZMF:

```
TSS ADD(SERACID) FACILITY(SERFAC)
```

NOTE Repeat this command for each user that requires this access, or put it in a profile to which users are attached.

For more information, refer to the *CA Top Secret OS Reference Guide Volume 3*, Section: TSS Command Function.

Step 3: Define Security Entities

Define security entities used to authorize administrative and package management functions as described in topic "Access to ChangeMan ZMF Functions" on page 40,

including the two subtopics "Administrator and Change Manager Security Entities" and "Mandatory System-Specific Security Entities".

The examples in this section use the values described in this table.

Value	Description
3	Subsystem ID of the SERNET instance running ChangeMan ZMF. The subsystem ID is imbedded in the fixed format security entities, such as CMN3GBAD.
$CHGMAN	General resource class for ChangeMan ZMF
ACTP	Four-character ChangeMan application mnemonic for the Accounts Payable application
APBUSMGR	Security entity for approval of ACTP change packages by the business manager
APPRJMGR	Security entity for approval of ACTP change packages by the project manager
TESTQA	Security entity for promotion to a QA test environment
TESTUT	Security entity for promotion to a unit test environment
USER111	TSO user ID of the ChangeMan ZMF Global Administrator
USER222	TSO user ID of the ChangeMan ZMF Application Administrator for the Accounts Payable application
USER333	TSO user ID of the Accounts Payable Department Manager
USER444	TSO user ID of the IT Project Manager for the Accounts Payable application
USER555	TSO user ID of the QA Test Coordinator who promotes packages to the QA test environment libraries
USER666	TSO user ID of a developer on the Accounts Payable application who creates ACTP packages and changes ACTP components
USER777	TSO user ID of a business analyst in the Accounts Payable department who is allowed to query packages, browse Accounts Payable programs, but not make changes
USER888	TSO user ID of the Operations Supervisor

The steps listed here show you how to define ChangeMan ZMF security entities in IBM Security Server RACF.

NOTE When you define a ChangeMan ZMF security entity in your security system, you define it with no universal access. READ or UPDATE access is established when you grant a user ID or group access to the security entity.

1 Define the five fixed-format ChangeMan ZMF administrative security entities under the $CHGMAN resource class.

```
RDEFINE $CHGMAN CMN3GBAD OWNER(@$CHGMAN) +
    UACC(NONE) APPLDATA('GLOBAL ADMINISTRATOR')
RDEFINE $CHGMAN CMN3LCAD OWNER(@$CHGMAN) +
    UACC(NONE) APPLDATA('LOCAL ADMINISTRATOR')
RDEFINE $CHGMAN CMN3REVR OWNER(@$CHGMAN) +
    UACC(NONE) APPLDATA('CHANGEMAN ZMF REVERT')
```

```
RDEFINE $CHGMAN CMN3BKOU OWNER(@$CHGMAN) +
    UACC(NONE) APPLDATA('CHAMGEMAN ZMF BACKOUT')
RDEFINE $CHGMAN CMN3MON OWNER(@$CHGMAN) +
    UACC(NONE) APPLDATA('MONITOR SCHEDULER LIMBO')
```

2 Permit ChangeMan ZMF administrators and application project managers access to the administration functions. In this example, access is granted to TSO user IDs, but you will permit access to groups for easier maintenance.

```
PE CMN3GBAD CLASS($CHGMAN) ID(USER111) +
    ACCESS(UPDATE)
PE CMN3LCAD CLASS($CHGMAN) ID(USER222) +
    ACCESS(UPDATE)
PE CMN3REVR CLASS($CHGMAN) ID(USER444) +
    ACCESS(UPDATE)
PE CMN3BKOU CLASS($CHGMAN) ID(USER444) +
    ACCESS(UPDATE)
PE CMN3BKOU CLASS($CHGMAN) ID(USER888) +
    ACCESS(UPDATE)
PE CMN3MON  CLASS($CHGMAN) ID(USER222) +
    ACCESS(UPDATE)
PE CMN3MON  CLASS($CHGMAN) ID(USER888) +
    ACCESS(UPDATE)
```

3 Define security entities for applications under the $CHGMAN resource class.

```
RDEFINE $CHGMAN ACTP OWNER(@$CHGMAN) UACC(NONE) +
   APPLDATA('ACCOUNTS PAYABLE APPLICATION')
```

4 Define security entities for package approvals under the $CHGMAN resource class.

```
RDEFINE $CHGMAN APPRJMGR OWNER(@$CHGMAN) +
    UACC(NONE) +
    APPLDATA('ACCOUNTS PAYABLE PROJECT MANAGER')
RDEFINE $CHGMAN APBUSMGR OWNER(@$CHGMAN) +
    UACC(NONE) +
    APPLDATA('ACCOUNTS PAYABLE BUSINESS MANAGER')
```

5 Permit package approvers access to approval security entities.

```
PE APPRJMGR CLASS($CHGMAN) ID(USER444) +
    ACCESS(UPDATE)
PE APBUSMGR CLASS($CHGMAN) ID(USER333) +
    ACCESS(UPDATE)
```

6 Define security entities for package promotion under the $CHGMAN resource class.

```
RDEFINE $CHGMAN TESTUT OWNER(@$CHGMAN) +
    UACC(NONE) +
    APPLDATA('UNIT TEST COORDINATOR')
RDEFINE $CHGMAN TESTQA OWNER(@$CHGMAN) +
    UACC(NONE) +
    APPLDATA('QA TEST COORDINATOR')
```

7 Permit developers and test coordinators access to promotion security entities.

```
PE TESTUT CLASS($CHGMAN) ID(USER666) +
    ACCESS(UPDATE)
PE TESTQA CLASS($CHGMAN) ID(USER555) +
```

```
ACCESS(UPDATE)
```

8 Permit access to applications by application administrators, package approvers, test coordinators, developers, and others.

```
PE ACTP CLASS($CHGMAN) ID(USER222) ACCESS(UPDATE)
PE ACTP CLASS($CHGMAN) ID(USER333) ACCESS(READ)
PE ACTP CLASS($CHGMAN) ID(USER444) ACCESS(READ)
PE ACTP CLASS($CHGMAN) ID(USER555) ACCESS(READ)
PE ACTP CLASS($CHGMAN) ID(USER666) ACCESS(UPDATE)
PE ACTP CLASS($CHGMAN) ID(USER777) ACCESS(READ)
PE ACTP CLASS($CHGMAN) ID(USER888) ACCESS(READ)
```

See "Access to ChangeMan ZMF Functions" on page 40 to see how execution of a ChangeMan ZMF function in an application may require access to both the functional security entity and to the application.

Step 4: Define Data Set Access

Because ChangeMan ZMF manages all development and production libraries, you may want to use the OPERATIONS attribute when you define the user ID for SERNET started tasks that run ChangeMan ZMF.

If you do not use the OPERATIONS attribute, the following sections show what authority is required for libraries and data sets that are required to run SERNET and ChangeMan ZMF or that are managed by ChangeMan ZMF.

Libraries and Data Sets in the SERNET Started Procedure

This table shows you what authority is required for libraries and data sets that are coded in the SERNET started procedure.

Access	Data Set Name	DD Name
UPDATE	somnode.CMNZMF.CMNPMAST	CMNPMAST
UPDATE	somnode.CMNZMF.CMNRECV	CMNRECV
UPDATE	somnode.CMNZMF.CMNCMPNT	CMNCMPNT
UPDATE	somnode.CMNZMF.CMNCMPNL	CMNCMPNL
UPDATE	somnode.CMNZMF.CMNLOG	CMNLOG
UPDATE	somnode.CMNZMF.CMNDELAY	CMNDELAY
UPDATE	somnode.CMNZMF.IADSP	CMNIMPCT
UPDATE	somnode.CMNZMF.IALOG	CMNIALOG
UPDATE	somnode.SERCOMC.TCPIPORT	SER#PARM
READ	somnode.SERCOMC.XMLSPACE	XMLSPACE
READ	somnode.SERCOMC.LICENSE[1]	SERLIC
READ	somnode.SERCOMC.PARMLIB	PARMLIB[2] HPSPLIB

Access	Data Set Name	DD Name
READ	somnode.CMNZMF.LOAD	STEPLIB ISPLLIB
READ	somnode.SERCOMC.LOAD	STEPLIB ISPLLIB
READ	somnode.CMNZMF.PANELS	STEPLIB
READ	somnode.CMNZMF.MESSAGES	ISPMLIB
READ	somnode.CMNZMF.SKELS	ISPSLIB
READ	somnode.CMNZMF.TABLES	ISPTLIB
READ	somnode.CMNZMF.CUSTOM.LOAD	STEPLIB ISPLLIB
READ	somnode.SERCOMC.CUSTOM.LOAD	STEPLIB ISPLLIB
READ	somnode.CMNZMF.CUSTOM.PANELS	ISPPLIB
READ	somnode.CMNZMF.CUSTOM.MESSAGES	ISPMLIB
READ	somnode.CMNZMF.CUSTOM.SKELS	ISPSLIB
READ	somnode.CMNZMF.CUSTOM.TABLES	ISPTLIB
READ	somnode.SISPMENU	ISPMLIB
READ	somnode.SISPTENU	ISPTABL
READ	CA Librarian or CA Panvalet product library (if required)	STEPLIB

1 The LICENSE library and SERLIC DD statement are used only if licenses are not stored in CSA or a load module.

2 This ddname may be specified in the DDNAME= keyword parameter.

Some of these libraries and data sets are also coded in file tailoring started procedures and in batch job JCL created from ChangeMan ZMF skeleton file tailoring.

For information about the libraries and data sets coded in the SERNET started procedure, see ""Step 10: Build SERNET JCL for ChangeMan ZMF" on page 72".

Other Libraries and Data Sets

This table shows you what authority is required for other libraries and data sets that ChangeMan ZMF uses or manages.

Access	Data Set
ALTER/CREATE/ UPDATE/DELETE	[Product Name Short No Marks] utility data sets See "Utility Data Sets" on page 44.
ALTER/CREATE/ UPDATE/DELETE	ChangeMan ZMF package staging libraries See "Staging Library Model Data Set Name" on page 45.
ALTER/CREATE	[Product Name Short No Marks] Baseline libraries These libraries are specified in ChangeMan ZMF application administration. See the *ChangeMan ZMF Administrator's Guide*.

Access	Data Set
UPDATE	Production libraries that [Product Name Short No Marks] will manage These libraries are specified in ChangeMan ZMF application administration. See the *ChangeMan ZMF Administrator's Guide*.
UPDATE	Test libraries [Product Name Short No Marks] will populate with the promotion function These libraries are specified in ChangeMan ZMF application administration. See the *ChangeMan ZMF Administrator's Guide*.
READ	The system procedure library where you store [Product Name Short No Marks] cataloged procedures See "Step 11: Build Default File Tailoring Procedure" on page 75.
READ	Other libraries that contain components that will be brought into a [Product Name Short No Marks] package with the Stage from Development function. See the *ChangeMan ZMF User's Guide*.

Step 5: Add OMVS Segment To Use TCP/IP

TCP/IP Services in z/OS Communications Server requires a z/OS UNIX security context, referred to as an OMVS segment, for the user ID associated with a SERNET instance.

For instructions on satisfying the requirement for an OMVS segment in RACF, see topic "Requirement for an OMVS Segment" in the IBM publication *z/OS Communications Server: IP Configuration Guide*.

Failure to add an OMVS segment results in the following error message during SERNET initialization:

```
SERA000E XCH TCP/IP INITAPI: RC=00001,ERRNO=00156
```

Step 6: Add PassTicket Support In Sernet

RACF PassTickets are a requirement for mainframe clients connecting via TCP/IP.

NOTE RACF PassTickets are not a requirement for ChangeMan ZDD or ChangeMan ZMF for Eclipse. These PassTickets are the result of the RACF Secure Signon Function and eliminate the need for clients to provide a password or passphrase that needs to be sent over a network. Additional information on PassTickets can be found in the 'Using the Secured Signon Function' section of the IBM-supplied 'Security Server RACF Security Administrator's Guide'.

PassTickets are application-specific so a Sernet-generated PassTicket is only valid for connecting to a Sernet started task. If you don't specify in your RDEFINE the parameter APPLDATA('NO REPLAY PROTECTION') then each PassTicket is valid for approximately ten minutes from the time it is issued and can only be used once. For that reason you must specify the APPLDATA('NO REPLAY PROTECTION') parameter.

RACF Administration Required

- Activate the PTKTDATA class by entering:

```
SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
```

- Refresh the PTKTDATA class by entering:

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

- Create a profile in the PTKTDATA class by entering:

```
RDEFINE PTKTDATA SERNET SSIGNON(user_must_choose) APPLDATA('NO
    REPLAY PROTECTION')
```

For further information. refer to the appropriate IBM RACF manual for further information, for example **"Defining Profiles in the PTKTDATA Class"**, in the manual z/OS Security Server RACF Security Administrator's Guide.

NOTE The value of **SERNET** in the above RDEFINE command is mandatory and should not be altered. You must provide the SSIGNON specification.

The SERSET Utility

Support for PassTickets in Sernet is provided via the execution of the SERSET utility. Each time a ChangeMan started task is brought up, during the initialization process, the SETSET utility gets invoked. This caters for the case where clients connect to a started task running on the same LPAR.

However the SERSET utility can also be executed as a batch utility. We provide a new member called SERSET in the delivered SERCOMC CNTL library. If you have a situation where users are signing onto ZMF from an LPAR which does not host a ZMF started task, you need to run the SERSET batch job on this LPAR. This must be done after each IPL to enable a remote connection to ZMF.

Generating a PassTicket

Sernet generates PassTickets when SERCLIEN calls SERXPTIK. The generation process requires authorization (key zero) so SERXPTIK executes as a PC routine and the sole purpose of the SERSET utility is to implement this routine.

For SERCLIEN to generate a PassTicket SERCLIEN only needs to know the PC number associated with SERXPTIK. To find this number SERCLIEN retrieves two system-level tokens, as follows:

```
SerNet.PTickTok - this contains the SerNet.PTickX value.
SerNet.PTickX - this contains the PC number.
```

NOTE The X in this token name corresponds directly to the TOKEN= value established when SERSET runs

Failures in PassTicket Generation

SERXPTIK calls the routine anchored in field RCVTPTGN of the RACF CVT. Errors will be returned to the caller of SERCLIEN with the following message:

```
SER6035E Passticket generation failed, RCVTPTGN RC=nnnn
```

RCVTPTGN and its accompanying return codes are documented under "Using the service to generate a PassTicket" in the *RACF Macros and Interfaces* manual.

TOKEN =Operand of SERSET

The SERSET member of the SERCOMC CNTL library contains an EXEC card that reads:

```
PTICKET EXEC PGM=SERSET,REGION=2M,PARM='TOKEN= '
```

The default value for TOKEN is **A** so this effectively reads:

```
PTICKET EXEC PGM=SERSET,REGION=2M,PARM='TOKEN=A'
```

When the JCL executes one of two message sequences will normally ensue:

```
SER1704I CSVDYLPA loaded SERXPTIK @ xxxxxxxx
SER1708I SerNet.PtickA token created
SER1708I SerNet.PtickTok token created
SER1709I Passticket support enabled
```

This sequence will appear when SERSET first executes after an IPL. It shows the loading of SERXPTIK and the creation of the two system-level tokens.

```
SER1701I Passticket support previously enabled under 'A' suffix
```

This message will appear if SERSET executes any subsequent time after the first execution following each IPL. It signifies that SerNet.PTickTok points at SerNet.PTick**A** and that the latter contains the PC number associated with SERXPTIK.

Refreshing SERXPTIK

SERXPTIK is loaded into common storage by SERSET. By design, it's a very small piece of code that should rarely change but, even so, on occasion it may need to be refreshed without an IPL.

To do this, rerun SERSET ensuring it will pick up the new version of SERXPTIK from STEPLIB and specifying a different TOKEN= value. For example, specifying TOKEN=**B** will result in the following message sequence:

```
SER1704I CSVDYLPA loaded SERXPTIK @ xxxxxxxx
SER1708I SerNet.PtickB token created
SER1708I SerNet.PtickTok  token created
SER1709I Passticket support enabled
```

Once this has executed SerNet.PTickTok will point at SerNet.PTick**B** and this will cause SERCLIEN to invoke the new version of SERXPTIK [via a different PC number].

The TOKEN= parameter will accept any value from A-Z and 0-9 but any use beyond A and, rarely, B would be highly unusual.

Step 7: Set Security for USS File Systems

If you want to use ChangeMan ZDD to access HFS files in Unix System Services on the mainframe, you must make additional entries in your security system.

The instructions here describe commands for z/OS Security Server RACF. If you use CA ACF2 or CA Top Secret, consult with your security administrator to determine the actions they must take in those security systems to accomplish the same objectives.

In the commands that follow, the following conventions are used:

- SERUSER is the user-id assigned to the SERNET/ZMF started task.
- SERGRP is the RACF group assigned to the SERNET/ZMF started task.

1 Assign a non-zero UID to SERUSER by manually assigning the next available value:

```
ALTERUSER SERUSER OMVS(UID(xxx))
```

2 Permit access for SERUSER to two resources so it can manage HFS in USS:

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(SERUSER) ACCESS(UPDATE)
PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV) ID(SERUSER) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
SETROPTS RACLIST(UNIXPRIV) REFRESH
```

3 Ensure that the SERUSER default group SERGRP has a GID:

```
ALTERGROUP SERGRP OMVS(GID(YYY))
```

Chapter 7

Starting, Connecting, and Stopping

This chapter describes how to start a SERNET started task and how to connect to ChangeMan ZMF. It also describes how to stop SERNET, giving ChangeMan ZMF users a grace period to end their sessions.

Start SERNET	98
Connect to the ChangeMan ZMF Primary Option Menu	100
Stop SERNET and ChangeMan ZMF	101

Start SERNET

Use a START command to start a SERNET instance and initialize ChangeMan ZMF. Variations of the START command are:

- `START SERPROC1`
- `S SERPROC.SERTASK2,ID=2`
- `S SERPROC,JOBNAME=SERTASK3,ID=3`

See "SERNET Started Task Names" on page 35 for a discussion of the START command and SERNET started task names.

NOTE You may use z/OS START command parameter REUSASID=YES unless you intend to use ChangeMan ZMF ERO. Example:

S SERPROC,JOBNAME=SERTASK3,REUSASID=YES,ID=3

See "Reusable Address Space Identifiers (ASID)" on page 32.

SERPRINT Messages

SERNET messages, Serena License Manager messages, and some ChangeMan ZMF messages are displayed in the SERPRINT sysout data set for the SERNET started task. SERNET messages are explained in the *ChangeMan ZMF Messages Guide*.

In a SERNET instance running ChangeMan ZMF, messages in SERPRINT have the following formats:

SER*nnnnt*
LIC*nnnnt*
CMN*nnnnt*

nnnn is a unique number among all SERNET messages.

SERPRINT Message Example

The date/time stamp at the left of each line is not shown.

```
SER0800I SerNet - Initialization in progress: CSA=00007000
SER0801I Execution parameters specified:
SER0801I DDNAME=ZMFOPTS
SER0801I < DDNAME: ZMFOPTS
SER0801I   SUBSYS=2
SER0801I   CMN=6395
SER0801I   XCH=6394
SER0801I   ASID=(16,60)
SER0801I   SDNOTIFY=M240
SER0801I   XML=YES
SER0801I >
SER0611I Storage protection is being used
SER0606I SEREX006 Activated
SER0604I The prefix for the dynamic allocation of log datasets is CMNSUP.INTL.MSG
LIC0010I Machine Seq: D5907  Type: 2465  Model: G99  Capacity: 558 MSU
LIC0011I LPAR ID: 18  Name: S004  Capacity: 190 MSU
LIC0012I Licensing version: 4.3  Date/Time: 2016/11/21 00:44:11  Key: 7061ae2a
LIC0045I ChangeMan ZDD licensed to SERENA SOFTWARE
LIC0010I Machine Seq: D5907  Type: 2465  Model: G99  Capacity: 558 MSU
LIC0011I LPAR ID: 18  Name: S004  Capacity: 190 MSU
LIC0012I Licensing version: 4.3  Date/Time: 2016/11/21 00:44:11  Key: 7061e02a
LIC0045I ChangeMan ZMF licensed to SERENA SOFTWARE
SER1018I          The following TCP/IP procedures are active: TCPIP
SER0660I Unix services are available
SER0821I ZDD (ChangeMan ZDD) licensed
SER0821I ECL (ChangeMan Eclipse) licensed
SER1004I XCH      TCP/IP v6 socket acquired
SER1000I XCH      TCP/IP environment active at 151.145.151.106..6394
SER1001I XCH      TCP/IP local host name: S004
SER0835I TCB Manager active
SER0832I Address Space Manager active
SER0833I XML Data Space Manager active
SER0821I ZMF (ChangeMan ZMF) licensed
SER1004I CMN      TCP/IP v6 socket acquired
SER1000I CMN      TCP/IP environment active at 151.145.151.106..6395
SER1001I CMN      TCP/IP local host name: S004
CMN2308I CMNSTART ZMF version: 8.1.2
CMN2308I CMNSTART ZMF version: 8.1.2
CMN2350I CMNSTART *** Warning *** Missing DD Name = SYSEXEC
CMN2300I CMNSTART Initialization in progress; @TCA=15B86000
SER6414I SERXMLDB Header: SERNET XML Dsect Cross Reference. Created: 2 Sep 2016 08:41:14. Version: 813
SER2301I CMNSTART Loading CMN services; @ADR=00008000
CMN2340I CMNSTART Loading active exits:
CMN2340I CMNSTART CMNEXINS loaded
CMN2340I CMNSTART CMNEX001 loaded
CMN2340I CMNSTART CMNEX008 loaded
...
CMN2340I CMNSTART CMNEX201 loaded
LIC0010I Machine Seq: D5907  Type: 2465  Model: G99  Capacity: 558 MSU
LIC0011I LPAR ID: 18  Name: S004  Capacity: 190 MSU
LIC0012I Licensing version: 4.3  Date/Time: 2016/11/21 00:44:17  Key: b07008bf
LIC0045I ChangeMan ZMF licensed to SERENA SOFTWARE
LIC0050I DB2 option licensed.
LIC0050I IMS option licensed.
LIC0050I Info Man option licensed.
LIC0050I Load Balance option licensed.
LIC0050I M+R option licensed.
LIC0050I Online Forms option licensed.
LIC0050I ERO option licensed.
CMN2800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN2.CMNPMAST                    Exclusive
CMN2800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN2.CMNLOG                     Exclusive
CMN2800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN2.CMNRECV                    Exclusive
CMN2800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN2.CMNCMPNT                   Exclusive
CMN2800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN2.CMNCMPNL                   Exclusive
CMN2300I CMNSTART Attach: U=CMNSTART,F1=ATTACH,F2=IADS,@TCA=181AB000,@TCB=8B7950
CMN2391I CMNIAM00 Attempting to open the I/A Dataspace.
CMN2392I CMNIAM00 I/A Dataspace OPEN, LDS: CMNSUP.INTL.CMN2.CMNIADSP
CMN2300I CMNSTART Attach: U=CMNSTART,F1=ATTACH,F2=SSV,@TCA=181DB000,@TCB=8B7638
CMN2385I CMNSSV   Attempting SSV initialization CMNSSV   (MVS-8.1.2) 20160804 21.14
CMN2388I CMNSSV   SSV initialization complete
CMN2300I CMNSTART Attach: U=CMNSTART,F1=ATTACH,F2=INFO,@TCA=181F5000,@TCB=8B70E0
```

(continued on next page)

```
CMN2385I CMNINFSB Attempting INFO Task Connect  CMNINFSB (MVS-8.1.2) 20160729 03.01
CMN2387I CMNINFSB Soap member is undefined.      0008
CMN2399I CMNINFSB INFO Connection disabled
SER0810I SerNet server "2" initialized and ready for communications
SER0824I Attempting to load PAN#1 to determine if feature present; Please ignore any associated CSV003I message.
SER0826I PAN#1 not found
SER0824I Attempting to load LIBR to determine if feature present; Please ignore any associated CSV003I message.
SER0826I LIBR not found
CMN2300I CMNSTART Attach: U=CMNSTART,F1=ATTACH,F2=SERVDLAY,@TCA=18200000,@TCB=897E88
CMN2480I CMNSTART Waiting for HLL exit address space to initialize.
CMN2460I CMNDELAY - START of processing
CMN2468I CMNDELAY - Waiting for resource => qname:rname
CMN2468I CMNDELAY - CHGMAN:CMNSUP.INTL.CMN2.CMNDELAY
CMN2469I CMNDELAY - Resource obtained successfully.
CMN2800I SERS     Opened VSAM file CMNSUP.INTL.CMN2.CMNDELAY
CMN2800I SERS     Closed VSAM file CMNSUP.INTL.CMN2.CMNDELAY
CMN2461I CMNDELAY - END of processing                      00000
CMN2481I CMNSTART Successfully connected to HLL exit address space.
CMN2492I Start of HLLX active exits list:
CMN2494I IntName  Typ Env  External Name        Debug Ids
CMN2495I APRV00XM STD REXX APRV00XM
CMN2495I APRV0103 STD REXX APPV0103
...
CMN2495I PUPD0108 STD REXX PUPD0108
CMN2493I End of HLLX active exits list.
CMN2305I CMNSTART Detach: U=CMNSTART,F1=DETACH,F2=SERVDLAY,@TCA=18200000,@TCB=897E88
CMN2303I CMNSTART Initialization Complete
SER0700I CPU ID: 00000D5907, Company: SERENA SOFTWARE
SER0704I Machine type: 2965, Model: G99, LPAR: S004 / 24
SER0701I Product: 210, Name: ChangeMan ZMF (Unlimited)
SER0701I Product: 213, Name: Client Pack Usrs=0000005
SER0702I Capacity MSU: 550-559, Hours: 1
SER0705I LPAR Capacity MSU: 190-199, Hours: 1
SER0703I Report time: 2016/11/21 01:00, Hours: 1, Key: 14D39346
SER8209I Logon accepted for USER015 Usertyp=ZMF Ccsid=00037 Tca=18201000 UTok=62C0BE2A Uhwm=00001
```

Subsystem IDs

Each SERNET instance started on an LPAR must be assigned a unique subsystem ID. If you attempt to start a SERNET instance with a subsystem ID that is the same as a SERNET instance that is already running, an error message is displayed, and the new started task is terminated.

```
SER0822E SerNet already active for subsystem ID "7" - Terminating
SER0999I SerNet server "7" termination complete, RC=16
```

Connect to the ChangeMan ZMF Primary Option Menu

You can edit ISR@PRIM in the IBM-supplied ISPF panel library to add ChangeMan ZMF as a selection from the main ISPF menu, or you can modify your local ISPF selection menus to invoke ChangeMan ZMF.

The menu selection character invokes the ChangeMan ZMF logon CLIST that contains LIBDEF statements for concatenated ChangeMan ZMF libraries. Do not execute the ISPEXEC SELECT command for program CMNINIT directly from a panel.

If you are running multiple development and production ChangeMan ZMF instances, add a menu selection for each instance. Do not put a selection for your test ChangeMan ZMF instance on a panel available to application developers.

Stop SERNET and ChangeMan ZMF

There are several ways to bring down the SERNET started task. The recommended method is to do an orderly shutdown with a modify command that includes a grace period to allow users to end their work and logoff ChangeMan ZMF. The following table lists some variations of the shutdown command.

Shutdown Command	Result
F SERs,SHUTDOWN	Orderly shutdown, 5 minute grace period
F SERs,SH	Command abbreviation, same result as SHUTDOWN
F SERs,HALT+	Alternate command; same result as SHUTDOWN
F SERs,SH,M10	Orderly shutdown, 10 minute grace period
F SERs,SH,30	Orderly shutdown, 30 minute grace period
F SERs,SH,0	Immediate orderly shutdown, 15 second maximum wait

- SERs is the started task procedure name, the started task identifier, or the JOBNAME specified on the START command. See "SERNET Started Task Names" on page 35.
- The keyword SHUTDOWN may be abbreviated SH.
- The keyword HALT is equivalent to SHUTDOWN and is provided because IBM® NetView® often uses HALT as the directive to bring down long running systems.
- The default grace period is 5 minutes.
- Minutes of grace period may be preceded with M to conform with other hour and minute designations.
- Grace periods longer than 30 minutes are set back to 30 minutes.

NOTE If you specify a grace period, that grace period is only used if there are one or more users signed on to the started task. If the grace period expires and the users have not signed out, they will be gracefully detached and the started task will terminate. However, if the users sign out before the grace period expires, the started task will shut down immediately after the last user signs out and not wait for the full grace period that is specified.

Other methods may be used to stop the SERNET started task if the SHUTDOWN command is not successful.

Command	Result
P SERs	Purge all users in session immediately and stop
F SERs,ABEND	Abend the started task with an S0C3 after closing VSAM files.
CANCEL SERs,DUMP	Cancel the job. **Caution!** CANCEL is **not recommended** because there may be open dialogs along the network and open data sets. Severe damage could result depending on what is occurring at the time this command is executed

Chapter 8

Modifying Skeletons

This chapter tells you how to modify skeletons to enable you to execute key ChangeMan ZMF batch processes.

Introduction	104
System Library and Data Set Names	104
CMN$$DSN - JOBLIB	108
CMN$$SEX - REXX SYSEXEC	108
CMN$$IAX - Impact Analysis Data Extract	109
CMN$PARM - Build Parameters	110
CA ACF2 LOGONID in JOB Statement Skeletons	111
Software Conflicts	111

Introduction

Even if you do not intend to immediately customize ChangeMan ZMF components to fit your change management process, you must modify some delivered skeletons to fit your local conditions or to match the way you set up components in previous chapters in this *Installation Guide*.

This chapter describes how you must change some skeletons to get your first ChangeMan ZMF test or demonstration instance running.

This chapter does not list all skeletons you might need to modify for remote site processing or advanced functions. For a full discussion of customizing ChangeMan ZMF, see the *ChangeMan ZMF Customization Guide.*

You should preserve the version of components that is delivered from Serena. When you modify a skeleton, copy the skeleton from the delivered library to a custom library, then edit the skeleton in the custom library. Concatenate your custom skeleton library in front of the delivered library in ChangeMan ZMF JCL.

System Library and Data Set Names

Most data set names in batch jobs submitted by ChangeMan ZMF are made available to skeleton file tailoring in ISPF variables. The data set names are entered through the user interface by administrators or users, or they are generated internally by ChangeMan ZMF, and they are stored on the package master.

However, a few ChangeMan ZMF data set names are not stored, and most system library names are not stored.

The data set names that are not stored are hard coded in one ChangeMan ZMF skeleton CMN$$DSN with *somnode* as a high level qualifier (HLQ). You must customize these as required to change *somnode* to high level qualifiers that comply with your local standards. This eliminates the need to manually update the many skeletons that use these definitions.

Each entry or group of entries for the SET statements looks like this:

```
)CM  CICS AND DB2 SYSTEM LIBRARIES
)CM  (CMN$$ARE,CMN$$CIC,CMN$$COE,CMN$$MPD,CMN$$MPG,CMN$$PLE)
)CM
)SET SDFHLOD = somnode.CICS.SDFHLOAD
)SET SDFHMAC = somnode.CICS.SDFHMAC
)SET SDSNLOD = somnode.CICS.SDSNLOAD
)SET DB2RNLB = somnode.RUNLIB.LOAD
```

The second CM line lists all the skeletons that use these variables.

Alternatively you can look at the variable use by the function, per the following table. This table lists skeletons that use the variables defined in CMN$$DSN, and lists the DD name

where the variable is used, and the low level nodes of the data set names that must be updated in CMN$$DSN.

TIP The skeletons in the table are listed by function so that you can customize the variables associated with functions that you intend to use immediately, and you can skip or defer work on the rest.

Function	Skeleton where varable is referenced	DDNAME	Library or Data Set
Batch job JCL: JOBLIB and STEPLIB concatenation	CMN$$DSN	&JOBLBDD	somnode.CMNZMF.CUSTOM.LOAD somnode.SERCOMC.CUSTOM.LOAD somnode.CMNZMF.LOAD somnode.SERCOMC.LOAD
Batch job JCL: SER#PARM DSN	CMN$$SPR	SER#PARM	somnode.SERCOMC.TCPIPORT
Batch job JCL: REXX library concatenation	CMN$$SEX	&REXXDD! &REXXDD!	somnode.CMNZMFt.CUSTOM.REX somnode.CMNZMFt.REX somnode.SERCOMCt.CEXEC somnode.CMNZMF.CUSTOM.REX somnode.CMNZMF.REX somnode.SERCOMC.CEXEC
Build: COBOL2 source	CMN$$CO2	STEPLIB	somnode.COB2COM
Build: IBM Enterprise COBOL source	CMN$$COE	STEPLIB	somnode.SIGYCOMP somnode.CICS.SDFHLOAD
Build: CICS precompile	CMN$$CIC	STEPLIB	somnode.CICS.SDFHLOAD
Build: CICS BMS MAP source	CMN$$MPG	SYSLIB	somnode.CICS.SDFHMAC SYS1.MACLIB
Build: CICS BMS MAP source (DESCT)	CMN$$MPD	SYSLIB	somnode.CICS.SDFHMAC SYS1.MACLIB
Build: IBM Enterprise PL/I source	CMN$$PLE	STEPLIB	somnode.SIBMZCMP somnode.SCEERUN somnode.CICS.SDFHLOAD
Build: Assembler source	CMN$$ASM	SYSLIB	SYS1.MACLIB SYS1.MODGEN
Build: C or C++ source	CMN$$CEE	STEPLIB	somnode.SCCNCMP somnode.SCEERUN somnode.SCEEH.H somnode.SDSNC.H
Build: SAS C source	CMNSASC	STEPLIB	somnode.SASC.V450F.LOAD, somnode.SASC.V450F.RDLB.LINKLIB somnode.SASC.V450F.MACLIBC
Build: ORACLE precompiler	CMN$$ORC	STEPLIB SYSLIB	somnode.ORAV60A.CMNDLOAD somnode.ORAV60A.SRCLIB
Build: C prelink	CMN$$PLK	STEPLIB SYSMSGS SYSLIB SYSIN	somnode.SCEERUN somnode.SCEEMSGP(EDCPMSGE) somnode.SCEECPP somnode.SCEELIB(C128) somnode.CBC.SCLBSID(IOC) somnode.CBC.SCLBSID(IOSTREAM) somnode.CBC.SCLBSID(COMPLEX) somnode.CBC.SCLBSID(COLL)

Function	Skeleton where varable is referenced	DDNAME	Library or Data Set
Build: Link edit SYSLIB concatenation	CMN$$SYL	&FIRSTDD! &FIRSTDD! &FIRSTDD! &FIRSTDD! &FIRSTDD!	somnode.SCEESPC somnode.SCEELKED somnode.SCEELKED somnode.SCEELKED somnode.IMS.RESLIB somnode.CICS.SDFHLOAD
Audit	CMN$$AUD	CMNIMPCT	somnode.CMNZMF&SUBSYS..IADSP
Promote: Remote promotion	CMN$$RPM	XNODE &LIBTYPE SEQ&LIBTYPE SEQXNODE SEQ&LIBTYPE SEQXNODE	somnode.&RPMDSN somnode.&STGLIB..&LIBTYPE somnode.&STGLIB..&LIBTYPE somnode.&RPMDSN FRDSN('somnode..&RPMDSN') + TODSN('somnode..&RMTDSN') + FRDSN('somnode..&STGLIB..&LIBTYPE') + TODSN('somnode..&RMTLIB..&LIBTYPE') + somnode.&RMTLIB..&LIBTYPE somnode.&RMTDSN
Install: Remote site	CMN11	CMN11ENQ	somnode.&RMTSITE..&PROJECT..CMN11
Install: Baseline ripple	CMN30	CMNIMPCT	somnode.CMNZMF&SUBSYS..IADSP
Install: IMS Option baseline ripple	CMN30I	CMNIMPCT	somnode.CMNZMF&SUBSYS..IADSP
Install: Update impact analysis	CMN37	CMNIMPCT	SOMNODE.CMNZMF&SUBSYS..IADSP
Backout: Reverse baseline ripple	CMN55	CMNIMPCT	somnode.CMNZMF&SUBSYS..IADSP
Backout: IMS Option reverse baseline ripple	CMN55I	CMNIMPCT	somnode.CMNZMF&SUBSYS..IADSP
Backout: Update impact analysis	CMN57	CMNIMPCT	somnode.CMNZMF&SUBSYS..IADSP
Installation scheduler: CA-7	CMN$$CA7	STEPLIB CA7CMDS BATCHIN BATCHOUT	somnode.CA7.LINKLIB somnode.CA7.COMMDS somnode.CA7.BATCHIN somnode.CA7.BATCHOUT
Installation scheduler: CA-7	CMN$$CAS	 BTIOUPT STEPLIB CA7CMDS BATCHIN BATCHOUT	)SET ENQLIB = somnode.BTIOUPT somnode.BTIOUPT somnode.CA7.LINKLIB somnode.COMMDS somnode.BATCHIN somnode.BATCHOUT
Installation scheduler: CONTROL-M	CMN$$CTM	SYSUT2 SYSUT2 STEPLIB DACHK DALOG DAJOB	somnode.CONTROLM.DATE(&SCHJBNM) somnode.CONTROLM.SCHEDLIB(&SCHJBNM) somnode.CONTROLM.LINKLIB somnode.CONTROLM.DATE(&SCHJBNM) somnode.CONTROLM.LOG somnode.CONTROLM.SCHEDLIB(&SCHJBNM)
Installation scheduler: JOBTRAC	CMN$$JBT	SYSUT2 JOBTRACV	somnode.JOBTRAC.JCLLIB somnode.JOBTRAC.CHKPOINT
Transmission vehicle: Connect:Direct	CMN$$NDM	DMMSGFIL SYSIN	somnode.NDM.MSG somnode.NDM.SIGNON
Transmission vehicle: Connect:Direct	CMN$$PND	DMPUBLIB DMMSGFIL SYSIN	somnode.NDMPLIB somnode.RNDMMID somnode.NDM.SIGNON NETMAP=somnode.RNDMMAP,ESF=YES

Function	Skeleton where varable is referenced	DDNAME	Library or Data Set
Transmission vehicle: NETMASTER	CMN$$N10	XNODE RPMNODE &STGLTYP SEQ&STGLTYP SEQXNODE SEQXRPM	somnode.&DVSNOD somnode.&RPMDSN somnode.&DVSLIB somnode.&DVSLIB somnode.&DVSNOD somnode.&RPMDSN FRDSN('somnode..&DVSNOD') + TODSN('somnode..&PRSNOD') + FRDSN('somnode..&RPMDSN') + TODSN('somnode..&RMTDSN') + FRDSN('somnode..&DVSLIB') + TODSN('somnode..&PRSLIB') TODSP(REP) +
Transmission vehicle: NETMASTER	CMN$$N11	SEQ&STGLTYP SEQXNODE SEQXRPM	somnode.&PRSLIB somnode.&PRSNOD somnode.&RMTDSN
Transmission vehicle: NETMASTER	CMN$$NET	SYSTSIN	somnode.NETMASTR.USERID
Transmission vehicle: XCOM	CMN$$XCD	SYSIN01	somnode.XCOM.SIGNON
Transmission vehicle: XCOM	CMN$$XCP	SYSIN01	somnode.XCOM.SIGNON
Message API	CMN$$NTF	SYSEXEC	SOMNODE.CMNZMF.SAMPLES

Execute these steps to change placeholder *somnode* to the appropriate HLQ.

1. Copy the skeleton CMN$$DSN from your delivered CMNZMF SKELS library to your custom SKELS library.
2. Now CMN$$DSN will allow you to easily change *somnode* with the appropriate data set or library name in your CMNZMF SKELS, reducing the possibility of error.

NOTES

1. The data set name ending in IADSP is the impact analysis VSAM LDS that you created and initialized in "Step 4: Define ChangeMan ZMF VSAM Files" on page 61.
2. The data set name ending in TCPIPORT is the PDS that you allocated in "Step 8: Allocate the SER#PARM TCP/IP Address Library" on page 70.
3. If you will install packages created on a ChangeMan ZMF D or DP instance to production libraries, you must customize skeleton CMN11 to provide a high level qualifier for the data set name at the CMN11ENQ DD statement. This is a temporarily cataloged data set that causes CMN11 jobs for a ZMF instance to single thread.
4. If you license the ChangeMan ZMF IMS Option, you must modify skeletons CMN30I and CMN55I even if you defer installation of the IMS Option until a later time.

CMN$$DSN - JOBLIB

Serena recommends against including ChangeMan ZMF load libraries in the LINKLIST. Skeleton CMN$$DSN also provides a JOBLIB or STEPLIB for all ChangeMan ZMF batch jobs that are generated by file tailoring.

TIP The delivered skeleton contains selection logic to concatenate "test" libraries in front of "production" libraries for a test instance of ChangeMan ZMF. (The lower case "t" designates a "test" environment and "test" libraries.)

For your first ChangeMan ZMF test or demonstration instance, just code a simple JOBLIB /STEPLIB statement. As you build other ChangeMan instances, you can add selection logic that fits your evolving environment.

Follow these steps to customize your JOBLIB skeleton variables.

1. Copy skeleton CMN$$DSN from your delivered CMNZMF SKELS library to your custom SKELS library.
2. Edit CMN$$DSN in your custom SKELS library. Code the same load library concatenation that you used for the STEPLIB in the SERNET.

Example:

```
)CM  THIS DEFINES THE VARIOUS LOAD LIBRARIES USED TO BUILD THE JOBLIB
)CM  CONCATENATION.
)CM  (CMN$$JBL)
)CM
)SET ZMFCTST = CMNTP.S6.V813.CMNZMF.CUSTOM.LOAD
)SET SERCTST = CMNTP.S6.V813.SERCOMC.CUSTOM.LOAD
)SET ZMFVTST = CMNTP.S6.CMNZMF.V8R1M3.LOAD
)SET SERVTST = CMNTP.S6.SERCOMC.V8R1M3.LOAD
)SET ZMFCPRD = CMNTP.S6.V813.CMNZMF.CUSTOM.LOAD
)SET SERCPRD = CMNTP.S6.V813.SERCOMC.CUSTOM.LOAD
)SET ZMFVPRD = CMNTP.S6.CMNZMF.V8R1M3.LOAD
)SET SERVPRD = CMNTP.S6.SERCOMC.V8R1M3.LOAD
```

CMN$$SEX - REXX SYSEXEC

Skeleton CMN$$SEX is delivered to provide a SYSEXEC library concatenation for ChangeMan ZMF batch jobs that are generated by file tailoring.

TIP The delivered skeleton contains selection logic to concatenate "test" libraries in front of "production" libraries for a test instance of ChangeMan ZMF. (The lower case "t" designates a "test" environment and "test" libraries.)

For your first ChangeMan ZMF test or demonstration instance, just code a simple SYSEXEC statement. As you build other ChangeMan instances, you can add selection logic that fits your evolving environment.

Follow these steps to customize your variables to use in the CMN$$SEX skeleton.

1. Edit CMN$$DSN in your custom SKELS library. Code the desired datasets.

Example:

```
)CM  THIS DEFINES THE VARIOUS REXX LIBRARIES USED TO BUILD THE SYSEXEC
)CM  CONCATENATION FOR REXX
```

```
)CM  (CMN$$SEX)
)CM
)SET ZMFCRXT = CMNTP.S6.V813.CMNZMF.CUSTOM.REX
)SET SERCRXT = CMNTP.S6.V813.SERCOMC.CUSTOM.REX
)SET ZMFVRXT = CMNTP.S6.V813.CMNZMF.CUSTOM.REX
)SET ZMFCRXP = CMNTP.S6.V813.SERCOMC.CUSTOM.REX
```

CMN$$IAX - Impact Analysis Data Extract

Skeleton CMN$$IAX builds JCL to run the impact analysis data extract, which creates sequential files BUNSPACE, CMPSPACE, and RELSPACE. These files are then input to the impact analysis LDS load job.

The impact analysis LDS is used by the package audit process and by the online Query Impact and Query BofM functions.

Serena recommends that you define the BUNSPACE, CMPSPACE, and RELSPACE files as generation data groups (GDG).

Execute these steps to customize skeleton CMN$$IAX.

1. Copy skeleton CMN$$IAX from your delivered CMNZMF SKELS library to your custom SKELS library.
2. If you are using GDGs for the BUNSPACE, CMPSPACE, and RELSPACE files, delete job step DELFILES at the top of the skeleton.
3. If you want to display messages that list relationships in which the subordinate component is found in no baseline library or in multiple baseline libraries, replace DUMMY with SYSOUT=* in these DD statements.

   ```
   //LODMISS   DD DUMMY
   //LODSAME   DD DUMMY
   //CPYMISS   DD DUMMY
   //CPYSAME   DD DUMMY
   ```

4. At DD names BUNSPACE, CMPSPACE, and RELSPACE:
   a. Replace the *somnode* placeholder with high level qualifiers that conform to your standards for permanent ChangeMan ZMF sequential data sets.
   b. If you are using GDGs, code a (+1) relative generation.
5. Code the CMNIMPCT DD statement with the name of the impact analysis LDS that you created in "Step 4: Define ChangeMan ZMF VSAM Files" on page 61.
6. Change these XTYP= SYSIN control statements to specify your library type(s) for exclusion:

XTYP=	Component Type	Option Sub-type
FMT	IMS MFS Load	IMS Option library sub-type F
MFS	IMS MFS Source	IMS Option library sub-type M

CMN$PARM - Build Parameters

Build parameters are provided for precompile, compile, prelink, and link skeletons in a set of ISPF variables, which are listed in this table.

Variable Name	Description
&DB2PPRM1	DB2 Precompile Parameters Part 1
&DB2PPRM2	DB2 Precompile Parameters Part 2
&DB2PPRM3	DB2 Precompile Parameters Part 3
&CICSPRM1	CICS Translate Parameters Part 1
&CICSPRM2	CICS Translate Parameters Part 2
&CICSPRM3	CICS Translate Parameters Part 3
&MAPDPRM1	BMS MAP DSECT Parameters Part 1
&MAPDPRM2	BMS MAP DSECT Parameters Part 2
&MAPDPRM3	BMS MAP DSECT Parameters Part 3
&COMPPRM1	Compile Parameters Part 1
&COMPPRM2	Compile Parameters Part 2
&COMPPRM3	Compile Parameters Part 3
&PLNKPRM1	Prelink Parameters Part 1
&PLNKPRM2	Prelink Parameters Part 2
&PLNKPRM3	Prelink Parameters Part 3
&LINKPRM1	Link Parameters Part 1
&LINKPRM2	Link Parameters Part 2
&LINKPRM3	Link Parameters Part 3

Values are assigned to these variables in a layered set of skeletons:

- CMN$$VAR - Initialize build parameter ISPF variables to blank.
- CMN$PARM - Assign system-wide values to build parameter variables according to component language.
- PRM$*aaaa* (where *aaaa* is an application mnemonic) - Overlay system-wide default values with application specific values in optional application skeletons.

In addition, when you execute build processing for a component, you can enter component-specific parameter values that are retained in component history.

Skeleton CMN$PARM is delivered with common build parameter settings for various component language types.

Execute these steps to set ISPF variables for build parameters.

1. Copy skeleton CMN$PARM from your delivered CMNZMF SKELS library to your custom SKELS library.
2. Assign system-wide values to build parameter variables according to language, or code your own selection logic.

3 If necessary, create an optional application skeleton in your CUSTOM SKELS library. Name the skeleton PRM$*aaaa*, where *aaaa* is an application mnemonic. Assign application-level overrides to build parameter variables, only where they are different than system-wide values in skeleton CMN$PARM.

NOTE The comments in skeleton CMN$$VAR provide more details about variables used in ChangeMan ZMF build processing.

CA ACF2 LOGONID in JOB Statement Skeletons

If your security system is IBM Security Server RACF or CA Top Secret, skip this step and go to the next skeleton topic.

If you use CA ACF2 as your security system, execute these steps to enable the LOGONID statement in each JOB statement skeleton.

1 Copy these JOB statement skeletons from your delivered CMNZMF SKELS library to your custom SKELS library.

```
CMN$$D2J
CMN$$JCD
CMN$$JNM
CMN$$NTF
CMN$$RPJ
CMN$$SJN
```

2 In the LOGONID statement in each skeleton, delete the `)CM` operator and the space that follows it so that the LOGONID statement begins in column 1:

```
----+----1----+----2----+----3----+----4----+----5
//*LOGONID CMNBATCH   <--- UNCOMMENT FOR ACF2
```

NOTE If you must use a batch user ID different from CMNBATCH, change each JOB statement skeleton and then modify security exit SERLCSEC. See "CA ACF2 Batch LOGONID" on page 81.

Software Conflicts

Some data center utilities from other software vendor include functions that conflict with facilities in ChangeMan ZMF. You may have to adjust ChangeMan ZMF batch JCL and skeletons to suppress the conflicting functions.

See "Software Conflicts" on page 27 for instructions about changing batch JCL to resolve potential conflicts. Translate those JCL changes into ChangeMan ZMF skeleton modifications.

Chapter 9

Installing Other Facilities

This chapter discusses installing other facilities of ChangeMan ZMF.

Installing XMLSERV	114
Installing CICS NEWCOPY Support	115
Configure Automated Operations Tools	119

Installing XMLSERV

XMLSERV is an XML page prototype tool that you can use to experiment with Serena XML Services.

This facility is intended for use by your ChangeMan ZMF administrator and others who want to execute ChangeMan ZMF functions from outside the ISPF interface. Most ChangeMan ZMF users will not execute XMLSERV.

NOTE You can bring up a test or demonstration instance of ChangeMan ZMF without installing XMLSERV.

See "REXX Runtime Environment" on page 26 for system requirements for this facility.

Follow these steps to configure and install the XMLSERV REXX exec.

1 Allocate user input and output XML libraries for XMLSERV.

Input	Data Set Type:	PDS or PDSE
Recommended last DSN node: XMLIN	Record Length:	255
	Record Format:	VB
Output	Data Set Type:	PDS or PDSE
Recommended last DSN node: XMLOUT	Record Length:	5000
	Record Format:	VB

NOTE Any user who executes XMLSERV must have UPDATE security access to the XML input and output data sets.

2 Copy member XMLSERV from the delivered SERCOMC CLIST library to your custom CLIST library.

3 Edit member XMLSERV in your custom CLIST library. Set the REXX variables in XMLSERV as follows.

Variable	Value
mapds	Code your MAPDATA data set name. You created the MAPDATA sequential file in "Step 5: Define XMLSPACE VSAM LDS" on page 63
serparm	Code the same library name as you coded in the SERNET started task JCL at DD name SER#PARM.
xmlinds	Code the user XML input data set name. Use TSO data-set-name format: enclose the data set name in single quotes if you want it unmodified or leave out the quotes to have the user's TSOID (or ZPREFIX) as the high-level qualifier.
xmloutds	Code the user XML output data set name. Use TSO data-set-name format: enclose the data set name in single quotes if you want it unmodified or leave out the quotes to have the user's TSOID (or ZPREFIX) as the high-level qualifier.

Variable	Value
subsys	Code the subsystem ID of the ChangeMan ZMF started task where you want to connect. The subsystem ID that you code here is the default when you execute XMLSERV, but you can change the subsystem ID on the panel when XMLSERV is running.
sysout	Code the sysout class.
somnode	Change all occurrences of *somnode* to point to your ChangeMan ZMF delivered libraries.
xamver	Do not change.
xmlver	Do not change.
product	Do not change.
test	Do not change. Use T for test mode only under the direction of Serena Technical Support.

When ChangeMan ZMF is running, you can execute XMLSERV from the ISPF Command Shell. For example, using the name of the custom CLIST library from "Step 1: Allocate CUSTOM Libraries" on page 60, execute the following command:

```
EX 'somnode.CMNZMF.VvRrMm.CUSTOM.CLIST(XMLSERV)'
```

CAUTION! XML services are intended for use by ChangeMan ZMF administrators and by technicians responsible for developing custom uses for ChangeMan ZMF internal functions.

1. Do not put the compiled REXX for XMLSERV in a public CLIST or REXX library or in a library included in the SYSPROC or SYSEXEC concatenations for the general population of TSO users.
2. Do not install XMLSERV for a production instance of ChangeMan ZMF.
3. XMLSERV uses module SERXMLBC. Execution of SERXMLAC, SERXMLBC, SERXMLCC and SERXMLRC now require authorization.

Installing CICS NEWCOPY Support

You must install ChangeMan ZMF components in CICS before you can use options OPS, OPQ, and XCI for NEWCOPY. You must also install CICS components that support the CICS external call interface (ECI) before you can use the XCI method.

Read topic "CMNCICS1 - CICS NEWCOPY" in the *ChangeMan ZMF Customization Guide* before you execute the procedures in this section.

If the default transaction IDs (SEMT, SEMQ, and SEML) or Transient Data Queue names in the examples below conflict with existing definitions in your CICS environment, you can change the first three characters of these ZMF resources by using the prefix subparameter of the CMMCICS1 PARM= statement, or by using the PREFIX= keyword parameter that is input to the SYSIN DD statement for CMNCICS1. See subtopics "PARM Options" and "SYSIN Parameters" in topic "CMNCICS1 - CICS NEWCOPY" in the *ChangeMan ZMF Customization Guide*.

Installing NEWCOPY Option OPS

Execute these steps to enable the newcopy option OPS.

1 Define transID SEMT and program CMNEMTP in group ZMFgrp, either manually using CEDA or as input to DFHCSDUP:

```
DEFINE TRANSACTION(SEMT) GROUP(ZMFgrp)
       PROGRAM(CMNEMTP) TWASIZE(0) PROFILE(DFHCICST) STATUS(ENABLED)
       TASKDATALOC(ANY) TASKDATAKEY(USER) STORAGECLEAR(NO)
       RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
       ROUTABLE(NO) PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO)
       RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
       CONFDATA(NO) OTSTIMEOUT(NO) ACTION(BACKOUT) WAIT(YES)
       WAITTIME(0,0,0) RESSEC(NO) CMDSEC(NO)
DEFINE PROGRAM(CMNEMTP) GROUP(ZMFgrp)
       LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
       USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY)
       EXECKEY(USER) CONCURRENCY(QUASIRENT) API(CICSAPI) DYNAMIC(NO)
       EXECUTIONSET(FULLAPI) JVM(NO) JVMPROFILE(DFHJVMPR)
```

2 Install ZMFgrp using CEDA, or add ZMFgrp to a lists of resource definition groups and ensure that the list is in the GRPLIST CICS startup parameter.

3 Either add the CMNZMF LOAD library to the DFHRPL concatenation for the target CICS region (preferred), or copy load module CMNEMTP from the CMNZMF LOAD library to a library in the DFHRPL concatenation.

Installing NEWCOPY Option OPQ

Execute these steps to enable the NEWCOPY option OPS.

1 Define transID SEMQ and program CMNEMTP in group ZMFgrp, either manually using CEDA or as input to DFHCSDUP:

```
DEFINE TRANSACTION(SEMQ) GROUP(ZMFgrp)
       PROGRAM(CMNEMTP) TWASIZE(0) PROFILE(DFHCICST) STATUS(ENABLED)
       TASKDATALOC(BELOW) TASKDATAKEY(USER) STORAGECLEAR(NO)
       RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
       ROUTABLE(NO) PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO)
       RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
       CONFDATA(NO) OTSTIMEOUT(NO) ACTION(BACKOUT) WAIT(YES)
       WAITTIME(0,0,0) RESSEC(NO) CMDSEC(NO)
DEFINE PROGRAM(CMNEMTP) GROUP(ZMFgrp)
       LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
       USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY)
       EXECKEY(USER) CONCURRENCY(QUASIRENT) API(CICSAPI) DYNAMIC(NO)
       EXECUTIONSET(FULLAPI) JVM(NO) JVMPROFILE(DFHJVMPR)
```

2 Install ZMFgrp using CEDA, or add ZMFgrp to a lists of resource definition groups and ensure that the list is in the GRPLIST CICS startup parameter.

3 Either add the CMNZMF LOAD library to the DFHRPL concatenation for the target CICS region (preferred), or copy load module CMNEMTP from the CMNZMF LOAD library to a library in the DFHRPL concatenation.

Installing NEWCOPY Option XCI

For information about activating the EXCI CALL interface on your z/OS sysplex, see the *CICS Transaction Server for z/OS CICS External Interfaces Guide*.

Execute these steps to enable the NEWCOPY option XCI.

1 Define dummy transID SEML, program CMNEMTP, connection SEMC , and session SEMC in group ZMFgrp, either manually using CEDA or as input to DFHCSDUP:

```
DEFINE TRANSACTION(SEML) GROUP(ZMFgrp)
DESCRIPTION(DUMMY TRANID FOR XCI SESSION)
       PROGRAM(DFHMIRS) TWASIZE(0) PROFILE(DFHCICSA) STATUS(ENABLED)
       TASKDATALOC(BELOW) TASKDATAKEY(USER) STORAGECLEAR(NO)
       RUNAWAY(SYSTEM) SHUTDOWN(DISABLED) ISOLATE(YES) DYNAMIC(NO)
       ROUTABLE(NO) PRIORITY(1) TRANCLASS(DFHTCL00) DTIMOUT(NO)
       RESTART(NO) SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
       CONFDATA(NO) OTSTIMEOUT(NO) ACTION(BACKOUT) WAIT(YES)
       WAITTIME(0,0,0) RESSEC(NO) CMDSEC(NO)
DEFINE PROGRAM(CMNEMTP) GROUP(ZMFgrp)
       LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)
       USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(ANY)
       EXECKEY(USER) CONCURRENCY(QUASIRENT) API(CICSAPI) DYNAMIC(NO)
       EXECUTIONSET(FULLAPI) JVM(NO) JVMPROFILE(DFHJVMPR)
DEFINE CONNECTION(SEMC) GROUP(ZMFgrp)
DESCRIPTION(CONNECTION FOR XCI NEWCOPY FUNCTION)
       ACCESSMETHOD(IRC) PROTOCOL(EXCI) CONNTYPE(GENERIC)
       SINGLESESS(NO) DATASTREAM(USER) RECORDFORMAT(U) QUEUELIMIT(NO)
       MAXQTIME(NO) AUTOCONNECT(NO) INSERVICE(YES) ATTACHSEC(IDENTIFY)
       BINDSECURITY(NO) USEDFLTUSER(NO) XLNACTION(KEEP)
DEFINE SESSIONS(SEMC) GROUP(ZMFgrp)
DESCRIPTION(SESSION DEFINITION FOR XCI NEWCOPY FUNCTION)
       CONNECTION(SEMC) PROTOCOL(EXCI) MAXIMUM(0,0) RECEIVEPFX(RG)
       RECEIVECOUNT(4) SENDSIZE(4096) RECEIVESIZE(4096) SESSPRIORITY(0)
       AUTOCONNECT(NO) BUILDCHAIN(YES) USERAREALEN(0)
       IOAREALEN(4096,4096) RELREQ(NO) DISCREQ(NO) NEPCLASS(0)
       RECOVOPTION(SYSDEFAULT)
```

NOTE The definition for transaction SEML uses default CICS profile DFHCICSA because that profile includes INBFMH(ALL).

2 Install ZMFgrp using CEDA, or add ZMFgrp to a lists of resource definition groups and ensure that the list is in the GRPLIST CICS startup parameter.

3 If the batch job that executes CMNCICS1 runs on an LPAR that does not have at least one CICS region with IRC=YES specified, you must generate an appropriate EXCI options module (DFHXCOPT) into a data set that is in the CMNCICS1 JOBLIB or STEPLIB concatenation.

```
//DOIT EXEC PROC=DFHAUPLE,NAME=SDFHEXCI,INDEX='SYS2.CICSTS22.CICS'
//ASSEM.SYSUT1 DD *
         DFHXCO TYPE=CSECT,                                          *
               TIMEOUT=0,             No timeout                    *
               TRACE=OFF,             Only Exception trace entries  *
               TRACESZE=16,           16K trace table               *
               DURETRY=30,            Retry SDUMPS for 30 seconds   *
               TRAP=OFF,              DFHXCTRA - OFF                *
               GTF=OFF,               GTF - OFF                     *
               MSGCASE=MIXED,         Mixed case messages           *
               CICSSVC=216,           EXCI WILL OBTAIN CICS SVC NUMBER *
               CONFDATA=SHOW,         Show user commarea data in trace *
               ABENDBKOUT=NO,         No global backout        @BA17427*
               SURROGCHK=YES          Perform surrogate-user check @P1C
         END   DFHXCOPT
//LNKEDT.SYSLMOD DD DSN=CMNTP.CICSTS22.CICS.SDFHEXCI,DISP=SHR
```

IMPORTANT! For critical information about DFHXCOPT, see the chapter titled "Using the EXCI options table, DFHXCOPT" in the *CICS Transaction Server for z/OS CICS External Interfaces Guide.*

4. Either add the CMNZMF LOAD library to the DFHRPL concatenation for the target CICS region (preferred), or copy load module CMNEMTP from the CMNZMF LOAD library to a library in the DFHRPL concatenation.

5. Include the SDFHEXCI load library in the JOBLIB or STEPLIB concatenation for the CMNCICS1 batch job step.

Configure Automated Operations Tools

Configure your automated operations tools to:

1. Recognize SERNET and ChangeMan ZMF messages written to console log.

2. Issue notifications or execute remedial tasks.

See "Console Log Messages" on page 49.

Chapter 10

Setting Up Housekeeping

This chapter discusses the batch jobs that perform maintenance for ChangeMan ZMF data stores.

Introduction	122
Backup Jobs	122
Restore Jobs	123

Introduction

You must perform periodic maintenance on files used by ChangeMan ZMF, particularly the VSAM files used to store package, component, and user activity information.

Before you implement ChangeMan ZMF in your development and production environments, you must create and install production batch jobs to perform housekeeping functions.

The "Batch Housekeeping Tasks" chapter in the *ChangeMan ZMF Administrators Guide* tells you how to build housekeeping jobs from sample JCL delivered in the CMNZMF CNTL library. That chapter describes the critical housekeeping functions needed to protect and maintain the ChangeMan ZMF instances that manage your application software.

You do not need production housekeeping jobs to bring up a test or demonstration instance of ChangeMan ZMF. Initially, you do not need a full set of housekeeping jobs.

However, soon after you bring up your first instance of ChangeMan ZMF, you should build jobs to back up and restore the VSAM master files to protect the work you do in that environment.

NOTE Proprietary unload/load jobs are recommended here rather than IDCAMS backup/restore because:

- Unload files are required for forward recovery process.
- For LDS, unload and load reorganizes the data and recovers space with physical deletes.
- For LDS, unload and load is the only way to change the physical file allocation.

Backup Jobs

See these topics in the "Batch Housekeeping Tasks" chapter in the *ChangeMan ZMF Administrators Guide* to see how to build the minimum required backup jobs from sample JCL delivered in the CMNZMF CNTL library.

- Unload/Back Up Package and Component Masters
- Unload Impact Analysis LDS
- Back Up And Clear Impact Analysis Log File
- Back Up And Clear the Recovery File
- Back Up And Clear the Log File
- Back Up Staging Versions VSAM Files

Restore Jobs

See these topics in the "Batch Housekeeping Tasks" chapter in the *ChangeMan ZMF Administrators Guide* to see how to build the minimum required restore jobs from sample JCL delivered in the CMNZMF CNTL library.

- Load/Restore Package and Component Masters
- Load Impact Analysis LDS
- Restore Staging Versions Master

Chapter 11

Installing A ChangeMan ZMF P Instance

This provides instructions for building a ChangeMan ZMF P instance.

Introduction	126
Installing P Instance Components	127
Configuring P Instance Security	131
Starting, Connecting, and Stopping	132
Modifying P Instance Skeletons	133
Installing Other P Instance Facilities	133
Setting Up P Instance Housekeeping	134

Introduction

The purpose of a ChangeMan ZMF P instance (environment) is to manage the contents of a set of production libraries that are under the control of ZMF at a production site. P instances are essentially slaves of D or DP instances where development is performed and baseline libraries are maintained.

P instances may reside on the same LPAR as the D or DP instance they work with, on a different LPAR with shared DASD, on a different LPAR with separate DASD, or on a completely separate z/OS machine at a physically distant location.

Differences between a P instance and development environments such as ALL, D, and DP instances include:

- No development activities are performed on a P instance.
- Component history is not maintained on a P instance.
- Impact analysis information is not maintained at a P instance.

Jobs for REXX reporting are the only JCL that is created on a P instance with ISPF skeleton file tailoring.

NOTE In this chapter, "the development instance" refers to the ChangeMan ZMF D or DP instance that manages the packages and components that are installed by the P instance that we are building.

Sharing Libraries and Data Stores

If a P instance resides the same LPAR as the development instance or on a different LPAR with shared DASD, the development instance and the P instance may share some files. This table shows which files can be shared and which cannot.

NOTE The file sharing described in this table also applies to sharing between development instances.

Started Procedure DDname	DSN Last Node	ChangeMan ZMF File	Type	Share	Notes
STEPLIB ISPLLIB ISPMLIB ISPPLIB ISPSLIB ISPTLIB	LOAD LOAD MESSAGES PANELS SKELS TABLES	ZMF delivered CMNZMF libraries ZMF delivered SERCOMC libraries	PDS	Yes	
STEPLIB ISPLLIB ISPMLIB ISPPLIB ISPSLIB ISPTLIB	LOAD LOAD MESSAGES PANELS SKELS TABLES	ZMF custom CMNZMF libraries ZMF custom SERCOMC libraries	PDS	Yes	Note 1
CMNPMAST	CMNPMAST	Package Master	VSAM	No	

Started Procedure DDname	DSN Last Node	ChangeMan ZMF File	Type	Share	Notes
CMNCMPNT	CMNCMPNT	Component Master	VSAM	No	Note 2
CMNCMPNL	CMNCMPNL	Long Name Component Master	VSAM	No	Note 2
CMNRECV	CMNRECV	Recovery File	VSAM	No	
CMNIMPCT	IADSP	Impact Analysis	LDS	No	Note 2
CMNIALOG	IALOG	Impact Analysis Log (fwd. rec.)	QSAM	No	Note 2
CMNLOG	CMNLOG	Activity Log	VSAM	No	
CMNDELAY	CMNDELAY	Delay File	VSAM	No	
	CMNSVSM CMNSVOM CMNSVDM	Staging Versions: Site Master Object Master Delta Master	VSAM	No	Note 2
HPSPLIB		Library containing staging version parameter member	PDS		Note 2
		Library containing SERNET keyword option member	PDS	Yes	Note 3
SER#PARM	TCPIPORT	TCP/IP address/port number library	PDS	Yes	
SERLIC		SER10TY license library	PDS	Yes	
XMLSPACE	XMLSPACE	XML Services data LDS	LDS	Yes	Note 4
	MAPDATA	XML Services data	QSAM	Yes	Note 4

Note 1: Custom libraries can be shared, but the customization must be designed for control by ZMF subsystem ID.
Note 2: This DDname/file is not used with P instances.
Note 3: The DDname for this library is set with the DDNAME= execution parameter for PGM=SERVER in the started procedure.
Note 4: The content of this file must be aligned with the same version of the delivered LOAD libraries.

Installing P Instance Components

To build P instance components, execute the steps in Chapter 5, "Installing ChangeMan ZMF Components" on page 59 with these adjustments.

1 Review "ChangeMan ZMF Libraries" on page 52 and "Step 1: Allocate CUSTOM Libraries" on page 60, then copy your delivered and custom libraries from the site of the development instance to the P site instance if one of the following is true:

- The data set naming conventions on the P instance site are different from the naming conventions on the development site.
- The ZMF delivered and custom libraries running the development instance are not accessible from the P instance site.

- You want a unique set of delivered and custom libraries for each ZMF instance, as you will if you use a phased migration when you upgrade the ChangeMan ZMF software.

2 Execute "Step 2: APF Authorize Delivered and CUSTOM Load Libraries" on page 60 on the P instance site if one of the following is true.

- You created a new set of ZMF delivered and custom libraries for the P instance.
- The P instance is on a different LPAR.

ZMF delivered and custom load libraries must be APF authorized on any LPAR where ZMF runs.

3 Execute "Step 3: Apply ChangeMan ZMF License" on page 61 to create licenses on the P instance site if one of these is true:

- You use a PDS(E) to store ZMF licenses and the license library used by the development instance is not available on the P site.
- You use CSA to store licenses and the P instance is the first ZMF instance in that LPAR.

NOTE ChangeMan ZMF is generally licensed by LPAR. If the new P instance is the first ZMF instance on an LPAR, you may need to contact your Serena account representative to obtain an additional ZMF license.

4 Execute "Step 4: Define ChangeMan ZMF VSAM Files" on page 61 on the P instance site with these changes.

a In job INITIAL1:

- Delete job step LONGDMY.
- In job step IDCAMS, delete DD statements:

  CMNIALOG
  LONGDMY

- In job step IDCAMS, for a P instance, these files are not deleted, only updated.

  CMNCMPNT - Component master VSAM KSDS
  CMNCMPNL - Long name component master VSAM KSDS
  IADSP - Impact analysis LDS

- Adjust VSAM space allocations as follows:

VSAM File	Unit	Pri	Sec
CMNPMAST (Package Master)	CYL	20	5
CMNLOG (Log File)	CYL	10	5
CMNDELAY (Delay File)	CYL	1	1
CMNRECV (Recovery File)	CYL	10	5

CAUTION! Before you submit job INITIAL1, ensure that you are using data set names that will be unique to the new P instance. Make sure that job INITIAL1 will not delete any existing data sets.

b In job INITIAL2:

- Delete these job steps:

  CMPNT - Clears and initializes the component master VSAM KSDS
  CMPNL - Clears and initializes the long name component master VSAM KSDS

- In job step INITPM, change the environment type in the program execution parameter for CMNINTPM to ENVTYP=P.
- Delete job step CMNIAIN0 - Clears and initializes the impact analysis LDS

CAUTION! Before you submit job INITIAL2, ensure that you are using data set names that will be unique to the new P instance. Make sure that job INITIAL2 will not delete any existing data sets.

5 Execute "Step 5: Define XMLSPACE VSAM LDS" on page 63 on page 79 on the P instance site.

6 **Skip** "Step 6: Define Staging Versions VSAM Files" on page 65 on page 80.

7 **Skip** "Step 7: Store SVPARM Control Member" on page 69.

8 If there is no SER#PARM library that is accessible from the P instance site, execute "Step 8: Allocate the SER#PARM TCP/IP Address Library" on page 70 on the P instance site. If a SER#PARM library is accessible from the P instance, use that library for the new P instance.

9 Verify that there is a CMNKEYS members in the TABLES library that you plan to use for the new P instance. See "Step 9: Create ISPF Keylist for EXPAND" on page 70.

10 Skip "Step 10: Build SERNET JCL for ChangeMan ZMF" on page 72 and follow these steps instead to create a started procedure for the new P instance.

a Copy the started procedure for the development instance to a new PDS(E) member for the P instance procedure.

b Delete these DD statements:

```
//CMNCMPNT
//CMNCMPNL
//*CMNCMPNT
//*CMNCMALT
//*CMNCMPNL
//*CMNCLALT
//CMNIMPCT
//CMNIALOG
//HPSPLIB
//HPSIN
```

c Code these DD statements with your delivered and custom ZMF library names for the P instance:

```
//STEPLIB
//ISPPLIB
//ISPMLIB
//ISPSLIB
//ISPLLIB
//ISPTLIB
```

d Change the data set name in the //XMLSPACE DD statement to match the LDS data set name you created in step #5 above.

e Change the data set name at the //SYSMDUMP DD statement to be unique for this started procedure. Create a GDG index for the new data set name.

f Change the data set names in these DD statements to match the data set names you created in step #4 above.

DDname	Last Node of File Name
//CMNPMAST	CMNPMAST
//CMNLOG	CMNLOG,
//CMNRECV	CMNRECV,
//CMNDELAY	CMNDELAY,

g If you use execution parameter DDNAME= for program SERVER in your started procedure, make sure that the library at the specified DD name is available at the P instance site. Copy the member containing SERNET keyword parameters for the ZMF development instance to a new member for the P instance.

h Update the SERNET keyword parameters with the subsystem ID and port number of the P instance.

```
SUBSYS=a
CMN=nnnn
```

See "Passing Parameters to SERNET" on page 36 and Appendix D, "SERNET Keyword Options" on page 151.

i Install the new SERNET started procedure in a system PROCLIB that is available at the P instance site.

See "ChangeMan ZMF P Instance" on page 143 for an example of a finished procedure.

11 Skip "Step 11: Build Default File Tailoring Procedure" on page 75, and follow these steps instead to build a default file tailoring procedure for the new P instance.

a Copy the default file tailoring procedures for the ZMF development instance to a new member for the P instance.

b Change the library names in the concatenation at the SYSEXEC DD statement to match the delivered and custom libraries available at the P instance site.

c Change all other file and library names to match those in the started procedure for the new P instance.

d Install the file tailoring started procedure in a system PROCLIB that is available at the P instance site.

NOTE The only skeleton file tailoring that is performed at a P instance builds JCL for REXX reporting batch jobs. You only need a default file tailoring procedure for a P instance.

12 Skip "Step 12: Build the Logon CLIST" on page 76, and follow these steps instead to build a logon CLIST for the new P instance.

a Copy the logon CLIST for the development instance to a new member for the P instance.

b Change parameter SS(n) to the subsystem ID for the new P instance.

c Change all file and library names to match those in the started procedure for the new P instance.

d Copy the connect CLIST into a system CLIST library that is available at the P instance site.

13 If the data set naming conventions on the P instance site are different from the naming conventions on the development site, execute "Step 13: Modify Exit Program CMNEXINS" on page 77 on the P instance site.

14 Execute "Step 14: APF Authorize Other Load Libraries" on page 78 on the P instance site.

Configuring P Instance Security

To configure security for a P instance, execute the steps in Chapter 6, "Configuring Security" on page 79 with these adjustments.

1 Skip "Step 1: Customize SERLCSEC" on page 80, but ensure that the ZMF custom LOAD library that will run the new P instance contains security interface program SERLCSEC. You do not need a unique version of SERLCSEC for the P instances that work with a particular development instance.

2 If the new ZMF P instance will run under a different instance of your security system than the ZMF development instance, execute the following:

a "Step 2: Add ChangeMan ZMF to Your Security System" on page 82

b "Step 3: Define Security Entities" on page 88

c "Step 4: Define Data Set Access" on page 91

d "Step 5: Add OMVS Segment To Use TCP/IP" on page 93

Starting, Connecting, and Stopping

Review Chapter 7, "Starting, Connecting, and Stopping" on page 97.

SERPRINT Message Example for P Instance

The date/time stamp at the left of each line is not shown.

```
SER0800I SerNet - Initialization in progress: CSA=00007000
SER0801I Execution parameters specified:
SER0801I DDNAME=ZMFOPTS
SER0801I < DDNAME: ZMFOPTS
SER0801I   SUBSYS=7
SER0801I   CMN=6384
SER0801I   XCH=6383
SER0801I   ASID=(16,60)
SER0801I   SDNOTIFY=M240
SER0801I   STAX=NO
SER0801I   CHUNK=255
SER0801I   RLLINIT=16382
SER0801I   RLLTHRSH=2048
SER0801I   RLLCOUNT=8192
SER0801I >
SER0804W Obsolete parameter ignored: CHUNK
SER0611I Storage protection is being used
SER0606I SEREX006 Activated
SER0604I The prefix for the dynamic allocation of log datasets is CMNSUP.INTL.MSG
LIC0010I Machine Seq: D5907  Type: 2465  Model: G99  Capacity: 558 MSU
LIC0011I LPAR ID: 18  Name: S004  Capacity: 190 MSU
LIC0012I Licensing version: 4.3  Date/Time: 2016/11/21 00:44:11  Key: 7061ae2a
LIC0045I ChangeMan ZDD licensed to SERENA SOFTWARE
LIC0010I Machine Seq: D5907  Type: 2965  Model: G03  Capacity: 58 MSU
LIC0011I LPAR ID: 18  Name: S004  Capacity: 190 MSU
LIC0012I Licensing version: 4.3  Date/Time: 2016/11/21 00:44:11  Key: 7061ae2a
LIC0045I ChangeMan ZMF licensed to SERENA SOFTWARE
SER1018I          The following TCP/IP procedures are active: TCPIP
SER0660I Unix services are available
SER0821I ZDD (ChangeMan ZDD) licensed
SER0821I ECL (ChangeMan Eclipse) licensed
SER1004I XCH      TCP/IP v6 socket acquired
SER1000I XCH      TCP/IP environment active at 151.155.151.106..6383
SER1001I XCH      TCP/IP local host name: S004
SER0835I TCB Manager active
SER0832I Address Space Manager active
SER0833I XML Data Space Manager active
SER0821I ZMF (ChangeMan ZMF) licensed
SER1004I CMN      TCP/IP v6 socket acquired
SER1000I CMN      TCP/IP environment active at 151.155.151.106..6384
SER1001I CMN      TCP/IP local host name: S004
CMN7308I CMNSTART ZMF version: 8.1.2
CMN7350I CMNSTART *** Warning *** Missing DD Name = SYSEXEC
CMN7300I CMNSTART Initialization in progress; @TCA=15B86000
SER6414I SERXMLDB Header: SERNET XML Dsect Cross Reference. Created: 2 Sep 2016 08:41:14. Version: 813
SER7301I CMNSTART Loading CMN services; @ADR=00008000
CMN7340I CMNSTART Loading active exits:
CMN7340I CMNSTART CMNEXINS loaded
CMN7340I CMNSTART CMNEX001 loaded
...
CMN7340I CMNSTART CMNEX201 loaded
LIC0010I Machine Seq: D5907  Type: 2965  Model: G03  Capacity: 58 MSU
LIC0011I LPAR ID: 18  Name: S001  Capacity: 19 MSU
LIC0012I Licensing version: 4.3  Date/Time: 2016/11/28 02:32:08  Key: fbcdccdd
LIC0045I ChangeMan ZMF licensed to SERENA SOFTWARE
LIC0050I DB2 option licensed.
LIC0050I IMS option licensed.
LIC0050I Info Man option licensed.
LIC0050I Load Balance option licensed.
LIC0050I M+R option licensed.
LIC0050I Online Forms option licensed.
LIC0050I ERO option licensed.
CMN7800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN7.CMNPMAST
CMN7800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN7.CMNLOG
CMN7800I CMNSTART Opened VSAM file CMNSUP.INTL.CMN7.CMNRECV
CMN7300I CMNSTART Attach: U=CMNSTART,F1=ATTACH,F2=INFO,@TCA=1AA1F000,@TCB=8BFE88
CMN7385I CMNINFSB Attempting INFO Task Connect  CMNINFSB (MVS-8.1.3) 20160729 03.01
CMN7387I CMNINFSB Soap member is undefined.      0008
CMN7399I CMNINFSB INFO Connection disabled
```

```
SER0810I SerNet server "7" initialized and ready for communications
SER0824I Attempting to load PAN#1 to determine if feature present; Please ignore any associated CSV003I message.
SER0826I PAN#1 not found
SER0824I Attempting to load LIBR to determine if feature present; Please ignore any associated CSV003I message.
SER0826I LIBR not found
SER0700I CPU ID: 00000D5907, Company: SERENA SOFTWARE
SER0704I Machine type: 2965, Model: G99, LPAR: S004 / 24
SER0701I Product: 210, Name: ChangeMan ZMF (Unlimited)
SER0701I Product: 213, Name: Client Pack Usrs=0000005
SER0702I Capacity MSU: 50-59, Hours: 1
SER0705I LPAR Capacity MSU: 190-199, Hours: 1
SER0703I Report time: 2016/11/28 02:32, Hours: 1, Key: 5B4198D6
CMN7300I CMNSTART Attach: U=CMNSTART,F1=ATTACH,F2=SERVDLAY,@TCA=1AA2A000,@TCB=8BFCF0
CMN7497I CMNSTART No active HLL exits are defined, the HLLX address space will not be started.
CMN7460I CMNDELAY - START of processing
CMN7468I CMNDELAY - Waiting for resource => qname:rname
CMN7468I CMNDELAY - CHGMAN:CMNSUP.INTL.CMN7.CMNDELAY
CMN7469I CMNDELAY - Resource obtained successfully.
CMN7800I SERS     Opened VSAM file CMNSUP.INTL.CMN7.CMNDELAY                    Exclusive
CMN7800I SERS     Closed VSAM file CMNSUP.INTL.CMN7.CMNDELAY
CMN7461I CMNDELAY - END of processing                      00000
```

Modifying P Instance Skeletons

The only skeleton file tailoring that is initiated by a P instance is for building REXX reporting batch jobs. Therefore, the only skeleton that must be customized for a P instance is:

- CMN$$DSN

Ensure that the ZMF custom SKELS library that will run the new P instance contains CMN$$DSN. Check the custom skeleton to make sure it will create the JCL you want for the P instance.

NOTE Most customers can use D and DP instance skeletons on a P instance by using the value of variable &SUBSYS to select instance-specific library names and data set names.

Installing Other P Instance Facilities

See Chapter 9, "Installing Other Facilities" on page 113.

1. Skip "Installing XMLSERV" on page 114. XMLSERV is a Serena XML Services prototypingL tool that is inappropriate for a P instance.
2. If you intend to have ChangeMan ZMF execute CICS NEWCOPY for programs that are installed by the new P instance:
   a. Review topic "CMNCICS1 - CICS NEWCOPY" in the ZMF 8.1.2 Customization Guide.
   b. Execute the steps in "Installing CICS NEWCOPY Support" on page 115 of this book to enable the ChangeMan ZMF CICS NEWCOPY facility in the CICS region that uses the production libraries managed by the new P instance.

Setting Up P Instance Housekeeping

Review Chapter 10, "Setting Up Housekeeping" on page 121, especially the "Introduction" on page 122. Build the batch housekeeping jobs for the new P instance that you require.

Housekeeping jobs for these files are not required because P instances do not use these files:

- CMNCMPNT -Component Master
- CMNCMPNL - Long Name Component Master
- IADSP - Impact Analysis
- CMNIALOG - Impact Analysis Log
- CMNSVSM - Staging Versions
- CMNSVOM - Staging Versions
- CMNSVDM - Staging Versions

Appendix A

Manual File Transfer and Expand

This appendix tells you how to manually transmit XMIT format files from a networked PC to a mainframe host and how to expand those files into PDS libraries.

Use the procedures in this appendix if you have problems with the automated procedures described in Chapter 4, "Unloading the Software" on page 51.

Transfer Files To Host Mainframe	136
Expand Host Target Files	139

Transfer Files To Host Mainframe

After the Setup Wizard has copied and decompressed the XMIT format mainframe software files to your PC or a network drive, you copy the files to the mainframe.

The transfer process must not convert the file into character format; the transfer must be binary. The receiving mainframe files has these characteristics:

File Organization	Sequential
Record Format	Fixed block
Record Length (LRECL)	80
Block Size	3120

Choose the transfer procedure that you prefer:

- To use FTP, see "File Transfer Using FTP" on page 136.
- To use 3270 emulator software, see "File Transfer Using 3270 Emulator" on page 137.

File Transfer Using FTP

This section describes how to use FTP to upload ChangeMan ZMF files from a PC to a mainframe host computer. These instructions assume that you have:

- TCP/IP FTP connectivity from your PC to the mainframe computer.
- The dotted decimal IP address of the mainframe.
- A mainframe user ID and password.
- Security authorization to allocate files on the mainframe.

Execute these steps on your PC:

1 Open a Command Prompt window:

Windows NT®:	**Start > Programs > Command Prompt**
Windows® 2000:	**Start > Programs > Accessories > Command Prompt**
Windows® XP:	**Start > Programs > Accessories > Command Prompt**
Windows® 7:	**Start > All Programs > Accessories > Command Prompt**

2 At the Command prompt, type the following command to start FTP and connect to the host mainframe. Specify the dotted decimal IP address of the target mainframe computer:

```
ftp nnn.nnn.nnn.nnn
```

Press **ENTER**.

3 When you are prompted for your user ID, type your mainframe user ID and press **ENTER**.

4 When you are prompted for your password, type your mainframe password and press **ENTER**.

5 At the FTP prompt, type the following command to set the transfer type to binary:

```
binary
```

Press **ENTER**.

6 At the FTP prompt, type each of these commands and press **ENTER** in the order shown:

```
literal site recfm=fb lr=80 blocksi=3120 cylinders
```

These commands set the DCB and the SPACE units for the receiving files that are automatically allocated on the mainframe.

7 At the FTP prompt, type this command to transfer a file from the PC to the mainframe host. (The sample command here is too long to display on this page without artificial breaks. Long command lines like this one will wrap in the **Command Prompt** window.)

```
put "C:\Program Files\Serena\
    ChangeMan ZMF Server\SERCOMC.V5R5M0.ASMSRC"
    'user111.binary.sercomc.v5r5m0.assemble'
```

The first file name is the PC file to be transmitted. Enclose the PC file name in double quotes if there are spaces in a directory name or in the file name. The PUT command is not case sensitive.

The second file name is the receiving file on the mainframe computer. Enclose the mainframe file in single quotes to suppress the addition of the "working directory" (user ID) as a high level qualifier.

NOTE The mainframe file you specify in the PUT command is not a PDS library. Specify an intermediate file name rather than a data set name you intend to use for a ChangeMan ZMF PDS library on the mainframe.

Press **ENTER**.

8 Type a PUT command at the FTP prompt and press **ENTER** for each file to be transferred to the host.

9 After you have transferred all files from the PC to the host mainframe computer, type the following command at the FTP prompt to disconnect from the mainframe computer and end the FTP session.

```
quit
```

Press **ENTER**.

10 Close the **Command Prompt** window.

11 Logon to the mainframe and verify that the new mainframe files have the proper record format, logical record length, and block size.

File Transfer Using 3270 Emulator

Extra!® from Attachmate Corporation and IBM Personal Communications are popular 3270 emulator programs. These and other 3270 emulator programs commonly use IND$FILE to transfer files between a PC and a mainframe host computer.

Execute these steps if you use 3270 emulator software on your PC to transfer ChangeMan ZMF files to the host mainframe computer:

1 In the File Transfer facility of your PC host emulator software, change the TSO binary transfer type to use these settings, or define a new TSO Binary transfer type:

Setting	Value
VM / TSO / CICS	TSO
Transfer Type	Binary
ASCII	NO
CRLF	NO
APPEND	NO
Record Format	Fixed (block)
Record Length	80
Block Size	3120
Allocation Units	Cylinders
Primary Allocation	1
Secondary Allocation	1

2 With the preceding settings, the file transfer software should allocate a sequential target file on the host for each file transfer. However, if you wish to preallocate one or more host target files, use these file allocation parameters:

```
SPACE=(CYL,(1,1)),
DCB=(DSORG=PS,RECEM=FB,LRECL=80,BLKSIZE=3120)
```

3 Run the file transfer function to transfer each file on the PC to a target sequential disk data set on the host.

NOTE The mainframe file you specify in the transfer facility is not a PDS library. Specify an intermediate file name rather than the data set name you intend to use for the ChangeMan ZMF PDS library on the mainframe.

4 Logon to the mainframe and verify that the new mainframe files have the proper record format, logical record length, and block size.

Expand Host Target Files

Execute these steps to expand the sequential XMIT files transferred from your PC into PDS libraries:

1 Choose Option **6** (Command) from the **ISPF/PDF Primary Option Menu**.

2 On the **ISPF Command Shell** panel, type the TSO RECEIVE command:

```
RECEIVE INDATASET('host.target.dataset.name')
```

Specify one of the sequential XMIT files that you uploaded from the PC.

3 The RECEIVE command prompts you with these messages.

```
INMR901I Dataset USER11.CMN512.GOLDCODE.CNTL.FIX from...
INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

Respond by typing a DATASET parameter to specify the data set name you chose for ChangeMan ZMF delivered libraries.

```
DATASET('node.SERCOMC.VnRnMn.LOAD') NEW|OLD
```

4 The RECEIVE command displays IEBCOPY sysout messages.

NOTE Serena recommends that you define the last node in your PDS library names the same as the last node of the XMIT files you are expanding.

If you need more information about using the RECEIVE receive command, type the following on the Command line.

```
TSO HELP RECEIVE
```

Appendix B

JCL and CLIST Examples

This appendix presents sample JCL and CLISTs from the distributed libraries.

Introduction	142
SERNET Started Procedure With ChangeMan ZMF	142
ChangeMan ZMF Logon CLIST	142
CMNxADSP Default File Tailoring Procedure	142
ChangeMan ZMF P Instance	143

Introduction

The following samples are in the CMNZMF and SERCOMC libraries on the ZMF installer. They should be modified according to instructions in this manual.

SERNET Started Procedure With ChangeMan ZMF

Model JCL for a SERNET started task is delivered in member SERVER in the delivered SERCOMC CNTL library.

Model JCL for ChangeMan ZMF DD statements is delivered in member STARTJCL in the delivered CMNZMF CNTL library.

ChangeMan ZMF Logon CLIST

Copy and customize member CMN from the CMNZMF CLIST library. This CLIST connects a user to the ChangeMan ZMF instance defined by the subsystem ID specified.

CMNxADSP Default File Tailoring Procedure

Model JCL for file tailoring started procedures is delivered in member CMNADSP in the delivered CMNZMF CNTL library.

To facilitate parallel processing, ChangeMan ZMF global administration allows you to have up to four different file tailoring procedures. However, the JCL in all of those procedures should be identical.

ChangeMan ZMF P Instance

The started procedure for a ChangeMan ZMF 8.1 P instance omits some DD statements included in the started procedures for ALL, DP, and D instances. Refer to the comments in the supplied CNTL members.

Appendix C

Applying Maintenance Releases

This appendix discusses the procedures for applying a maintenance release to an existing ChangeMan instance.

Introduction	146
Updating Delivered Libraries	146
Updating Custom Libraries	147
Managing Copybook Changes	148
Binding DB2 DBRM	148
Managing XML Services Changes	149
Synchronizing Impact Analysis Data	149
Updating Housekeeping Jobs	150

Introduction

For any ChangeMan ZMF maintenance release, there are some customary actions that you execute to apply changes to your existing ZMF instances.

This Appendix describes those tasks. See the Readme for tasks that apply to the specific maintenance release you wish to install.

Updating Delivered Libraries

The ChangeMan ZMF installer for this maintenance release contains both the Full version and the Fix version and also a Delta version of the maintenance release software. When you run the installer, you are prompted for the version that you want to install.

- The Full version contains all components for ChangeMan ZMF, including components that are new and changed in maintenance releases since the major version release.

  Customers upgrading from a prior version must install the Full version. For example, if you are currently running ZMF 7.1.x and want to install ZMF 8.1.x, you must install the Full version.

- The Fix version contains only those components that are added or changed in maintenance releases since the major version release.

- The Delta version contains only those components that have been added or changed since the immediately prior GA version of the release. As an example, the Delta version delivered with ZMF 8.1.2 contains only components updated since 8.1.1.01.

  If you are upgrading from ZMF 6.1.x or ZMF 7.1.x to ZMF 8.1, you must install the Full version.

NOTE Maintenance releases contain all patch releases since the last maintenance release. For example, ZMF 8.1.3 contains all patch releases GA'd between ZMF 8.1 and ZMF 8.1.3.

Execute these steps to update your delivered libraries with components changed in the maintenance release:

1. Before you update your delivered libraries, make a copy of your current delivered libraries to use in a later task.
2. If you are installing a Fix version, copy the contents of the maintenance release libraries into your delivered libraries, replacing like-named members.
3. If you are installing a full version, replace your current delivered libraries with the maintenance release libraries.

Updating Custom Libraries

The *ChangeMan ZMF Installation Guide* tells you to allocate these custom libraries.

CMNZMF	SERCOMC
ASMSRC	ASMSRC
CLIST	LOAD
CNTL	
PANELS	
LOAD	
MESSAGES	
REX	
SKELS	

After you update your delivered libraries, follow this procedure to update your custom libraries:

1. Allocate a new set of custom libraries.
2. Copy any custom components that are all original code into your new custom libraries.
3. If you use the ChangeMan ZMF M+R Option:
   - **a** On the Versions panel, define your base and derivative libraries as follows:

     Base: Delivered library for old version

     Drv1: Custom library for old version

     Drv2: Delivered library for new version
   - **b** On the Consolidation Workplace panel, reconcile your old customization with the new maintenance release components.
   - **c** When you have reconciled all of your customization, export reconciled components to your new custom library.
4. If you do not use the ChangeMan ZMF M+R Option:
   - **a** Determine what components are changed in the maintenance release by comparing delivered libraries before and after updates for the maintenance release.
   - **b** Find components in your custom libraries that are changed in the maintenance release.
   - **c** Determine whether your modifications will be valid in the maintenance release.
   - **d** Copy the new delivered version to a new custom library and reapply your modifications.
5. Assemble custom source members into your custom load library.

IMPORTANT! Some changes in a maintenance release may make your modifications obsolete or require you to change your modifications.

Managing Copybook Changes

Occasionally, copybooks delivered in the CMNZMF ASMCPY library or the SERCOMC ASMCPY library are changed in a maintenance release.

Follow this procedure to discover copybook changes and rebuild load modules as necessary.

1. Compare you current delivered copybook libraries to the libraries delivered in the maintenance release.
   a. CMNZMF ASMCPY
   b. SERCOMC ASMCPY
2. If you customized exit programs in the CMNZMF ASMSRC library or the SERCOMC ASMSRC library that use a copybook that has changed, assemble those exits using copybook libraries from the new release, even if you do not change your customization in the program source.

   ASMCPY members CMNEX* and SEREX* are included in the ASMSRC members with the same name. For example, copybook CMNEX001 is contained in source member CMNEX001. So, if copybook CMNEX001 is changed, you must reassemble a *customized* exit CMNEX001.
3. The SERCOMC ASMCPY library contains COBOL copybooks with member names XMLC****. If you have written custom COBOL programs that use XMLC**** copybooks that have changed in the new release, recompile those custom programs using copybook libraries from the new release, even if you do not change your COBOL program source.

Binding DB2 DBRM

If you license the ChangeMan ZMF DB2 Option or the ERO Option, you must bind the changed DBRM delivered in a maintenance release.

This table lists all DB2 DBRM delivered with the ChangeMan ZMF DB2 Option and the ERO Option.

DBRM	Option	BIND JCL	Function
CMNDB2AD	DB2	DB2OPTN	DB2 Option
CMNDB2AT	DB2	DB2OPTN	DB2 Option
CMNDB2RQ	ERO	CMNDB2RP	Release management
CMNDB2SQ	DB2/ERO	BINDPKG	DB2 option PLAN lookup
CMNDB2VB	DB2	DB2OPTN	DB2 Option
CMNIACVT	DB2	CMNIACVB	I/A conversion 6.1-7.1
CMNRACHK	ERO	CMNRAUTB	Check ERO I/A table
CMNRAHAR	ERO	CMNRAUTB	Shrink history table
CMNRARPT	ERO	CMNDB2RP	Release audit

CMNRARTM	ERO	CMNDB2RP	Release audit
CMNRA905	ERO	CMNDB2RP	Release audit

IMPORTANT! The DBRM for CMNDB2SQ must be bound into DB2 packages in all DB2 subsystems where the ChangeMan ZMF DB2 Option performs binds or manages stored procedures, user defined functions, or triggers. Refer to the DB2 GSG for further details on bind requirements.

If your library compares indicate that DBRMLIB members are changed in the maintenance release, take these actions:

1. If you license the DB2 Option, follow the instructions in the *ChangeMan ZMF DB2 Option Getting Started Guide* to bind changed SQL into a DB2 package.
2. If you license the ERO Option, follow the instructions in the *ChangeMan ZMF ERO Getting Started Guide* to bind changed SQL into DB2 packages.

NOTE DBRM for ChangeMan ZMF programs are created with DB2 precompiler parameter VERSION(AUTO), so you can bind the SQL before you actually install the load modules for the new versions of the programs.

Managing XML Services Changes

Most ChangeMan ZMF functions are executed by XML Services, even when you use the ISPF interface.

Follow this procedure to upgrade XML Services for the new release unless the Readme tells you that XML Services have *not* changed and that the procedure is not required.

1. If you are currently using XML Services and/or have written programs that use XML Services, contact Support for a version of the **XML Delta Report** that identifies which XML Services have changed between your current release and the new maintenance release. Use this report to determine if you are using any of the XML Services that have changed, and then plan this part of your upgrade accordingly.
2. Run the job in member XMLLOAD in your custom CNTL library to delete, define, and reload the XMLSPACE VSAM LDS and the MAPDATA sequential file with the new data in member XMLDATA in the delivered SERCOMC CNTL library.

NOTE Load the new data at the same time that you start using the new load modules delivered with the release. If you back out the upgrade, restore the previous contents of the XMLSPACE VSAM LDS and the MAPDATA sequential file.

Synchronizing Impact Analysis Data

To ensure the integrity of the audit function, run a full impact analysis maintenance process after you apply the maintenance release. See topic "Generating and Updating

Impact Analysis Data," in Chapter 4, "Setting Up Global Administration" in the *ChangeMan ZMF Administrator's Guide*.

Updating Housekeeping Jobs

Updates to ChangeMan ZMF components sometimes result in changes to the sample JCL delivered in the CNTL file.

If you have built housekeeping jobs from the sample JCL in the CNTL library, follow this procedure to update your housekeeping jobs:

1. Compare the delivered CNTL library for your current ChangeMan ZMF release to the delivered CNTL library for the new release.
2. Use the table in topic "Sample JCL / Housekeeping Job Cross Reference" at the end of the "Batch Housekeeping Tasks" chapter in the *ChangeMan ZMF 8.1.2 Administrator's Guide* to see what housekeeping jobs use the sample JCL that has changed.
3. Update your housekeeping jobs with the changes in the sample JCL in the new release.

Appendix D

SERNET Keyword Options

This appendix describes keyword options that may be used with a SERNET started task.

General rules for SERNET keyword options:

- SERNET keyword options may be input in any order.
- The syntax for SERNET keyword options is flexible. Any of the following formats is acceptable:

```
keyword(value)
keyword(value1,value2),...)
keyword=value
keyword=(value1,value2,...)
```

- To preserve lower-case characters in a keyword value, enclose the value in single or double quotes. Examples:

```
NTFYURL='tt_server.serena.com:8080/zmfws/...'
NTFYURL="tt_server.serena.com:8080/zmfws/..."
```

When keyword options are coded in a file for input to a SERNET started task, these additional rules apply:

- One or more keyword options may be coded in a single record.
- Multiple keyword options are separated by spaces, commas, semicolons, or any combination of these.
- A slash (/) or asterisk (*) marks the start of a comment, and the rest of the record is ignored.

apl

Purpose	Starts a SERNET application and/or specifies a TCP/IP port number.
Application(s)	All
Format	apl[=port]
Valid Values	apl CMN ChangeMan ZMF CPX Comparex STR StarTool SYN ChangeMan SSM XCH ChangeMan ZDD XCHMSG ChangeMan ZDD/Messenger port Number between 1024 and 65535. These `apl` values require a port number: CPX STR XCH XCHMSG If port number is not specified, cross memory services are used instead of TCP/IP. Do not use the same port number for multiple applications.
Default Value	There is no default value for `apl`. If port number is not specified, cross memory services are used.
When Required	Always
Comments	Formats from previous versions of SERNET are compatible, but APPC application name is ignored. apl=([APPCaplname],port)
Examples	CMN=6000 Starts ChangeMan ZMF with port 6000 CMN Starts ChangeMan ZMF using cross-memory services only SYN Starts ChangeMan SSM XCH=5000 Starts ChangeMan ZDD with port 5000

ASID

Purpose	Specifies an override to the default maximum number of concurrent address spaces and time-out minutes.
Application(s)	ChangeMan ZMF
Format	`ASID=(n,m)` where: n Maximum number of concurrent address spaces m Time-out in wall clock minutes
Valid Values	n Numeric, maximum 256 concurrent address spaces m Numeric, maximum 255 wall clock minutes
Default Value	Omitting this parameter is the same as `ASID=(16,20)`

Comments	The maximum number of address spaces is checked when an application needs to start an address space for an asynchronous operation. An example of an asynchronous process that uses a separate address space is ChangeMan ZMF ISPF file tailoring for install JCL. SERNET periodically checks the list of address spaces that it has started, and if an address space runs longer than the time-out limit, it is terminated. Such an address space may be in a loop or have some other problem, and it can prevent another address space from starting when the maximum number of address spaces is reached. The ASID time out limit is similar to the TIME parameter for an ordinary job except it is wall clock time, not CPU time. The elapsed run time for an address space is calculated as the difference between the store clock (STCK) value taken when the address space was created and the store clock value taken when the check is made. The maximum number of address spaces is 256, and the maximum number of time-out minutes is 255, but whatever you input is displayed in SERPRINT. If you enter a number of minutes greater than 255, a time-out limit of 255 is used, which is 4 hours 15 minutes.

ASINITi

Purpose	For an initiator, specifies an override to the default maximum number of concurrent address spaces and the job classes assigned to an initiator.
Application(s)	ChangeMan ZMF
Format	`ASINITi=(n,c)` where: i Initiator number n Maximum number of concurrent address spaces that can be created c Job classes assigned to the initiator
Valid Values	n Numeric, maximum 256 concurrent address spaces
Default Value	
Comments	The address space limit is checked when an application needs to start an address space for an asynchronous operation. An example of an asynchronous process that uses a separate address space is ChangeMan ZMF ISPF file tailoring for install JCL. The maximum number of address spaces is 256, but whatever you input is displayed in SERPRINT.

AUTOMESSENGER

Purpose	Tells SERNET whether to automatically append a Messenger JCL fragment onto submitted batch jobs and comment out any NOTIFY= parameter found in JOB statements.
Application(s)	ChangeMan ZDD/Messenger
Format	AUTOMESSENGER=[NO|YES|NOTIFY]
Valid Values	NO Do not append the Messenger JCL YES Append the Messenger JCL unless it is already present NOTIFY Append the Messenger JCL if a NOTIFY=userID parameter is found within the first four JOB statement images and the Messenger JCL is not already present. If these conditions are met, also comment out the NOTIFY=userID.

Default Value	Omitting this parameter is the same as AUTOMESSENGER=NO
When Required	

Comments	The Messenger JCL fragment is delivered in SERCOMC.CNTL($SERNTFY) on the SERNET distribution tape, and it is copied to a public PROCLIB. Treatment of JOB statement parameter NOTIFY=userID specified in the keyword parameter AUTOMESSENTER may be overridden by code in exit program SEREX002. The Messenger JCL fragment contains these statements:
	`//  EXEC $SERNTFY`
	`//*`
	`//*`
	`//*`

COMPRESS

Purpose	Determines whether compression is forced or dynamic when data is transmitted through TCP/IP.
Application(s)	All applications running under the SERNET started task
Format	COMPRESS=[0|1] COMPRESS=[NO|YES]
Valid Values	0,NO Dynamic compression. Large messages are compressed, but small messages (under 20K) are not compressed. 1,YES Compression is forced. Every response from z/OS is compressed at the highest level regardless of size.
Default Value	Omitting this parameter is the same as COMPRESS=0
When Required	
Comments	The overhead for compressing short messages is high, so COMPRESS=0 is most efficient. Use COMPRESS=1 to encrypt all messages.

CONNECTCHECK

Purpose	Activates a SERNET security check to allow logon to ChangeMan ZDD or ChangeMan ZMF only if the user ID or group has READ access to a FACILITY class profile.
Application(s)	ChangeMan ZDD, ChangeMan ZMF
Format	CONNECTCHECK=[NO|YES]
Valid Values	NO Do not perform a security check to see who is allowed to logon to ZDD or ZMF. YES Perform a security check to see who is allowed to logon to ZDD or ZMF.
Default Value	Omitting this parameter is the same as CONNECTCHECK=NO

When Required	Never
Comments	If the security check is enabled with CONNECTCHECK=YES, the user ID must have READ access to these FACILITY class profiles to logon to ZMF or ZDD respectively: SERNET.CONNECT.*sysname*.XCHsubsys SERNET.CONNECT.*sysname*.CMNsubsys Where: *sysname* is the four-character SMF ID of the LPAR where the SERNET instance runs. *subsys* is the one-character subsystem ID of the SERNET started task.

CONAUTH

Purpose	Turns on a security check that restricts access to a ChangeMan ZDD server or a ChangeMan ZMF instance.
Application(s)	ZDD and CMN
Format	CONAUTH=[Y|N]
Valid Values	Y Perform check for READ access to resource N Do not perform check for READ access to resource.
Default Value	Omitting this parameter is the same as CONAUTH=N
When Required	Not required
Comments	The RACF resources that are checked are named for the application and for the subsystem ID of the SERNET started task: Class: FACILITY Resource: SERNET.CONNECT.aaas Where: aaa is the application s is the SERNET subsystem ID. Examples: SERNET.CONNECT.XCHA SERNET.CONNECT.CMNB

DB2

SERNET keyword option DB2=subs is obsolete, and code that allows this keyword option will be removed in a future SERNET release. This keyword option has no relationship to the ChangeMan ZMF DB2 Option, and it is not used by ChangeMan ZDD.

CAUTION! If you use SERNET keyword option DB2=subs, legacy code for a discontinued Serena product establishes a DB2 thread that cannot be detached except by shutting down the SERNET server.

DDNAME

Purpose	Directs SERNET to get additional keyword parameter data from a data set coded in the started task JCL, and specifies the DD name where that data set is found.
Application(s)	All applications running under the SERNET started task
Format	DDNAME=ddname
Valid Values	ddname A sequential data set or a PDS member containing SERNET keyword parameters
Default Value	If this parameter is omitted, SERNET reads keyword parameters only through the PARM statement.
When Required	When there are more than 100 bytes of keyword parameter data (an IBM limit)
Comments	The data set can be fixed or variable, blocked or unblocked, but the record length cannot exceed 255 bytes.
Examples	DDNAME=PARMLIB PARM input from //PARMLIB DD data set

ESTAE

Purpose	Enables or disables abend recovery. Turns off the ESTAE exit for attached users.
Application(s)	All applications running under the SERNET started task
Format	ESTAE=NO
Valid Values	YES Enable recovery routines. Started task recovers from abends and continues processing. NO Disable recovery routines. Started task stops and dumps when an abend occurs.
Default Value	Omitting this parameter is the same as ESTAE=YES
When Required	ESTAE=NO is used primarily for testing.
Comments	**Do not use ESTAE=NO with any production systems.** Set to NO only when instructed to by Serena Technical Support.

EXPIRE

Purpose	Sets a future system clock time when the SERNET started task automatically shuts down.
Application(s)	All applications running under the SERNET started task
Format	EXPIRE=HhhMmm hh hours are preceded by "H" and mm minutes are preceded by "M" Hours are specified before minutes
Valid Values	hh 0 to 23; values larger than 23 are reset to 23 mm 0 to 59; values larger than 59 are reset to 59
Default Value	Omitting this parameter allows SERNET to run until it is shut down externally
When Required	Not required

Comments	The shutdown time may be changed to a later time with the EXTEND modify command.
Examples	EXPIRE=H23M59 Shutdown at 1 minute before midnight EXPIRE=H12 Shutdown at noon (12:00 high) EXPIRE=H20M15 Shutdown at 8:15 PM

EX003

Purpose	Disables SERJES calls to SEREX003 and reverts to whatever security mechanisms are supplied by the installed security system. As delivered, SEREX003 is enabled.
Application(s)	ChangeMan ZDD
Format	EX003=NO
Valid Values	NO Disable SEREX003
Default Value	Omitting this parameter leaves exit program SEREX003 enabled
Comments	SEREX003 is a SERJES exit that performs security checking when RACF® JESJOBS or JESSPOOL resource classes may not be active.

EX005

Purpose	Disables calls to exit program SEREX005, which provides library member level security.
Application(s)	ChangeMan ZMF
Format	EX005=NO
Valid Values	NO Disable SEREX005
Default Value	Omitting this parameter leaves exit program SEREX005 enabled
Comments	SEREX005 provide library member level security. This exit is called after the library access rules in your security system are applied. As delivered, SEREX005 is enabled.

IAM

Purpose	Determine if direct access data sets are IAM® instead of VSAM.
Application(s)	All applications running under the SERNET started task
Format	IAM=YES
Valid Values	YES Open direct access (BDAM) data sets determined to be IAM as IAM
Default Value	Omitting this parameter causes direct access (BDAM) data sets to be listed as "OTHER" and opened as VSAM
When Required	
Comments	Innovation Access Method (IAM) is a product of Innovation Data Processing. IAM is a substitute access method for VSAM that reduces DASD and CPU utilization and enhances performance for direct access files.

IEX

Purpose	Specifies the exit program that enforces standards for CA Librarian® -DESC and -PGMR cards when a module is added.
Application(s)	ChangeMan ZDD
Format	IEX=exitname
Valid Values	exitname Name of exit program
Default Value	Omitting this parameter is the same as having no exit program
When Required	CA Librarian environment where usage of -DESC and -PGMR cards is enforced through an exit program
Comments	
Examples	IEX=ADDCHK Input exit name is ADDCHK IEX=SHOPLIBR Input exit name is SHOPLIBR

IMS

Purpose	Names the IMS™ subsystem that is attached for an intermittent dialog. May also specify Normal Buffer Allocation (NBA) and Overflow Buffer Allocation (OBA) for a Fast Path region.
Application(s)	Serena Connect
Format	IMS=subs IMS=(subs,NBA=nn,OBA=oo) Order of NBA and OBA is not significant.
Valid Values	subs IMS subsystem nn Fast Path Normal Buffer Allocation. Any numeric value expressed in 1 to 8 characters. oo Fast Path Overflow Buffer Allocation. Any numeric value expressed in 1 to 8 characters.
Default Value	Omitting this parameter means no IMS subsystem is accessed. Omitting NBA and OBA or entering NBA=0,OBA=0 leaves in effect the values specified in the IMS sysgen.
When Required	Specify the IMS subsystem if the SERNET IMS Option has been licensed. (The SERNET IMS Option is not the same as the ChangeMan ZMF IMS Option.) Specify NBA and OBA if the values specified in the IMS sysgen are insufficient.
Comments	In practice, IMS subsystems may be accessed other than the one specified in the IMS keyword option. IMS databases can be downloaded through existing PSBs (Program Specification Blocks). The NBA and OBA subparameters are almost never used. They operate independently.
Examples	IMS=IMS2 IMS subsystem is IMS2 IMS=(IMS2,NBA=9,OBA=5) Optional number of database buffers for normal and overflow

LANGUAGE

Purpose	Specifies the language code for SERNET message template module (SERMXxxx).
Application(s)	SERNET
Format	LANGUAGE(ENU)
Valid Values	ENU
Default Value	ENU

LCLCCSID

Purpose	Specifies the Coded Character Set Identifier (CCSID) of text data that is stored in the ChangeMan ZMF server for clients that send and receive UNICODE request and response messages.
Application(s)	ChangeMan ZMF
Format	LCLCCSID(37) LCLCCSID=37
Valid Values	Any valid CCSID. The default is LCLCCSID(37) which is US EBCDIC and should suffice for most customers. Japanese customers who use DBCS CCSID 00939 should specify LCLCCSID=00939.
Default Value	LCLCCSID(37)
When Required	When the ChangeMan ZMF server has been localized for a CCSID other than US EBCDIC
Comments	In ChangeMan ZMF, Double Byte Character Set (DBCS) data can be in three places: ■ Package title ■ Programmer name ■ Inside a component When clients such as TeamTrack send XML requests to the MVS server, they send UNICODE XML. The server on MVS converts the incoming XML to its local CCSID as specified in the LCLCCSID startup parameter. When sending XML responses to the clients that sent UNICODE, the server converts the XML from its local CCSID to UNICODE. This process preserves any DBCS characters that may be contained in the XML.
Examples	LCLCCSID=37 LCLCCSID(37) LCLCCSID=000939 LCLCCSID(000939)

LIB

Purpose	Specifies the CA Librarian update module name.
Application(s)	ChangeMan ZMF, ChangeMan ZDD
Format	LIB=librname
Valid Values	libname Valid load module name or alias

Default Value	Omitting this parameter leaves the CA Librarian update module name as LIBR
When Required	In a CA Librarian environment where the CA Librarian update module has been renamed to something other than LIBR
Comments	If ZMF cannot find the CA Librarian update module, the result is user message "File Type LIB not supported" and return code 16.
Examples	LIB=LIBRARY CA Librarian update module name is LIBRARY LIB=SHOPLIBR CA Librarian update module name is SHOPLIBR

MIGRAT

Purpose	Specifies a pseudo volume serial for disk data sets that are migrated to a secondary storage format or medium.
Application(s)	All applications running under the SERNET started task
Format	MIGRAT=volser
Valid Values	volser One- to eight-character pseudo volume serial specified by the disk storage management system
Default Value	Omitting this parameter leaves the pseudo volume serial set to MIGRAT
When Required	When a disk storage management product uses a volume serial other than MIGRAT for migrated data sets
Comments	IBM product DFSMShsm™ uses MIGRAT as the volume serial for migrated data sets.
Examples	MIGRAT=ARCIVE Pseudo volume serial for DMS/OS MIGRAT=FDRABR Pseudo volume serial for FDR® ABR®

NOSCHED

Purpose	Suppresses the ChangeMan ZMF internal scheduler.
Application(s)	ChangeMan ZMF
Format	NOSCHED
Valid Values	NOSCHED The ChangeMan ZMF internal scheduler is suppressed
Default Value	Omitting this parameter leaves the ChangeMan ZMF scheduler in force
When Required	Never
Examples	NOSCHED

NOTIFYPROC

Purpose	Specifies the name of the Job Notification cataloged procedure to be inserted as the last step in batch jobs that are submitted from ZDD. The Job Notification facility sends a job completion message from the host back to the work station that submitted the job.
Application(s)	ChangeMan ZDD

Format	NOTIFYPROC=PROCNAME
Default Value	Omitting this parameter is the same as coding NOTIFYPROC=$SERNTFY

NTFYINT

Purpose	Specifies the time interval in seconds between cycles of the of the ChangeMan ZMF event notification facility.
Application(s)	Application Lifecycle Framework (ALF)
Format	NTFYINT=n n Seconds in 1-10 decimal digits
Valid Values	1 to 4294967295
Default Value	Omitting this parameter is the same as coding NTFYINT=60
When Required	Never
Comments	With each cycle, the event notification facility scans the ZMF Log File for applicable ZMF events, includes and excludes events according to filters defined in the file at DDname ALFFLTR in the ZMF started procedure (if present), and emits event notifications to the Web Services URL specified in SERNET keyword option NTFYURL. The Log File scan starts with the record after the last record read in the prior event notification cycle. The time between cycles of the event notification facility is defined in SERNET keyword option NTFYINT. For details of how to set up ALFFLTR, see the ALFFLTR member supplied in the SAMPLES dataset.

NTFYURL

Purpose	The URL where Web Services are installed to receive messages from the ChangeMan ZMF event notification facility.
Application(s)	Application Lifecycle Framework (ALF)
Format	NTFYURL='hostname:port/almzmfalf/services/ZMFALFEventRouter' hostname Server name where Serena ALM Web Services are installed port Port number for ALM Web Services **Note:** Use single or double quotes to prevent forcing the URL to upper case.
Default Value	Omitting this parameter disables the ZMF event notification facility.
When Required	This keyword parameter is required to run the event notification facility.
Comments	With each cycle, the event notification facility scans the ZMF Log File for applicable ZMF events, includes and excludes events according to filters defined in the file at DDname ALFFLTR in the ZMF started procedure (if present), and emits event notifications to the Web Services URL specified in SERNET keyword option NTFYURL. The Log File scan starts with the record after the last record read in the prior event notification cycle. The time between cycles of the event notification facility is defined in SERNET keyword option NTFYINT. For details of how to set up ALFFLTR, see the ALFFLTR member supplied in the SAMPLES dataset.
Examples	NTFYURL='alm_host:8080/almzmfalf/services/ZMFALFEventRouter' NTFYURL="alm_host:8080/almzmfalf/services/ZMFALFEventRouter"

RUNFOR

Purpose	Sets a time interval after startup when the SERNET started task is automatically shut down.
Application(s)	All applications running under the SERNET started task
Format	RUNFOR=HhhMmm hh hours are preceded by "H" and mm minutes are preceded by "M". Hours are specified before minutes.
Valid Values	hh 0 to 23; values larger than 23 are reset to 23. mm 0 to 59; values larger than 59 are reset to 59.
Default Value	Omitting this parameter allows SERNET to run until it is shut down externally
When Required	Never
Comments	Messages issued: SER0811I Automatic termination (expiration) set for 10:55 SER0940I Warning; EXPIRE/RUNFOR time nearing: 10:55 SER0941I EXPIRE/RUNFOR time reached; Shutting down The shutdown time may be changed to a later time with the EXTEND modify command.
Examples	RUNFOR=H23M59 Shut down SERNET after 23 hours and 59 minutes RUNFOR=M999 Shut down SERNET after 59 minutes RUNFOR=H1 Shut down SERNET after one hour

SDNOTIFY

Purpose	Deprecated. Use the **TIMEOUT** keyword option instead.

STAX

Purpose	Controls the behavior of the **ATTN** and **PA1** keys in applications running under SERNET.
Application(s)	All applications.
Format	STAX=NO
Valid Values	NO Make SERNET compatible with ISPF session managers that use the ATTN and PA1 keys to switch sessions. Users are not disconnected from ISPF applications running under SERNET when they press the **ATTN** or **PA1** keys.
Default Value	Omitting this parameter is the same as STAX=YES, which disconnects users from applications running under SERNET when they press the PA1 or ATTN keys.
When Required	Never
Examples	STAX=NO Do not disconnect users from ISPF applications running under SERNET when they press **ATTN** or **PA1**.

SUBSYS

Purpose	Specifies the subsystem ID for the SERNET started task.
Application(s)	All applications running under the SERNET started task.
Format	SUBSYS=x
Valid Values	One character: 0-9, A-Z, @, #, $, and blank (null) **Note:** Serena strongly recommends against using a blank (null) subsystem ID.
Default Value	Omitting this parameter sets the subsystem ID to blank (null)
When Required	Serena strongly recommends against using a blank (null) subsystem ID.
Examples	SUBSYS=A Subsystem identifier is A SUBSYS=5 Subsystem identifier is 5

TCPIP

Purpose	Identifies the TCP/IP server that handles communication for the SERNET started task.
Application(s)	ChangeMan ZMF, ChangeMan ZDD
Format	TCPIP=tcpiproc TCPIP.smfi=tcpiproc
Valid Values	tcpiproc Procedure name. smfi SMFID of another z/OS system.
Default Value	For TCP/IP 3.1, no entry is the same as `TCPIP=TCPIPROC`. For TCP/IP 3.2 and higher, SERNET 5.2 and higher ignores this keyword. Program SERCOMM dynamically deduces the name of the TCP/IP started task.
When Required	TCP/IP 3.1.
Comments	The IBM default name for TCPIP address spaces is TCPIP.
Examples	`TCPIP=TCP` TCP/IP procedure name is TCP. TCPIP.SMFA=JOBTCPIP TCP/IP procedure name is JOBTCPIP on system whose SMFID is SMFA.

TIMEOUT

Purpose	Specifies a period of inactivity after which a user is automatically detached by a "watchdog" facility. TIMEOUT can also be specified as SDNOTIFY.
Application(s)	All applications running under the SERNET started task
Format	TIMEOUT=nnn TIMEOUT=(nnn,nnn)
Valid Values	nnn Number of inactive minutes before user is detached. Minutes may be preceded by "M". Value is from 1 to 32767. If only one value nnn is specified, it applies to all users and clients. If two values are specified (nnn,nnn), the first value applies to TSO users and the second value applies to other clients such as ZDD or TeamTrack.

Default Value	Omitting the keyword lets users run no matter how long their session is inactive
When Required	Use of this parameter to enable an automatic detach is recommended.
Comments	A ChangeMan ZMF edit-in-stage session using ISPF edit turns off the timer for TIMEOUT. The inactivity limit set with z/OS parameter JWT(0100) in SYS1.PARMLIB(SMFPRM00) takes precedence over TIMEOUT and will end your TSO session with ABEND=S522 even if you are in a ZMF edit-in-stage session. However, if your ZMF administrator sets application parameter EDIT STAGING RECOVERY MODE ON to YES, you can recover the temporary file containing your edited text.
Examples	TIMEOUT=50 Detach users who are inactive for 50 minutes TIMEOUT=M090 Detach users who are inactive for 90 minutes

TRACE

Purpose	Sets initial trace options at SERNET startup.
Application(s)	SERNET, ChangeMan ZMF
Format	TRACE(SER|CMN,n,n,n...)
Valid Values	SER SERNET CMN ChangeMan ZMF n Trace class, an integer 1-32: 1 All CMN traces not listed below 2 CMNATACH user conversation incoming request and outgoing response. I.E. "TRACE IN …" and "TRACE OUT …". CMNENTFY event notification. CMNSCHED CMN scheduler 3 VSAM CMAST data interim results. This shows long and short CMAST data in a common format 4 CMNVRLIO VSAM i/o response (major/minor/key/rc/fdbk)
Default Value	Omitting the keyword leaves no tracing enabled
Examples	TRACE(CMN,1,2) TRACE on for ChangeMan ZMF, classes 1 and 2 TRACE(SER,1,2) TRACE on for SERNET, classes 1 and 2

WARN

Purpose	Controls XML Services syntax warnings facility.
Application(s)	XML Services
Format	WARN[(YES)] WARN=[YES|NO] WARN
Default Values	Omitting the keyword leaves XML Services syntax warning disabled at SERNET startup.
When Required	Never

Comments	When SERXMLI is unable to recognize an XML tag, it can write a warning message to SERPRINT. There are three ways to enable or disable these warning messages: ■ SERNET keyword option input to the started task at startup: WARN=[YES|NO] ■ SERNET modify command: /F jobname,WARN,[YES|NO] See "WARN" on page 179. ■ XML tag in the <header> of the request message: <warn>Y</warn> See the *Serena XML Services User's Guide.*
Examples	WARN Enable XML Services syntax warning WARN(YES) Enable XML Services syntax warning WARN=YES Enable XML Services syntax warning

XML

Purpose	Disables Serena XML Services.
Application(s)	ChangeMan ZMF
Format	XML=NO
Valid Values	NO Disables Serena XML Services.
Default Value	Omitting this option leaves Serena XML Services enabled.
When Required	Never

Appendix E

Modify Commands

This appendix describes modify commands that may be issued to a SERNET started task.

Introduction	168
Standard Modify Commands	168
Restricted Modify Commands	181

Introduction

These modify commands may be issued from the operator console or through SDSF by authorized users. Responses to these commands are displayed in the SERPRINT sysout data set of the SERNET instance.

Standard Modify Commands

An application administrator or a console operator may use these modify commands to manage a SERNET started task and the Serena applications that run under it.

ABEND

Purpose	Abend the started task with an 0C3.
Application(s)	All applications running under a SERNET started task.
Format	/F jobname,ABEND Where: jobname Started task jobname
Comments	Use of this modify command is not recommended without instructions from Technical Support.
Examples	`/F SERT5,ABEND`
Response	`ASER0850I Operator command: ABEND` `SER0952W Intentional abend <S0C3> requested`

ASID

Purpose	Modify the maximum number of concurrent address spaces and time-out minutes.
Application(s)	All applications running under a SERNET started task
Format	/F jobname,ASID[=(n,m)] Where: jobname Started task jobname n Maximum number of concurrent address spaces m Time-out in wall clock minutes

Comments	ASID values are set by: ■ Default at startup (16 address spaces and 20 minutes time-out expiration) ■ SERNET keyword option ASID at startup ■ ASID modify command The maximum number of address spaces is checked when an application needs to start an address space for an asynchronous operation. An example of an asynchronous process that uses a separate address space is ChangeMan ZMF ISPF file tailoring for install JCL. SERNET periodically checks the list of address spaces that it has started, and if an address space runs longer than the time-out limit, it is terminated. Such an address space may be in a loop or have some other problem, and it can prevent another address space from starting when the maximum number of address spaces is reached. The ASID time out limit is similar to the TIME parameter for an ordinary job except it is wall clock time, not CPU time. The elapsed run time for an address space is calculated as the difference between the store clock (STCK) value taken when the address space was created and the store clock value taken when the check is made. The maximum number of address spaces is 256, and the maximum number of time-out minutes is 255, but whatever you input is displayed in SERPRINT. If you enter a number of minutes greater than 255, a time-out limit of 255 is used, which is 4 hours 15 minutes.
Examples	`/F SERT5,ASID=(64,30)`

Response	`SER0850I Operator command: ASID=(64,30)`
	`SER0901I Number of Address Spaces:64 Expiration Timeout:30 Minutes`
	`SER0904I 0 active address spaces are running at this time.`

ASIDS

Purpose	Display the maximum number of concurrent address spaces and time-out minutes. Display the number of address spaces that are running as well as the address space requests that are waiting to run and the classes assigned.
Application(s)	All applications running under a SERNET started task
Format	/F jobname,ASIDS Where: jobname Started task jobname
Comments	See modify command ASID for an explanation of the maximum number of concurrent address spaces and expiration time-out minutes. The maximum number of address spaces is 256, and the maximum number of time-out minutes is 255, but whatever was input is displayed in SERPRINT.
Examples	`/F SERT5,ASIDS`

Response	`SER0850I Operator command: ASIDS`
	`SER0901I Number of Address Spaces:16 Expiration Timeout:20 Minutes`
	`SER0904I 0 active address spaces are running at this time.`

ASINITi

Purpose	For an initiator, modify the maximum number of concurrent address spaces that can be created and the job classes assigned to the initiator.
Application(s)	All applications running under a SERNET started task

Format	/F jobname, ASINITi=(n,c) Where: jobname Started task jobname i Initiator number n Maximum number of concurrent address spaces c Job classes
Comments	The address space limit is checked when an application needs to start an address space for an asynchronous operation. An example of an asynchronous process that uses a separate address space is ChangeMan ZMF ISPF file tailoring for install JCL. The maximum number of address spaces is 256, but whatever you input is displayed in SERPRINT.
Examples	`/F SERT5,ASINIT2=(50,ABC)`

Response	`SER0850I Operator command: ASINIT3=(10,ABC)`
	`SER0901I Number of Address Spaces:22 Expiration Timeout:25 Minutes`
	`SER0902I Initiator:1 Classes:ABCD Maximum:4 Active:0`
	`SER0902I Initiator:2 Classes:ABCD Maximum:4 Active:0`
	`SER0902I Initiator:3 Classes:ABC Maximum:10 Active:0`
	`SER0902I Initiator:4 Classes:ABCD Maximum:4 Active:0`

ASINITS

Purpose	For each initiator, display the assigned job classes, the maximum number of concurrent address spaces that can be created, and the number of active address spaces.
Application(s)	All applications running under a SERNET started task
Format	/F SERs,ASINITS Where: jobname Started task jobname
Comments	This command also display the address space requests that are waiting to run and the classes assigned.
Examples	`/F SERT5,ASINITS`

Response	`SER0850I Operator command: ASINITS`
	`SER0901I Number of Address Spaces:32 Expiration Timeout:25 Minutes`
	`SER0902I Initiator:1 Classes:ABCD Maximum:4 Active:0`
	`SER0902I Initiator:2 Classes:ABCD Maximum:4 Active:0`
	`SER0902I Initiator:3 Classes:ABCD Maximum:4 Active:0`
	`SER0902I Initiator:4 Classes:ABCD Maximum:4 Active:0`

ATTACH

Purpose	Attach a subtask to an application running under the SERNET started task.
Application(s)	ChangeMan ZMF
Format	/F jobname,aaa,ATTACH,tttttttt Where: jobname STC jobname aaa Application tttttttt subtask CMN ChangeMan ZMF HLLX High Level Language Exits SSV Staging Versions IADS Impact Analysis NFT Event Notification

Comments	See the DETACH modify command. For SSV, the three staging version VSAM masters are attached to the started task. For IADS, the impact analysis data space is created and populated from the VSAM LDS at DD statement CMNIMPCT, and DD statement CMNIALOG is allocated and opened.
Examples	`/F SERT5,CMN,ATTACH,SSV` Attach the Staging Versions subtask `/F SERT5,CMN,ATTACH,IADS` Attach the Impact Analysis subtask

Responses	`SER0850I Operator command: CMN,ATTACH,SSV`
	`CMN5300I CMNSTART Attach:`
	`U=CMNSTART,F1=ATTACH,F2=SSV,@TCA=17139000,@TCB=8C3CF0`
	`CMN5315I CMNSTART SSV Task Attached`
	`CMN5385I CMNSSV Attempting SSV initialization CMNSSV (MVS-6.1.1) 20090506 10.14`
	`CMN5388I CMNSSV SSV initialization complete`
	
	`SER0850I Operator command: CMN,ATTACH,IADS`
	`CMN5300I CMNSTART Attach:`
	`U=CMNSTART,F1=ATTACH,F2=IADS,@TCA=17115000,@TCB=8C329`
	`CMN5315I CMNSTART IADS Task Attached`
	`CMN5391I CMNIAM00 Attempting to open the I/A Dataspace.`
	`CMN5392I CMNIAM00 I/A Dataspace OPEN, LDS: CMNTP.SERT5.CMNZMF.V6R1M1.IADSP`

DETACH

Purpose	Detach a subtask from an application running under the SERNET started task.
Application(s)	ChangeMan ZMF
Format	/F jobname,aaa,DETACH,tttttttt Where: jobname — STC jobname aaa — Application subtask tttttttt — CMN ChangeMan ZMF; HLLX High Level Language Exits; SSV Staging Versions; IADS Impact Analysis; NTF Event Notification
Comments	See the ATTACH modify command. For SSV, the three staging version VSAM masters are closed and deallocated from the started task. For IADS, the impact analysis LDS is synchronized with the data space, the data space is closed, LDS is closed and deallocated from the started task, and DD statement CMNIALOG is closed and deallocated. **CAUTION!** When impact analysis is detached, these functions are affected: ■ Package audit fails with RC=20 ■ Baseline ripple and reverse baseline ripple jobs yield RC=04 in step DSPTM, and impact analysis updates are written to the CMNDELAY file. Transactions in the CMNDELAY file are not processed until the SERNET/ZMF started task is restarted. ■ Query Impact and Query BOM return no results.

Examples	`/F SERT5,CMN,DETACH,SSV`	Detach the Staging Versions subtask
	`/F SERT5,CMN,DETACH,IADS`	Detach the Impact Analysis subtask

Responses	`SER0850I Operator command: CMN,DETACH,SSV`
	`CMN5389I CMNSSV Terminating SSV`
	`CMN5387I CMNSSV SSV terminated`
	`CMN5305I CMNSTART Detach:`
	`U=CMNSTART,F1=DETACH,F2=SSV,@TCA=17107000,@TCB=8C3CF0`
	`CMN5316I CMNSTART SSV Task Detached`
	
	`SER0850I Operator command: CMN,DETACH,IADS`
	`CMN5394I CMNIAM00 Closing the I/A Dataspace.`
	`CMN5395I CMNIAM00 The I/A Dataspace is now offline.`
	`CMN5305I CMNSTART Detach:`
	`U=CMNSTART,F1=DETACH,F2=IADS,@TCA=170F5000,@TCB=8C42A08`
	`CMN5316I CMNSTART IADS Task Detached`	

EXTEND

Purpose	Extend the time when the SERNET started task automatically shuts down as specified in the EXPIRE or RUNFOR runtime parameters.
Application(s)	All applications running under a SERNET started task
Format	/F jobname,EXTEND,HhhMmm Where: jobname — Started task jobname hh — Hours to extend the started task expire time mm — Minutes to extend the started task expire time
Comments	The new system clock time for shutdown is displayed.
Examples	`/F SERT5,EXTEND,H00M15`
Response	`SER0850I Operator command: EXTEND,H00M15` `SER0870I Expiration time extended to 07:53`

HALT

Purpose	Stop the SERNET started task with an orderly termination process that closes VSAM files and disconnects from system resources. This command is the same as SHUTDOWN and is provided to comply with NetView terminology.
Application(s)	SERNET and applications running under the SERNET started task
Format	/F jobname,HALT,[n|Mn] Where: jobname — Started task jobname n — Grace period in minutes before attached users are forcibly detached and termination of ChangeMan ZMF and SERNET is initiated. If a grace period is not specified, the grace period defaults to 5 minutes. If a grace period of 0 is specified, the shutdown is immediate. The grace period minutes subparameter may be preceded by M.

Comments	No users are allowed to logon after a HALT command is issued. During the grace period, SERNET checks every 60 seconds for attached users. If all users have logged off, termination is initiated. If attached users are found, this message is sent to each TSO ID: SER2000I CMN Serena Network shutting down; Please exit! CN(INTERNAL) After termination is initiated, the process may take several minutes to complete.
Examples	/F SERT5,HALT,1 /F SERT5,HALT /F SERT5,HALT,0 /F SERT5,HALT,1M
Response	SER0850I Operator command: HALT,0 SER0944I SerNet orderly SHUTDOWN initiated; No grace period; Immediate terminat CMN5389I CMNSSV Terminating SSV CMN5394I CMNIAM00 Closing the I/A Dataspace. CMN5395I CMNIAM00 The I/A Dataspace is now offline. CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNPMAST CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNCMPNT CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNLOG CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNRECV CMN5810I CMNSTART Change Man VSAM Files closed CMN5820I CMNSTART Change Man VSAM File Statistics CMN5820I CMNSTART CMNPMAST CMNCMPNT CMNLOG CMNRECV CMNDELAY C CMN5820I CMNSTART Read Update 00000005 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Read Only 00000003 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Read Next 00000447 00000000 00000000 00000000 00000001 0 CMN5820I CMNSTART Write (Sync) 00000000 00000000 00000002 00000004 00000000 0 CMN5820I CMNSTART Write (ASync) 00000000 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Update 00000004 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Delete 00000000 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Start 00000000 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Start Generic 00000149 00000000 00000000 00000000 00000001 0 CMN5820I CMNSTART End Request 00000002 00000001 00000000 00000000 00000000 0 CMN5820I CMNSTART Dup Records 00000000 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Record locks 00000005 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART Max locks used 00000001 00000000 00000000 00000000 00000000 0 CMN5820I CMNSTART CI locks found 00000000 00000000 00000000 00000000 00000000 0 CMN5387I CMNSSV SSV terminated CMN5305I CMNSTART Detach: U=CMNSTART,F1=DETACH,F2=IADS,@TCA=170F5000,@TCB=8C42A SER0989I Waiting for application to terminate: CMN CMN5305I CMNSTART Detach: U=CMNSTART,F1=DETACH,F2=INFO,@TCA=170FF000,@TCB=8C1E8 CMN5305I CMNSTART Detach: U=CMNSTART,F1=DETACH,F2=SSV,@TCA=17108000,@TCB=8BF088 CMN5313I CMNSTART Termination Complete SER0990I Statistics; Real time: H02M30S53 SER0991I Statistics; Applications: 4 SER0992I Statistics; Users attached: 0 SER0993I Statistics; Called count: 0 SER0994I Statistics; Concurrent max: 0 SER0995I Statistics; Abended: 0 SER0996I Statistics; Detached inactive: 0 SER0999I SerNet server "5" termination complete, RC=0

HOLD

Purpose	Stop the ChangeMan ZMF internal scheduler from submitting installation jobs for packages that specify CMN for the Installation Job Scheduler.
Application(s)	ChangeMan ZMF
Format	/F jobname,CMN,HOLD Where: jobname Started task jobname
Comments	Install jobs are not held for packages that use MANUAL or OTHER for the Installation Job Scheduler. Issue the HOLD modify command in the P environment started task to hold packages that have been distributed from a D or DP environment.

Examples	`/F SERT5,CMN,HOLD`
Response	`SER0850I Operator command: CMN,HOLD` `CMN5317I CMNSTART All scheduled jobs held`

LOCKS

Purpose	Display information about serialization locks in force.
Application(s)	ChangeMan ZDD
Format	/F jobname,XCH,LOCK Where: jobname Started task jobname
Comments	Data displayed: <User_ID> Logon ID for a given user <T_Elapsed> Number of hours, minutes, and seconds that have elapsed since the lock was placed. If the hours exceed 24, days are displayed and seconds are dropped. <Type> Type of lock applied. If the literal 'ENQ' is displayed, an ENQ is outstanding under Qname SPFEDIT for the displayed data set name and likely member name in parenthesis. <Locked_DsName (Member)...> Data set name and member name on which a temporary serialization lock was placed. Total= Total number of locks that have been placed since the SERNET instance was started
Examples	`/F SERT5,XCH,LOCK` `/F SERT5,XCH,LOCKS`

Response	`SER0850I Operator command: XCH,LOCK`
	`SER2013I XCH <User_ID><T_Elapsed><Type><Locked_DsName(Member)...> Total=0`
	`SER2016I XCH No active locks found`

RELEASE

Purpose	Release installation jobs that are being held by the HOLD modify command.
Application(s)	ChangeMan ZMF
Format	/F jobname,CMN,RELEASE Where: jobname Started task jobname
Comments	See the HOLD modify command.
Examples	`/F SERT5,CMN,RELEASE`
Response	`SER0850I Operator command: CMN,RELEASE` `CMN5320I CMNSTART All scheduled jobs released`

SHUTDOWN

Purpose	Stop the SERNET started task with an orderly termination process that closes VSAM files and disconnects from system resources. This command is the same as HALT, which is provided to comply with NetView terminology.
Application(s)	SERNET and applications running under the SERNET started task

Format	/F jobname,SHUTDOWN,[n|Mn] Where: jobname — Started task jobname n — Grace period in minutes before attached users are forcibly detached and termination of ChangeMan ZMF and SERNET is initiated. If a grace period is not specified, the grace period defaults to 5 minutes. If a grace period of 0 is specified, the shutdown is immediate. The grace period minutes subparameter may be preceded by M. SHUTDOWN can be coded as SH, SHUT, or HALT.
Comments	No users are allowed to logon after a SHUTDOWN command is issued. During the grace period, SERNET checks every 60 seconds for attached users. If all users have logged off, termination is initiated. If attached users are found, this message is sent to each TSO ID: SER2000I CMN Serena Network shutting down; Please exit! CN(INTERNAL) After termination is initiated, the process may take several minutes to complete.

Examples	`/F SERT5,SHUTDOWN,1`
	`/F SERT5,SH`
	`/F SERT5,SH,0`
	`/F SERT5,SHUTDOWN,1M`

Response	`SER0850I Operator command: SH,0`
	`SER0944I SerNet orderly SHUTDOWN initiated; No grace period; Immediate terminat`
	`CMN5389I CMNSSV Terminating SSV`
	`CMN5394I CMNIAM00 Closing the I/A Dataspace.`
	`CMN5395I CMNIAM00 The I/A Dataspace is now offline.`
	`CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNPMAST`
	`CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNCMPNT`
	`CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNLOG`
	`CMN5800I CMNSTART Closed VSAM file CMNTP.SERT5.CMNZMF.V6R1M1.CMNRECV`
	`CMN5810I CMNSTART Change Man VSAM Files closed`
	`CMN5820I CMNSTART Change Man VSAM File Statistics`
	`CMN5820I CMNSTART CMNPMAST CMNCMPNT CMNLOG CMNRECV CMNDELAY C`
	`CMN5820I CMNSTART Read Update 00000005 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Read Only 00000003 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Read Next 00000447 00000000 00000000 00000000 00000001 0`
	`CMN5820I CMNSTART Write (Sync) 00000000 00000000 00000002 00000004 00000000 0`
	`CMN5820I CMNSTART Write (ASync) 00000000 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Update 00000004 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Delete 00000000 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Start 00000000 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Start Generic 00000149 00000000 00000000 00000000 00000001 0`
	`CMN5820I CMNSTART End Request 00000002 00000001 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Dup Records 00000000 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Record locks 00000005 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART Max locks used 00000001 00000000 00000000 00000000 00000000 0`
	`CMN5820I CMNSTART CI locks found 00000000 00000000 00000000 00000000 00000000 0`
	`CMN5387I CMNSSV SSV terminated`
	`CMN5305I CMNSTART Detach:`
	`U=CMNSTART,F1=DETACH,F2=IADS,@TCA=170F5000,@TCB=8C42A`
	`SER0989I Waiting for application to terminate: CMN`
	`CMN5305I CMNSTART Detach:`
	`U=CMNSTART,F1=DETACH,F2=INFO,@TCA=170FF000,@TCB=8C1E8`
	`CMN5305I CMNSTART Detach:`
	`U=CMNSTART,F1=DETACH,F2=SSV,@TCA=17108000,@TCB=8BF088`
	`CMN5313I CMNSTART Termination Complete`
	`SER0990I Statistics; Real time: H02M30S53`
	`SER0991I Statistics; Applications: 4`
	`SER0992I Statistics; Users attached: 0`
	`SER0993I Statistics; Called count: 0`
	`SER0994I Statistics; Concurrent max: 0`
	`SER0995I Statistics; Abended: 0`
	`SER0996I Statistics; Detached inactive: 0`
	`SER0999I SerNet server "5" termination complete, RC=0`

SNAPENV

Purpose	Display information on started task region memory usage.
Application(s)	ChangeMan ZMF
Format	/F jobname,SNAPENV Where: jobname Started task jobname
Comments	
Examples	`/F SERT5,SNAPENV`
Response	`SER0850I Operator command: SNAPENV` `SER0868I EPvt used=43,036K avail=1,701,860K Pvt used= 676K avail=8,516K`

STATS

Purpose	Display VSAM file access statistics for the ChangeMan ZMF package master, component master, log, recovery, and delay files.
Application(s)	ChangeMan ZMF
Format	/F jobname,CMN,STATS Where: jobname Started task jobname
Comments	If you license the ChangeMan ZMF INFO Option, statistics for the INFO VSAM interface file are also displayed.
Examples	`/F SERT5,CMN,STATS`

Response	`SER0850I Operator command: CMN,STATS`
	`CMN5820I CMNSTART Change Man VSAM File Statistics`
	`CMN5820I CMNSTART CMNPMAST CMNCMPNT CMNLOG CMNRECV CMNDELAY`
	`CMN5820I CMNSTART Read Update 00000005 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Read Only 00000003 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Read Next 00000438 00000000 00000000 00000000 00000001`
	`CMN5820I CMNSTART Write (Sync) 00000000 00000000 00000001 00000004 00000000`
	`CMN5820I CMNSTART Write (ASync) 00000000 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Update 00000004 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Delete 00000000 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Start 00000000 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Start Generic 00000146 00000000 00000000 00000000 00000001`
	`CMN5820I CMNSTART End Request 00000002 00000001 00000000 00000000 00000000`
	`CMN5820I CMNSTART Dup Records 00000000 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Record locks 00000005 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART Max locks used 00000001 00000000 00000000 00000000 00000000`
	`CMN5820I CMNSTART CI locks found 00000000 00000000 00000000 00000000 00000000`

TCPIP

Purpose	Manage TCP/IP for applications running under SERNET: ■ Gracefully detach inactive users connected to a SERNET started task through TCP/IP, and stop TCP/IP without disrupting users connected through cross memory services. ■ Start TCP/IP for use by SERNET. ■ Gracefully disconnect inactive users using TCP/IP, stop and start TCP/IP without disrupting users connected through cross memory services.
Application(s)	SERNET

Format	/F jobname,TCPIP,action,[mmm],[tcpipname] Where: jobname — started task jobname action — START, STOP, RESTART mmm — Minutes of inactivity after a STOP command before a TCP/IP user is automatically detached. If *mmm* not specified, default is 0. tcpipname — TCPIP procedure name for START and RESTART
Comments	After the TCPIP command is entered, subsequent actions such as detaching inactive users, stopping TCPIP, and restarting TCPIP are executed on the minute at hh:mm:00:00. STOP: When the STOP command is entered, no further TCP/IP connections are accepted, currently attached users are listed in SERPRINT, and the following MVSEND message is issued to users connected through TCP/IP: SER2060T TCP/IP connections being terminated due to operator request CN(INTERNAL) This message is reissued each minute until the TCPIP user disconnects or is automatically detached. A TCPIP user is automatically detached when they have been inactive for the specified *mmm* interval. After all TCPIP users disconnect or are automatically detached, the controlling task (SERMAIN) calls SERCOMM to close the socket and process a TERMAPI. START: This command starts TCP/IP communications (INITAPI) provided that the TCP/IP API has not already been initialized. If *tcpipname* is not specified in the TCPIP command, INITIAPI (SERCOMM) connects to the default STC stack TCPIP, or it leaves the selection up to the TCP/IP code that parses the SYSTCPD DD statement. If *tcpipname* is specified, INITIAPI (SERCOMM) specifies this STC stack name when creating the TCP/IP environment. RESTART: The RESTART command is a combination of the STOP and START commands. TCPIP is restarted one minute after the last inactive TCPIP user is detached. **Note:** Use discrete STOP and START commands when you want to recycle the TCP/IP STC. Use the RESTART command with a *tcpipname* parameter when you want to switch TCP/IP stacks.

Examples	/F SERT5,TCPIP,STOP /F SERT5,TCPIP,START /F SERT5,TCPIP,RESTART /F SERx,TCPIP,STOP,IMMED
Response	SER0850I Operator command: TCPIP,STOP SER0916I TCPIP Stop Requested SER1020I SYSTEM TCP/IP SETSOCKOPT: RC=FFFF ErrNo=1027 TCA=00010000 Task=SER TCB=008DAE88 TIE=15B62F50 Sock=0000 SER1020I SYSTEM TCP/IP CLOSE: RC=FFFF ErrNo=1027 TCA=00010000 Task=SER TCB=008DAE88 TIE=15B62F50 Sock=0000 SER0920I Current Users:0 Maximum Users:32767 SER2010I XCH ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0 SER2012I XCH No active users found SER2010I CMN ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0 SER2012I CMN No active users found SER1020I XCH TCP/IP CLOSE: RC=FFFF ErrNo=1027 TCA=15B6E000 Task=XCH00001 TCB=008CC888 TIE=15B7C150 Sock=0000 SER0850I Operator command: TCPIP,START SER0915I TCPIP Start Requested SER1018I SYSTEM The following TCP/IP procedures are active: TCPIP SER1020I SYSTEM TCP/IP INITAPI: RC=FFFF ErrNo=10197 TCA=00010000 Task=SER TCB=008DAE88 TIE=15B62F50 Sock=0000 SER1020I CMNSTART TCP/IP BIND: RC=FFFF ErrNo=48 TCA=15B98000 Task=CMN TCB=008CC0D8 TIE=15BC6090 Sock=0000 SER1014E CMNSTART *Error* Specified port number already in use: 6051 SER1000I XCH TCP/IP environment active at 10.35.11.100..6052 SER1001I XCH TCP/IP local host name: C001 SER0826I LIBR not found SER0850I Operator command: TCPIP,RESTART SER0917I TCPIP Restart Requested SER0920I Current Users:0 Maximum Users:32767 SER2010I XCH ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0 SER2012I XCH No active users found SER2010I CMN ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0 SER2012I CMN No active users found SER1000I XCH TCP/IP environment active at 10.35.11.100..6052 SER1001I XCH TCP/IP local host name: C001 SER0850I Operator command: TCPIP,STOP,IMED SER0916I TCPIP Stop Requested SER1020I SYSTEM TCP/IP SETSOCKOPT: RC=FFFF ErrNo=1027 TCA=00010000 Task=SER TCB=008DAE88 TIE=15B62F50 Sock=0000 SER1020I SYSTEM TCP/IP CLOSE: RC=FFFF ErrNo=1027 TCA=00010000 Task=SER TCB=008DAE88 TIE=15B62F50 Sock=0000 SER0920I Current Users:2 Maximum Users:32767 SER2010I XCH ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0 SER2012I XCH No active users found SER2010I CMN ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=2 SER2011I CMN 001 USER240 H00M01S05 H00M01S04 10.35.11.100 SER2011I CMN 002 USER239 H00M00S36 H00M00S33 10.30.224.18

USERS

Purpose	Display users connected to the SERNET started task. If the application is specified, only users connected to that application are displayed. If SUMMARY is specified, only a count of users connected to each application is displayed.
Application(s)	All applications running under a SERNET started task
Format	/F jobname[,aaa],USERS /F jobname,USERS[,SUMMARY] Where: jobname Started task jobname aaa Application running under SERNET

Comments	Data displayed: ###: Three-digit relative number for the user. User_ID: Logon ID for the user. T_Elapsed: Number of hours, minutes, and seconds that have elapsed since the initial logon. If hours exceed 24, days are displayed and seconds are dropped. T_LastAct: Number of hours, minutes, and seconds that have elapsed since the last contact with the SERNET instance. If hours exceed 24, days are displayed and seconds are dropped. Partner Identifier: Fully qualified partner logical unit name or TCP/IP address of the workstation. Nothing is displayed if the connection is through cross memory services. Current/Max=0/0: No longer maintained by SERNET.

Examples	`/F SERT4712,USERS`
	`/F SERT4712,CMN,USERS`
	`/F SERT4712,USERS,SUMMARY`

Response	`SER0850I Operator command: USERS`
	`SER0920I Current Users:0 Maximum Users:32767`
	`SER2010I XCH  ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0`
	`SER2012I XCH  No active users found`
	`SER2010I CMN  ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0`
	`SER2012I CMN  No active users found`
	
	`SER0850I Operator command: CMN,USERS`
	`SER2010I CMN  ###<User_ID><T_Elapsed><T_LastAct><Partner Identifier> Users=0`
	`SER2012I CMN  No active users found`
	
	`SER0850I Operator command: USERS,SUMMARY`
	`SER0920I Current Users:0 Maximum Users:32767`
	`SER2009I XCH  Users: 0`
	`SER2009I CMN  Users: 0`

WARN

Purpose	Controls XML Services syntax warnings facility.
Application(s)	XML Services
Format	/F jobname,WARN,[YES|NO] Where: jobname Started task jobname
Comments	When SERXMLI is unable to recognize an XML tag, it can write a warning message to SERPRINT. There are three ways to enable or disable these warning messages: ■ SERNET modify command: /F jobname,WARN,[YES|NO] ■ SERNET keyword option input to the started task at startup: WARN=[YES|NO] See "WARN" on page 164. ■ XML tag in the <header> of the request message: <warn>Y</warn> See the *Serena XML Services User's Guide.*

Examples	`/F SERT4712,WARN,YES`	Enable XML Services syntax warning
	`/F SERT4712,WARN,NO`	Enable XML Services syntax warning
	`/F SERT4712,WARN`	Display status of XML Services syntax warning

Response	`SER0850I Operator command: WARN,YES`
	`SER0960I XML syntax warning has been turned on`
	
	`SER0850I Operator command: WARN,NO`
	`SER0959I XML syntax warning has been turned off`
	
	`SER0850I Operator command: WARN`
	`SER0961I XML syntax warning is: Off`	

Restricted Modify Commands

These modify commands will add significant processing overhead and are to be used only as directed by Serena Technical Support. They are included here to provide format specifications and examples.

NETTRACE

Purpose	Display the contents of communications buffers to trace data that flows across the network. Dump the contents of each buffer into the SERPRINT sysout data set.
Application(s)	All applications running under a SERNET started task
Format	/F jobname,NETTRACE {ON|OFF|DISPLAY},[LUNAME|TCP|USER], TRACESIZE=size Where: jobname — Started task jobname ON — Enable NETTRACE as specified OFF — Disable NETTRACE as specified DISPLAY — Display all NETTRACE specifications currently enabled LUNAME=data — Trace this LU. Abbreviation: L=. Trailing * wildcard permitted. TCP=data — Trace this TCP/IP address. Abbreviation: T= for trace. Trailing * wildcard permitted in each address segment. USER=data — Trace this user ID. Abbreviation: U=. Trailing * wild card permitted. TRACESIZE=data — Specifies maximum amount of data to display for each call to the network tracing routines. Default is 1024 bytes, minimum is 128 bytes.
Comments	The TRACE modify command is not a toggle switch. You must turn OFF any NETTRACE that you turn ON. You cannot use a wildcard to turn OFF multiple TRACE,ON commands. NETTRACE can generate high volumes of output if all users are traced or there is a high level of activity for individual users that are traced.

Examples	`/F SERT5,NETTRACE,ON,USER=USER239`
	`/F SERT5,NETTRACE,ON,TCP=111.11.111.111`
	`/F SERT5,NETTRACE,DISPLAY`
	`/F SERT5,NETTRACE,OFF,U=USER239`
	`/F SERT5,NETTRACE,OFF,T=111.11.111.111`

Response	`SER0850I Operator command: NETTRACE,ON,USER=USER239` `SER0864I NETTRACE enabled: USER=USER239,SIZE=00256`

REFRESH

Purpose	Delete and reload specified SERNET load module.
Application(s)	SERNET
Format	/F jobname,REFRESH,modname Where: jobname — Started task jobname modname — Module to be refreshed

Comments	This modify command only refreshes load modules that are loaded by SERNET, which are typically named SERxxxxx. This modify command does not refresh load modules loaded by applications such as ZMF, whose load modules are typically named CMNxxxxx.
Examples	`/F SERT5,REFRESH,SEREX005`

Response	`SER0850I Operator command: REFRESH,SERCOPY`
	`SER1402I Name=SERCOPY old/new EPA=16250DF8/17148DF8 RC=0`
	`SER1403I Name=SERCOPY old/new token=1464C86E-00003208/E1C3C67E-00003208`
	`SER1405I Name=SERCOPY SERCOPY (MVS-7.1.2A) 20120725 13.21 Copy utility`
	`SER0850I Operator command: REFRESH,SERLCSEC`
	`SER1402I Name=SERLCSEC old/new EPA=00040470/00040470 RC=0`
	`SER1403I Name=SERLCSEC old/new token=BF73B8F5-80000A92/BF73B8F5-80000A92`
	`SER1405I Name=SERLCSEC SERLCSEC/SAF; (MVS-) 20090717 21.29`

TRACE

Purpose	Modify the TRACE options in effect.
Application(s)	SERNET, ChangeMan ZMF

Format	/F jobname,TRACE[,DISPLAY] [,ON,CMN|SER,CLASS=(n,n,n...)] [,OFF,CMN|SER,CLASS=(n,n,n,...)] [,ON,USER=uuuuuuu[*] [,OFF,USER=uuuuuuu[*] [,OFF,ID=n] ON is a synonym for YES OFF is a synonym for NO Where: jobname — Started task jobname DISPLAY — Display all TRACE facilities currently enabled ON or YES — Enable TRACE as specified OFF or NO — Disable TRACE as specified SER — SERNET CMN — ChangeMan ZMF n — Trace class, numeric 1-32 1 - All CMN traces not listed below 2 - CMNATACH user conversation incoming request and outgoing response. I.E. "TRACE IN …" and "TRACE OUT …". CMNENTFY event notification. CMNSCHED CMN scheduler. 3 - VSAM CMAST data interim results. This shows long and short CMAST data in a common format 4 - CMNVRLIO VSAM i/o response (major/minor/key/rc/fdbk) USER=uuuuuuu — Limit SERNET trace to this user ID (trailing wildcard permitted) ID=m — Numeric identifier for trace USER that is used to turn trace user limit off
Comments	If no user ID is specified, all user IDs are traced. The TRACE,ON,USER= command may be issued more than once to add user IDs to the list of eligible messages. TRACE options are logically ORed, meaning that if a message matches one of the specifications, the message is displayed. ChangeMan ZMF messages are never affected by a user ID limit and are always shown for all user IDs. All trace classes can also be enabled/disabled with the following format: /F stcname,TRACE,YES,CMN /F stcname,TRACE,YES,SER **Once the problem requiring the trace has been recreated, immediately disable all the traces by replacing any 'YES' with 'NO' and/or 'ON' with 'OFF' in each of the above modify commands.**

Trace Examples

In the following examples of the TRACE modify command:

- Each command that is entered on the operator console or in SDSF is followed by the response that is displayed in the SERPRINT data set.
- Trace specifications in these examples are cumulative. Trace DISPLAY commands show the cumulative trace options in effect.

- No TRACE keyword option was included in the SERNET startup parameters.

1 Display the current trace settings.

```
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0966I Classes active for component SER: None
SER0966I Classes active for component CMN: None
```

2 Turn on trace Class 1 and 2 for SERNET and ChangeMan ZMF:

```
/F SERT5,TRACE,ON,SER,CLASS=(1,2)
SER0850I Operator command: TRACE,ON,SER,CLASS=(1,2)
SER0964I Trace classes set
/F SERT5,TRACE,ON,CMN,CLASS=(1,2)
SER0850I Operator command: TRACE,ON,CMN,CLASS=(1,2)
SER0964I Trace classes set
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0966I Classes active for component SER: 01 02
SER0966I Classes active for component CMN: 01 02
```

3 Limit trace to user ID USER239 and USER240:

```
/F SERT5,TRACE,ON,USER=USER239
SER0850I Operator command: TRACE,ON,USER=USER239
SER0924I Trace started for user: USER239, ID: 1
/F SERT5,TRACE,ON,USER=USER240
SER0850I Operator command: TRACE,ON,USER=USER240
SER0924I Trace started for user: USER240, ID: 2
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0922I       information follows:
SER0923I Trace ID: 2 User ID: USER240
SER0923I Trace ID: 1 User ID: USER239
SER0966I Classes active for component SER: 01 02
SER0966I Classes active for component CMN: 01 02
```

CAUTION! Trace classes 1 and 2 ON for both SERNET and ChangeMan ZMF produces the most verbose output. Over 350 lines of trace data is displayed in the SERPRINT data set when one user executes the following steps:

1 Logon to ChangeMan ZMF

2 Package list for the ACTP application

3 Log off ChangeMan ZMF

4 Turn off Class 2 trace for ChangeMan ZMF:

```
/F SERT5,TRACE,OFF,CMN,CLASS=(2)
SER0850I Operator command: TRACE,OFF,CMN,CLASS=(2)
SER0964I Trace classes set
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0922I       information follows:
SER0923I Trace ID: 2 User ID: USER240
SER0923I Trace ID: 1 User ID: USER239
SER0966I Classes active for component SER: 01 02
SER0966I Classes active for component CMN: 01
```

5 Turn off Class 1 trace for ChangeMan ZMF:

```
/F SERT5,TRACE,OFF,CMN,CLASS=(1)
SER0850I Operator command: TRACE,OFF,CMN,CLASS=(1)
SER0964I Trace classes set
```

```
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0922I       information follows:
SER0923I Trace ID: 2 User ID: USER240
SER0923I Trace ID: 1 User ID: USER239
SER0966I Classes active for component SER: 01 02
SER0966I Classes active for component CMN: None
```

6 Turn turn off the USER239 trace limit and the USER240 trace limits:

```
/F SERT5,TRACE,OFF,ID=1
SER0850I Operator command: TRACE,OFF,ID=1
SER0883I TRACE ID 1 turned off
/F SERT5,TRACE,OFF,USER=USER240
SER0850I Operator command: TRACE,OFF,USER=USER240
SER0883I TRACE ID 2 turned off
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0966I Classes active for component SER: 01 02
SER0966I Classes active for component CMN: None
```

7 Turn off Class 1 and 2 trace for SERNET:

```
/F SERT5,TRACE,OFF,SER,CLASS=(1,2)
SER0850I Operator command: TRACE,OFF,SER,CLASS=(1,2)
SER0964I Trace classes set
/F SERT5,TRACE,DISPLAY
SER0850I Operator command: TRACE,DISPLAY
SER0966I Classes active for component SER: None
SER0966I Classes active for component CMN: None
```

Appendix F

Technical Notes

This appendix contains technical information that may be helpful when installing ChangeMan ZMF.

TCP/IP Trouble Shooting	188
Trusted User ID For ALM Logon To ChangeMan ZMF	189

TCP/IP Trouble Shooting

This section provides advice to the installer responsible for making SERNET connect to the TCP/IP address space.

Verifying the TCP/IP Port

A common concern early in the effort is determining whether SERNET is listening on the proper port. You can enter the following command on the ISPF Command Shell command line, or at the TSO READY prompt, to display a list of sockets and their corresponding states:

```
NETSTAT ALLCONN
```

The NETSTAT command can produce a high volume of output in a large network. The REPORT option can be added to the command to write the output to a data set. The syntax for adding the REPORT option is:

```
NETSTAT ALLCONN REPORT
```

The naming convention for the output data set is:

```
userid.NETSTAT.ALLCONN
```

Detecting Errors in the TCP/IP API

All error conditions in the TCP/IP API are written to //SERPRINT. A complete listing of all TCP/IP calls is available by toggling the TEST keyword option. Contact Serena Technical Support about using TEST.

Each call to the API is detailed in messages with the following format:

```
SERs000I user-ID TCP/IP function RC=rc,ERRNO=errno
```

To find the specific reason for the error, locate the error number (errno) in Appendix D.1 of the IBM publication *TCP/IP for MVS: API Reference*. Function (function) and return code (rc) offer further advice.

Verifying Network Connectivity

Use the PING command to verify network connectivity to a specific IP address.

```
PING P390MVS

PING 192.3.255.21
```

NOTE PING only verifies that network connectivity exists and is not completely blocked by a firewall. It cannot determine if a specific port number is available for connection.

ERRORNO 156 - Missing OMVS Segment

Failure to add an OMVS segment to the user ID associated with a SERNET instance results in the following error message during SERNET initialization:

```
SERA000E XCH TCP/IP INITAPI: RC=00001,ERRNO=00156
```

See "Access to TCP/IP Functions" on page 43.

Trusted User ID For ALM Logon To ChangeMan ZMF

There are two ways to define an alternate or "trusted" user ID for ChangeMan ZMF to allow impersonation:

- Code the user ID in table ACF2BAT in local security routine SERLCSEC. Using this method, you can enable the user ID for particular subsystem IDs or for all subsystem IDs.
- Permit the user ID READ access to resource SERENA.SERNET.AUTHUSR in the FACILITY class. If you want to use a different resource in the FACILITY class, you can change the resource name coded in security exit program SERLCSEC.

By using the FACILITY class, your security administrator can manage impersonation without changing anything in ChangeMan ZMF.

Appendix G

ZDDOPTS: ChangeMan ZDD XML Options

This chapter tells you how to customize the behavior of the ChangeMan ZDD client when accessing a ChangeMan ZMF server.

Introduction	192
AUDIT, DEMOTE, and PROMOTE Options	196
BUILD Options	199
COMMAND Options	205
LIBTYPE Options	207
PKGCREAT Options	209
PKGPROP Options	213

Introduction

This appendix describes the ChangeMan ZDD settings that can be specified for the mainframe server started tasks. These settings apply to ChangeMan ZMF functions accessed through ZDD.

These settings are specified in XML format and stored as members of a PDS library. This library must be allocated to the ChangeMan ZMF server started task using ddname ZDDOPTS:

```
//ZDDOPTS DD DSN=somnode.ZDDOPTS,DISP=SHR
```

The ZDDOPTS library is allocated to the started task to which the settings apply.

NOTE Skip this appendix if any of these apply:

- You do not license ChangeMan ZMF.
- You will not use ChangeMan ZDD to access ZMF.
- Your version of ZMF is not 5.3.6 or higher.

The first time you initiate a ZDD function that uses these options, the following actions are performed:

1. XML pages are read from the appropriate ZDDOPTS member on the mainframe.
2. XML is parsed to get the option values.
3. Option values are stored in the Windows registry on your PC.

Option information is deleted from the registry when you log off from Windows or restart Windows.

IMPORTANT! If you update a ZDDOPTS member, you must do a refresh options from the ZDD client for the new settings to take effect:

1. In Windows Explorer, right-click either a ChangeMan ZDD server node or a ChangeMan ZMF instance.
2. On the Serena Network shortcut menu, click Refresh Options.

It is not necessary to restart the started task.

If you want to use the same options for all servers, allocate the same ZDDOPTS library to all servers. If you want unique behavior for a server, code a different library name at ddname ZDDOPTS.

ZDDOPTS Members

The following members can be included in the ZDDOPTS library.

Member Name	Purpose
AUDIT	User-defined options for audit package
BUILD	Default settings for build, recompile, and relink dialogs
COMMAND	Enables or disables menu commands
DEMOTE	User-defined options for demote package

Member Name	Purpose
LIBTYPE	Shows or hides ChangeMan library types
PKGCREAT	User-defined options for package creation
PKGPROP	Allow or disallow package property fields to be changed
PROMOTE	User-defined options for promote package

XML Syntax

Members in the ZDDOPTS library contain well-formed XML pages, but this XML does not follow all of the same syntax rules that apply to Serena XML Services.

The top level XML element for each ZDDOPTS XML page is the <options> element:

```
<options name="XXXXXXXX" strict=...>
...
</options>
```

...where XXXXXXXX is the name of the ZDDOPTS member.

Element names (example: <options>) and attribute names (example: name=) in ZDDOPTS members are case sensitive and should always be specified using lower case. Attribute values in double quotes are not case sensitive and can be specified in upper, lower, or mixed case.

The optional **strict="Y"** or **strict="N"** attribute above specifies whether unrecognized keywords or element names are ignored. The default is **strict="N"**, which ignores unrecognized keywords. This default facilitates migrating to new versions of ChangeMan ZDD that support new keywords or element names. By ignoring unrecognized keywords, older ZDD clients can still be used without displaying an error message when a ZDDOPTS member is read.

There may be times, however, when you want to specify **strict="Y"**. With **strict="N"**, misspelled keywords or element names will simply be ignored. If you have a ZDDOPTS member that doesn't appear to be working correctly, specify **strict="Y"** so that any misspelled names will be reported as errors.

The **strict** attribute can be specified in the **<options>** element of any ZDDOPTS member.

Multiple Attribute Values

Some attributes allow multiple values to be specified. When specifying multiple values, separate the values with spaces, for example:

```
<profile application="TST* DEMO X*">
```

Most attributes on <profile> elements accept multiple values.

NOTE All ChangeMan ZDD users must be at version 3.2 or higher before specifying multiple attribute values in any ZDDOPTS member. Earlier versions of ZDD do not support multiple values and this will be considered a syntax error.

Wildcard Patterns

Some attribute values allow wildcard patterns to be specified. You can use the following wildcard characters in the pattern string:

Character	Function
*	Matches zero or more characters
?	Matches a single character

For example:

- The pattern "A*" would match any string that starts with the letter A.
- The pattern "*Z" would match any string that ends in the letter Z.
- The pattern "A*Z" would match any string that starts with A and ends with Z.
- The pattern "A??D" would match a string that starts with A, followed by exactly two characters, and followed by D.
- The pattern "*" matches any string.

The following table gives examples of how wildcard patterns work.

Pattern	Description	Match	No Match
A*	Match any string that starts with the letter A.	A AB ABCDEFGH	B BA
*Z	Match any string that ends in the letter Z.	Z WXYZ	ZA AZA
A*DE	Match a string that starts with A and ends with the letters DE.	ABCDE AXXXXXDE	ABCDEF
A??DE	Match a string that starts with A, followed by exactly two characters, followed by DE.	ABCDE AXYDE	ABCDEF AXYZDE
*	Match any string.	Z ABCDEFGH	

Translating Comments in ZDDOPTS Members

IMPORTANT! This section describes modifications you may need to make if you are using certain code pages on your mainframe.

ZDDOPTS members use the following XML syntax for comments:

```
<!-- xxx -->
```

If you use certain code pages on your mainframe, and if you follow the standard procedure in Chapter 4, "Unloading the Software" on page 51 to populate the SERCOMC SAMPXML library on the mainframe, the ! in XML comments may be changed to a different character. This change renders the XML invalid.

This table shows how ! is translated for each code page:

Code Page	Language	Translation
00424	Hebrew	!
00875	Greek	]
01140	English-US, Portuguese	!
01141	German	Ü
01142	Danish, Norwegian	_
01143	Finish, Swedish	_
01144	Italian	é
01145	Spanish	]
01146	English-UK	!
01147	French	§
01148	Malay	]
01149	Icelandic	Æ
01153	Polish, Hungarian	]
01154	Russian	]
01155	Turkish	Đ
01156	Latvian, Lithuanian	!
01157	Estonian	_
01158	Ukrainian	]
01164	Vietnamese	]

When ! is changed to a different character, users see the following error message (or something similar) the first time they execute a ZMF function that uses a ZDDOPTS member with a comment:

```
ChangeMan [ChangeMan ZMF instance] on server [ChangeMan ZDD server]
    has invalid XML data specified for ZDDOPTS in [library(member)].
Missing equals sign between attribute and attribute value.
```

There are two solutions to this problem:

- On the mainframe, manually edit the members that you copy from the SERCOMC SAMPXML library to the ZDDOPTS library to fix the ! in comments. The hexadecimal code for ! is 5A.

  or

- Manually upload the sample ZDDOPTS files *as text files* to the mainframe from the workstation where you installed the ChangeMan ZDD client. The translation of ASCII text to hexadecimal will preserve the ! in comments. The default location for ZDDOPTS members on your workstation is:

  C:\Program Files\Serena\ChangeMan ZDD\Samples\ZddOpts

AUDIT, DEMOTE, and PROMOTE Options

The AUDIT, DEMOTE, and PROMOTE members of the ZDDOPTS library specify user-defined options for the Audit, Demote, and Promote Package wizards. Each of these members contains one or more profiles. Each profile is a complete set of options for the corresponding wizard.

Profiles are specified by application. You may specify one or more applications associated with each profile. Applications may be specified using wildcard patterns. The first matching profile is the one used.

The structure of an AUDIT, DEMOTE, or PROMOTE member is as follows:

```
<options name= ...>    (“AUDIT”, “DEMOTE”, or “PROMOTE”)

  <profile ...>
    <field name= .../>
    <field name= .../>
  </profile>

  <profile ...>
    <field name= .../>
    <field name= .../>
  </profile>

</options>
```

Elements

This table shows the elements for the AUDIT, DEMOTE, and PROMOTE members.

Element	Description	Attributes
options	Top-level document element	name strict
profile	Complete set of user variables for the audit, demote, or promote package wizards.	application
field	Properties for a particular field.	name length default readonly label required uppercase validation

Attributes for <options> Element

These are the attributes for the <options> element.

Attribute	Description	Values	Multiple Values	Length	Default
name	ZDDOPTS member name	“AUDIT” “DEMOTE” “PROMOTE”	No	8	Value required
strict	Report error or ignore unrecognized keywords	"Y" or "N"	No	1	"N"

Attributes for <profile> Element

This table shows the attributes for the <profile> element. Attributes can be coded in any order.

Attribute	Description	Values	Multiple Values	Length	Default
application	Application mnemonic	Wildcard pattern	Yes	0 - 4	“*”

Attributes for <field> Element

This table shows attributes for the <field> element. Attributes can be coded in any order.

Attribute	Description / Values	Multiple Values	Length	Default
name	Field name. Values are described in "Name Values for <field> Element" on page 199.	No	1 - 16	Value required

Attribute	Description / Values	Multiple Values	Length	Default
length	Maximum number of characters. Valid values: 1 - 72	No	2	Field length
default	Default value for field.	No		Field is left blank
readonly	Determines whether the field can be changed by users. "Y" Field cannot be changed. "N" Field can be changed.	No	1	"N"
label	Defines the field label. Can be any text, any case. The number of characters that are actually displayed depends on the size of the display field.	No	0 – 32	Value required
required	Determines whether a field is required or can be left blank. "Y" This field is required and cannot be left blank. "N" This field can be left blank.	No	1	"N"
uppercase	Determines whether the case of the field will remain as entered or will be changed to upper case. "Y" Field will be changed to upper case. "N" Field will remain as entered.	No	1	"N"
validation	Validation performed on data entered in the field. "YN" Y or N "ALPHA" A-Z or national (#, $, @) "ALPHANUM" A-Z, national (#, $, @), 0-9 "NUMBER" 0-9 "SYMBOL" Same as ALPHANUM, except first character must be alphabetic or national. "DSNAME Valid data set name "NONE" No validation performed	No	0 – 8	"NONE"

Name Values for <field> Element

This table describes values for the name attribute of the <field> element.

Name Value	Description	User Defined	Values	Length
UserVariable01 - UserVariable05	User options	Yes	Any text, subject to validation	0-8
UserVariable06 - UserVariable10	User options	Yes	Any text, subject to validation	0-72

Example

Sample AUDIT, DEMOTE, and PROMOTE members are delivered with the ZDD server components in the SERCOMC SAMPXML library.

You can also view the sample members, formatted by your default XML browser, on your desktop workstation in file:

```
C:\Program Files\Serena\ChangeMan ZDD Client\Samples\ZddOpts\
    member.xml
```

where *member* is AUDIT, DEMOTE, or PROMOTE.

BUILD Options

The BUILD member of the ZDDOPTS library specifies default settings for the Build, Recompile, and Relink dialog boxes. The BUILD member includes one or more profiles. Each profile is a complete set of rules for the Build dialog.

Profiles are specified by application, language, build procedure, and library type. Any or all of these can be specified using wildcard patterns. The first matching profile is the one used.

The structure of a BUILD member is as follows:

```
<options name="BUILD">

  <multiple init= .../>

  <profile ...>
    <field name= .../>
    <field name= .../>
  </profile>

  <profile ...>
    <field name= .../>
    <field name= .../>
  </profile>

</options>
```

Elements

This table shows the elements for the BUILD member.

Element	Description	Attributes
options	Top-level document element.	name strict
multiple	Specifies how the Build dialog box fields are to be initialized when multiple components are being built at the same time.	init
profile	Complete set of fields for Build dialog box.	application libtype language procname
field	Properties for a particular field in the dialog box.	name default label length readonly required uppercase validation

Attributes for <options> Element

This table shows the attributes for the <options> element.

Attribute	Description	Values	Multiple Values	Length	Default
name	ZDDOPTS member name	"BUILD"	No	5	Value required
strict	Report error or ignore unrecognized keywords	"Y" or "N"	No	1	"N"

Attributes for <multiple> Element

Attributes for the <multiple> element are shown below.

Attribute	Description	Values	Multiple Values	Length	Default
init	Field initialization for multiple components	"None" "First" "All"	No	0 - 5	"All"

The <multiple> element specifies how the Build dialog box fields are to be initialized when multiple components are being built at the same time. The following table describes the values that can be specified for the **init** attribute:

Values	Description
None	None of the dialog box fields will be filled in.
First	Dialog box fields will be filled in with values from the first component for which history or a designated build procedure can be found.
All	History for all of the components will be examined. Those fields that contain the same value for all components will be filled in. Fields for which values differ from component to component will be left blank.

Attributes for <profile> Element

This table shows the attributes for the <profile> element. Attributes can be coded in any order.

Attribute	Description	Values	Multiple Values	Length	Default
application	Application mnemonic	Wildcard pattern	Yes	0 - 4	"*"
libtype	Library type	Wildcard pattern	Yes	0 – 3	"*"
language	Language	Wildcard pattern	Yes	0 – 8	"*"
procname	Build procedure	Wildcard pattern	Yes	0 – 8	"*"
name	Profile name		No	1 - 8	
function	Build functions	"Build" "Recomp" "Relink" "Query"	Yes		

Attributes for <field> Element

Attributes for the <field> element are shown in the following table. Attributes can be coded in any order.

NOTE Some of the attributes apply only to user-defined fields that display on the User Options page of the Build wizard. They are: "label", "length", "required", "uppercase", and "validation".

Attribute	Description / Values	Multiple Values	Length	Default
name	Field name. Values are described in "Name Values for <field> Element" on page 203.	No	1 - 16	Value required
default	Defines what value is automatically inserted in the Build dialog field when the field is initially blank.	No	Depends on the field	Field is left blank
readonly	Determines whether the field on the Build dialog can be changed by users. "Y" Field cannot be changed. "N" Field can be changed.	No	1	"N"
label	Defines the field label on the Build dialog. Can be any text, any case. The number of characters that are actually displayed depends on the width of the displayed characters. **NOTE** This is an attribute for the UserOption fields only.	No	0 – 32	Value required
length	The maximum number of characters allowed for entry in a UserOption field. Cannot exceed the field size defined for that particular UserOption. See "Name Values for <field> Element" on page 203 for specific lengths. **NOTE** This is an attribute for the UserOption fields only.	No	2	Max. length defined for a User Option field.
required	Determines whether a field is required, whether a field can be left blank. **NOTE** This is an attribute for the UserOption fields only. "Y" This field is required and cannot be left blank. "N" This field can be left blank. Delivered ZMF skeletons process blank User Options as if they were set to "N".	No	1	"N"
uppercase	Determines whether text will be folded to upper case. **NOTE** This is an attribute for the UserOption fields only. "Y" Text will be folded to uppercase. "N" Text will remain as entered (no conversion to uppercase).	No	1	"N"

Attribute	Description / Values		Multiple Values	Length	Default
validation	Validation performed on data entered in this field. **NOTE** This is an attribute for the UserOption fields only.		No	0 – 8	"NONE"
	"YN"	Y or N			
	"ALPHA"	A-Z or national (#, $, @)			
	"ALPHANUM"	A-Z, national (#, $, @), 0-9			
	"DSNAME	Valid data set name			
	"NUMBER"	0-9			
	"SYMBOL"	Same as ALPHANUM, except first character must be alphabetic or national.			
	"NONE"	No validation performed			

Name Values for <field> Element

This table describes the values that may be specified for the **name** attribute of the <field> element.

Name Value	Description	User Defined	Values	Length
Language	Language name	No	Alphanumeric	0-8
BuildProc	Build procedure name	No	Alphanumeric	0-8
Db2PreCompile	DB2 precompile option	No	"Y" or "N"	1
Db2Subsystem	DB2 subsystem name	No	Alphanumeric	0-4
CompileOptions	Compile parameters	No	Any text	0-34
LinkOptions	Link parameters	No	Any text	0-34
UserOption01 – UserOption20	User options	Yes	Any text, subject to validation	0-1
UserOption101 – UserOption105	User options	Yes	Any text, subject to validation	0-1
UserOption201 – UserOption203	User options	Yes	Any text, subject to validation	0-2
UserOption301 – UserOption303	User options	Yes	Any text, subject to validation	0-3
UserOption401 – UserOption403	User options	Yes	Any text, subject to validation	0-4
UserOption801 – UserOption805	User options	Yes	Any text, subject to validation	0-8
UserOption1001 – UserOption1002	User options	Yes	Any text, subject to validation	0-10
UserOption1601 – UserOption1602	User options	Yes	Any text, subject to validation	0-16

Name Value	Description	User Defined	Values	Length
UserOption3401 – UserOption3402	User options	Yes	Any text, subject to validation	0-34
UserOption4401 – UserOption4402	User options	Yes	Any text, subject to validation	0-44
UserOption6401 – UserOption6405	User options	Yes	Any text, subject to validation	0-64
UserOption7201 – UserOption7205	User options	Yes	Any text, subject to validation	0-72

Example

A sample BUILD member is delivered with the ZDD server components in the SERCOMC SAMPXML library.

You can also view the BUILD sample, formatted by your default XML browser, on your desktop workstation in file:

```
C:\Program Files\Serena\ChangeMan ZDD Client\Samples\ZddOpts\
    Build.xml
```

COMMAND Options

The COMMAND member of the ZDDOPTS library allows certain menu commands to be disabled in the ChangeMan ZDD user interface. The COMMAND member includes one or more profiles. Each profile is a complete set of command settings.

Profiles can be specified by either user ID or security group name. Using group name requires that the server be at the SerNet 7.1.3+ level.

If both user= and group= filters are specified in the profile, the profile will be selected if either filter matches. Both the user ID and group name filters may contain wildcard patterns. The first matching profile will be the one used.

The structure of a COMMAND member is as follows:

```
<options name="COMMAND">

  <profile ...>
    <command name= .../>
    <command name= .../>
  </profile>

  <profile ...>
    <command name= .../>
    <command name= .../>
  </profile>

</options>
```

Elements

The following table shows the elements for the COMMAND member.

Element	Description	Attributes
options	Top-level document element.	name strict
profile	Complete set of command definitions.	user
command	Enable or disable menu commands.	name enable

Attributes for <options> Element

The following table shows the attributes for the <options> element.

Attribute	Description	Values	Multiple Values	Length	Default
name	ZDDOPTS member name	"COMMAND"	No	7	Value required
strict	Report error or ignore unrecognized keywords	"Y" or "N"	No	1	"N"

Attributes for <profile> Element

The following table shows the attributes for the <profile> element.

Attribute	Description	Values	Multiple Values	Length	Default
user	z/OS user ID(s)	Wildcard pattern	Yes	0 – 7	"*"
group	z/OS security group(s) *(requires SerNet 7.1.3+)*	Wildcard pattern	Yes	0 – 7	None

Attributes for <command> Element

The following table shows the attributes for the <command> element. Commands are always enabled by default.

The command name may contain wild characters. If a command name matches multiple <command> elements with wild character patterns, the first matching <command> element is used for that command.

Attribute	Description	Values	Multiple Values	Length	Default
name	Command type	"ApprovePackage" "AuditPackage" "BackoutPackage" "BuildComponent" "ChangePackageDate" "ChangePackageStatus" "ChangeSchedule"	No	1 – 16	Value required

Attribute	Description	Values	Multiple Values	Length	Default
name (continued)	Command type	"CheckInComponent" "CheckOutComponent" "ClosePackage" "CreatePackage" "DeleteComponent" "DeletePackage" "Demote" "FreezePackage" "HoldSchedule" "LockComponent" "OpenPackage" "Promote" "RebuildJcl" "RecompileComponent" "RefreezeComponent" "RefreezePackage" "ReleaseSchedule" "RelinkComponent" "RenameComponent" "ResetAuditLock" "RevertPackage" "ScratchComponent" "SubmitXml" "UndeletePackage" "UnfreezeComponent" "UnfreezePackage" "UnlockComponent" "UploadDataSet" "UploadUnixFile" "*"	No	1 – 16	Value required
enable	Indicates whether command is enabled or disabled.	"Y" or "N"	No	1	"Y"

Example

A sample COMMAND member is delivered with the ZDD server components in the SERCOMC SAMPXML library.

You can also view the COMMAND sample, formatted by your default XML browser, on your desktop workstation in file:

```
C:\Program Files\Serena\ChangeMan ZDD Client\Samples\ZddOpts\
    Command.xml
```

LIBTYPE Options

You can suppress the display of application library types in ChangeMan ZDD with XML entries in the LIBTYPE member of the ZDDOPTS library. Use the LIBTYPE member to

emulate customization in ZMF exit program CMNEX035 you have made to suppress the display of "hidden" library types.

NOTE ZMF exit program CMNEX035 contains multiple tables to suppress the display of library types in particular functions such as checkout, stage, browse compressed listing, and browse baseline. The LIBTYPE member of ZDDOPTS suppresses all displays of specified library types in baseline and package folders in the specified applications, effectively disabling all ZMF functions for those library types in the specified applications.

The LIBTYPE member includes one or more profiles. Each profile is a complete set of library type settings for an application. Application can be specified using wildcard patterns. The first matching profile is the one used.

The structure of a LIBTYPE member is as follows:

```
<options name="LIBTYPE">

  <profile ...>
    <libtype name= .../>
    <libtype name= .../>
  </profile>

  <profile ...>
    <libtype name= .../>
    <libtype name= .../>
  </profile>

</options>
```

Elements

This table show the elements for the LIBTYPE member.

Element	Description	Attributes
options	Top-level document element.	name strict
profile	Complete set of library type definitions.	Application
libtype	Show or hide properties for a particular library type. Library type can be specified using a wildcard pattern.	name show

Attributes for <options> Element

These are the attributes for the <options> element.

Attribute	Description	Values	Multiple Values	Length	Default
name	ZDDOPTS member name	"LIBTYPE"	No	7	Value required
strict	Report error or ignore unrecognized keywords	"Y" or "N"	No	1	"N"

Attributes for <profile> Element

These are the attributes for the <profile> element.

Attribute	Description	Values	Multiple Values	Length	Default
application	Application name	Wildcard pattern	Yes	1 - 4	"*"

Attributes for <libtype> Element

This table shows the attributes for the <field> element. The default is to show all a library types.

Attribute	Description / Values	Multiple Values	Length	Default
name	3 character library type or wildcard pattern.	No	1-3	None
listing	Listing library type that corresponds to a source library type.	No	1-3	"LST"
show	Indicates whether library type is shown or hidden. "Y" Show library type "N" Hide library type	No	1	"Y"

Example

A sample LIBTYPE member is delivered with the ZDD server components in the SERCOMC SAMPXML library.

You can also view the LIBTYPE sample, formatted by your default XML browser, on your desktop workstation in file:

```
C:\Program Files\Serena\ChangeMan ZDD Client\Samples\ZddOpts\
    LibType.xml
```

PKGCREAT Options

The PKGCREAT member of the ZDDOPTS library specifies user-defined options for the New Package wizard. The PKGCREAT member includes one or more profiles. Each profile is a complete set of rules for the New Package wizard.

Profiles are specified by application. You may specify one or more applications associated with each profile. Applications may be specified using wildcard patterns. The first matching profile is the one used.

The structure of a PKGCREAT member is as follows:

```
<options name="PKGCREAT">
```

```
    <profile ...>
      <field name= .../>
      <field name= .../>
    </profile>

    <profile ...>
      <field name= .../>
      <field name= .../>
    </profile>

  </options>
```

Elements

This table shows the elements for the PKGCREAT member.

Element	Description	Attributes
options	Top-level document element.	name strict
profile	Complete set of user variables for the New Package wizard.	application
field	Properties for a particular field in the New Package dialog boxes.	name length default readonly label required uppercase validation

Attributes for <options> Element

These are the attributes for the <options> element.

Attribute	Description	Values	Multiple Values	Length	Default
name	ZDDOPTS member name	"PKGCREAT"	No	8	Value required
strict	Report error or ignore unrecognized keywords	"Y" or "N"	No	1	"N"

Attributes for <profile> Element

This table shows the attributes for the <profile> element. Attributes can be coded in any order.

Attribute	Description	Values	Multiple Values	Length	Default
application	Application mnemonic	Wildcard pattern	Yes	0 - 4	"*"

Attributes for <field> Element

This table shows attributes for the <field> element. Attributes can be coded in any order.

Attribute	Description / Values	Multiple Values	Length	Default
name	Field name. Values are described in "Name Values for <field> Element" on page 212.	No	1 - 16	Value required
length	Maximum number of characters. Valid values: 1 - 72	No	2	Field length
default	Default value for field.	No		Field is left blank
readonly	Determines whether the field can be changed by users. "Y" Field cannot be changed. "N" Field can be changed.	No	1	"N"
label	Defines the field label. Can be any text, any case. The number of characters that are actually displayed depends on the size of the display field.	No	0 – 32	Value required
required	Determines whether a field is required or can be left blank. "Y" This field is required and cannot be left blank. "N" This field can be left blank.	No	1	"N"
uppercase	Determines whether the case of the field will remain as entered or will be changed to upper case. "Y" Field will be changed to upper case. "N" Field will remain as entered.	No	1	"N"
validation	Validation performed on data entered in the field. "YN" Y or N "ALPHA" A-Z or national (#, $, @) "ALPHANUM" A-Z, national (#, $, @), 0-9 "DSNAME Valid data set name "NUMBER" 0-9 "SYMBOL" Same as ALPHANUM, except first character must be alphabetic or national. "NONE" No validation performed	No	0 – 8	"NONE"

Name Values for <field> Element

This table describes values for the name attribute of the <field> element.

Name Value	Description	User Defined	Values	Length
UserVarLen101 - UserVarLen199	User options	Yes	Any text, subject to validation	0-1
UserVarLen201 - UserVarLen211	User options	Yes	Any text, subject to validation	0-2
UserVarLen301 - UserVarLen310	User options	Yes	Any text, subject to validation	0-3
UserVarLen401 - UserVarLen410	User options	Yes	Any text, subject to validation	0-4
UserVarLen801 - UserVarLen810	User options	Yes	Any text, subject to validation	0-8
UserVarLen1601 - UserVarLen1605	User options	Yes	Any text, subject to validation	0-16
UserVarLen4401 - UserVarLen4405	User options	Yes	Any text, subject to validation	0-44
UserVarLen7201 - UserVarLen7205	User options	Yes	Any text, subject to validation	0-72

Example

A sample PKGCREAT member is delivered with the ZDD server components in the SERCOMC SAMPXML library.

You can also view the PKGCREAT sample, formatted by your default XML browser, on your desktop workstation in file:

```
C:\Program Files\Serena\ChangeMan ZDD Client\Samples\ZddOpts\
    PkgCreat.xml
```

PKGPROP Options

The PKGPROP member of the ZDDOPTS library is used to disallow users from changing selected fields of the package properties. The PKGPROP member includes one or more profiles. Each profile is a complete set of property field settings.

Profiles can be specified by either user ID or security group name. Using group name requires that the server be at the SerNet 7.1.3+ level.

If both user= and group= filters are specified in the profile, the profile will be selected if either filter matches. Both the user ID and group name filters may contain wildcard patterns. The first matching profile will be the one used.

The structure of a PKGPROP member is as follows:

```
<options name="PKGPROP">

  <profile ...>
    <field name= .../>
    <field name= .../>
  </profile>

  <profile ...>
    <field name= .../>
    <field name= .../>
  </profile>

</options>
```

Elements

This table shows the elements for the PKGPROP member.

Element	Description	Attributes
options	Top-level document element.	name strict
profile	Complete set of field definitions.	user
field	Allow or disallow field to be changed.	name readonly

Attributes for <options> Element

This table shows the attributes for the <options> element.

Attribute	Description	Values	Multiple Values	Length	Default
name	ZDDOPTS member name	"PKGPROP"	No	7	Value required
strict	Report error or ignore unrecognized keywords	"Y" or "N"	No	1	"N"

Attributes for <profile> Element

This table shows the attributes for the <profile> element.

Attribute	Description	Values	Multiple Values	Length	Default
user	Mainframe user ID	Wildcard pattern	Yes	0 - 7	"*"
group	z/OS security group(s) *(requires SerNet 7.1.3+)*	Wildcard pattern	Yes	0 – 7	None

Attributes for <field> Element

This table shows the attributes for the <field> element. Attributes can be coded in any order.

Attribute	Description	Values	Multiple Values	Length	Default
name (continued)	Field name	"PackageTitle" "PackageLevel" "RequestorName" "RequestorPhone" "WorkRequestId" "Department" "UnplannedReason" "TemporaryDays" "Release" "PackageDescription"	No	1 – 21	Value required
name (continued)	Field name	"ProblemAction" "Scheduler" "InstallInstructions" "PredessorJobs" "SuccessorJobs" "AffectedApplications" "InstallSite" "InstallDateTime" "ContactName" "ContactPhone" "ParticipatingPackages" "UserOptions" "*"	No	1 – 21	Value required
readonly	Determines whether the field can be changed by users.	"Y" Field cannot be changed. "N" Field can be changed.	No	1	"N"

Example

A sample PKGPROP member is delivered with the ZDD server components in the SERCOMC SAMPXML library.

You can also view the PKGPROP sample, formatted by your default XML browser, on your desktop workstation in file:

```
C:\Program Files\Serena\ChangeMan ZDD
    Client\Samples\ZddOpts\PkgProp.xml
```

Index

Symbols

! in ZDDOPTS members 194
$CHGMAN resource class 89

A

abend
    ESTAE recovery 156
    S0C4 26, 32
    S878 25
ABEND modify command 168
Abend-AID 27
ACF2
    add ChangeMan ZMF 82
    batch logon ID 81
    job statement skeletons 104
    logon ID 104
address mode 25
address space
    non-swappable 32
Adobe Acrobat 11
allocate
    component master 62, 128
    delay file 62, 128
    impact analysis file 72
    log file 62, 128
    package master 62, 128
    recovery file 62, 128
    XMLSPACE 63
APF authorization
    dynamic APF list 78
    load libraries 28, 60, 78
    static APF list 78
ASID
    keyword option 152
    maximum concurrent 152, 168, 169
    modify command 168
    reusable 32, 98
    subsystem 32
ASIDS
    maximum concurrent 169
    modify command 169
ASINITi keyword option 153
ASINITi modify command 169
ATTACH modify command 170
attributes
    multiple values 193
AUDIT member 196–199
authority levels, table 42
automated operations tools 114
AUTOMESSENGER keyword option 153

B

batch jobs
    considerations 46
    install job names 47
    user job names 46
Batch Local Shared Resources (BLSR) 33, 73
buffers, tracing 181
BUILD member 199–204

C

cataloged procedures
    build 75
change package
    see package
ChangeMan ZDD XML Options 191
ChangeMan ZMF
    customization 48
    development protection 16
    implementation strategy 20
    started task 172
    system requirements 24
    upgrades 32
client
    customizing 192–215
CMN$$IAX impact analysis skeleton 109
CMNADSP file tailoring procedure 43
CMNEX008 exit program 47
CMNEXINS exit program 44
    customize 77
code pages
    translating exclamation point in ZDDOPTS 194
COMMAND member 205–207
common security interface 43
Common System Area (CSA) 25
component master
    access statistics 176
    allocation 62, 128
COMPRESS keyword option 154
connect to SERNET 170

console log messages 49, 114
customizing client 192–215

D

DASD
  migrated datasets 160
  system requirements 25
  volume serial ID 160
data compression 154
dataset considerations
  access authority table 91
  CMNEXINS exit program 44
  DASD management 45
  naming conventions 45, 77
  planning for install 43
  staging model datasets 45
  temporary list datasets 44
  utility datasets 44, 45
DB2 keyword option 155
DB2 Option
  detach SERNET 171
  SERNET keywords 155
DDNAME keyword option 156
delay file
  access statistics 176
  allocation 62, 128
DEMOTE member 196–199
DETACH (DB2) modify command 171
disconnect from DB2 subsystem 171
display
  file access statistics 176
  users 178

E

eNetwork Communications Server
  error 156 188
enqueues 174
ESTAE keyword option 156
EX003 keyword option 157
exclamation point
  translating in ZDDOPTS 194
exit programs
  CMNEX008 47
  CMNEXINS 44, 77
  Librarian exit program 158
  SEREX003 157
expand files
  automated 57
  manual 139
EXPIRE keyword option 156
EXTEND modify command 172

F

file access methods
  IAM 157
  VSAM 33
  VSAM linear dataset (LDS) 63
file access statistics 176
file tailoring
  CMNADSP procedure 43
  security considerations 43
file transfer
  3270 emulation 137
  automated 55
  binary files 136
  FTP 136
  manual 136

H

HALT modify command 172
HIPER-CACHE 28
HiperLoad Plus 28
HOLD modify command 173

I

IAM keyword option 157
IEX keyword option 158
impact analysis file
  CMN$$IAX maintenance skeleton 109
  sequential file 72
IMS keyword option 158
IMS subsystem 158
installation
  allocate SER#PARM address file 70
  APF load library authorization 78
  apply ChangeMan ZMF license 61
  build cataloged procedures 75
  build logon CLIST 76
  build started task JCL 72
  define dataset access authority 91
  define staging versions files 65
  define VSAM files 61
  define XMLSPACE dataset 63
  initialize package master 61
  install job names 47
  IPL 29, 78
  software conflicts 27
  store SVPARM control member 69
installation planning
  DASD management 45
  JCL customization 35
  JCL samples 33
  non-swappable address space 32

security 40
staging versions 48
SYSOUT authority 47
SYSTCPD DDNAME 39
system linkage index 32
TCP/IP 39
upgrades 32
utility datasets 44
z/OS subsystem 32
instances
ChangeMan ZMF 35
production instance 35
SERNET 35
subsystem ID 35
system linkage index 32
test instance 35
IPL 29, 78
ISPF
connect to ChangeMan ZMF 100
LIBDEF 100
panel library 100

J

JCL
customization 35
multiple ZMF instances 35
samples 142
started task 33
started task procedure 142
subsystem ID 35
SVPARM control member 74
VSAM optimization 33
JOBLIB 26, 27, 33

L

LCLCCSID keyword option 159, 160, 161, 164
LIB keyword option 159
Librarian 26, 158, 159
update module 159
library concatenations
APF authorization 78
CLIST 76, 142
ISPF LIBDEF 100
JOBLIB 26, 27, 33
LINKLIST 26, 27, 33
STEPLIB 26, 27, 33
LIBTYPE member 207–209
LINKLIST 26, 27, 33
load libraries
APF authorization 28, 60
LOCKS modify command 174
log file
access statistics 176
allocation 62, 128
logon
build CLIST 76
logon CLIST
build 76
ISPF LIBDEF 100
sample JCL 142

M

MAINVIEW SRM StopX37/II 28
MAPDATA dataset 64
memory management 25
messages
when using ZDDOPTS members 194
MIGRAT keyword option 160
migration dataset volume 160
modify commands
ABEND 168
ATTACH 170
DETACH (DB2) 171
EXTEND 172
HALT 172
HOLD 173
LOCKS 174
NETTRACE 181
REFRESH 176, 181
RELEASE 174
restricted 181
SHUTDOWN 174
standard 168
STATS 176
TRACE 178, 182
USERS 178
modifying skeletons
and software conflicts 111

N

NETSTAT command 188
NETTRACE modify command 181
NOSCHED keyword options 160

O

online help 13

P

package
described 16

lifecycle 17
package master
access statistics 176
allocation 62, 128
initialize 61
Panvalet 26
parameter passing 36
PDSFAST 27
PDSMAN 27
PING command 188
PKGCREAT member 209–212
PKGPROP member 213–215
PROMOTE member 196–199

R

RACF 85
RECEIVE command 57, 139
recovery file
access statistics 176
allocation 62, 128
reentrant code 25
refresh load module 176, 181
REFRESH modify command 176, 181
refreshing ZDDOPTS 192
RELEASE modify command 174
REXX
runtime environment requirements 26
RUNFOR keyword option 162

S

S0C4 abend 26, 32
S878 abend 25
SAF 43
scheduler
hold install jobs 173
release install jobs 174
suppress 160
SDNOTIFY keyword option 162
security
dataset access authority table 91
disable 82
exclusive library access 40
planning 40
requirements 24
stubbing 82
SYSOUT access rules 47
security entities
$CHGMAN resource class 89
administrator 41
authority levels 42
change manager 41
resource class 86
RSRCCL parameter 86
setup example 88
security exits
disable SERJES exit call 157
SEREX003 157
security setup
access to ChangeMan ZMF functions 40
ACF2 batch logon ID 81
add ChangeMan ZMF to security system 82
RACF 85
security entities 41
SERLCSEC exit program assembly 81
SERLCSEC exit program customization 80
SERLCSEC security exit program 82
utility datasets 44
security software
ACF2 82, 104
APF 28, 60, 78
SAF 43
Top Secret 87
SER# member 70
SER#PARM
DDNAME 37
TCP/IP address file 70
SER10TY license manager
apply license 61
SERLIC DDNAME 38
SEREX003 exit program 157
serial file locks 174
SERIFACE interface 26
SERJES security exit 157
SERLCSEC security exit program
&STUB switch 82
assembling 81
customization 80
SERLIC DDNAME 38
SERNET
address spaces 152, 153, 169
architecture 19
attach 170
connect 170
connecting to 76
delay shutdown 172
described 19
detach DB2 subsystem 171
disconnect from DB2 subsystem 171
enqueues 174
halt 172
instances 35
JCL samples 142
locks 174
logon CLIST 76
memory requirements 24
non-swappable address space 32
notifications 153
operating period 162

parameter dataset 156
parameter passing 36
region size 24
SER# member 70
SERPRINT messages 98
shutdown 101, 156, 162, 172, 174
shutdown commands, table 101
start command 98
started task 35
started task names 35
started task procedure 142
subsystem ID 100, 163
TCP/IP addressing 37, 70
timeout setting 152
trace buffers 181
trace facility 178, 182
ZDDOPTS ddname 192–215
SERNET keyword options
ASID 152
ASINITi 153
AUTOMESSENGER 153
commonly used 37
COMPRESS 154
DB2 155
DDNAME 156
ESTAE 156
EX003 157
EXPIRE 156
IAM 157
IEX 158
IMS 158
LCLCCSID 159, 160, 161, 164
LIB 159
MIGRAT 160
NOSCHED 160
PARM statement 74
required options, listed 37
RUNFOR 162
SDNOTIFY 162
special case options, listed 37
started task procedure 74
STAX 162
SUBSYS 163
TCPIP 163
XML 165
SERNET modify commands
ASINITi 169
SERPRINT
messages 98
SYSOUT class 76
session switching behavior 162
shutdown 101, 172, 174
SHUTDOWN modify command 174
skeletons
and software conflicts 111
CMN$$IAX impact analysis 109
software compatibility 26
ISPF 162
Librarian 26
Panvalet 26
SERIFACE interface 26
software conflicts 27
Abend-AID 27
HIPER-CACHE 28
HiperLoad Plus 28
MAINVIEW SRM StopX37/II 28
PDSFAST 27
PDSMAN 27
VSAM file performance tools 28
software requirements
REXX runtime environment 26
security 24
staging model dataset 45
staging versions
define VSAM files 65
delta master 65
described 48
file allocations 65
HPSPLIB DDNAME 74
installation planning 48
object master 65
site master 65
SVPARM control member 69, 74
VSAM file name table 65
start command 35, 98
started task
build JCL 72
JCL procedure 33, 142
names 35
subsystem ID 35, 100, 163
STATS modify command 176
STAX keyword option 162
STEPLIB 26, 27, 33
stop SERNET and ChangeMan ZMF 101
StopX37/II 28
stubbing security 82
SUBSYS keyword option 163
Subsystem Communications Vector Table (SSCVT) 25
subsystem ID 35, 100, 163
SVPARM control member 69
HPSPLIB ddname 74
SYSOUT
access rules 47
SYSOUT class 76
System Authorization Facility (SAF)
see SAF
system linkage index 32
system requirements 24
data transport 24
ISPF version 24
memory 24

operating system 24
software 24
storage 25

T

TCP/IP
addresses 37, 70
API errors 188
data compression 154
eNetwork Communications Server 188
error 156 188
multiple started tasks 39
NETSTAT command 188
PING command 188
port IDs 70
SER# member 70
SER#PARM address file 70
SER#PARM DDNAME 37
SERNET support 37, 70
server identification 163
SYSTCPD DDNAME 39
verify network connectivity 188
verify port ID 188
TCPIP keyword option 163
timeout 162
Top Secret
add ChangeMan ZMF 87
TPC/IP
port IDs 37
TRACE modify command 178, 182
trace SERNET buffers 181

U

unload software
expand PDS libraries 57
transfer files to host 55
user timeout 162
USERS modify command 178
utility datasets 44, 45, 77

V

VSAM file initialization
security 86
VSAM performance 33
Batch Local Shared Resources (BLSR) 33, 73
HIPER-CACHE 28
HiperLoad Plus 28
optimization 33
optimizing 33
tools 28

W

WARN 179
wildcard, patterns in XML 194

X

XML
syntax in ZDDOPTS 193
XML keyword option 165
XML Services
cautions 115
disable 165
MAPDATA dataset 64
REXX runtime environment 26
XMLLOAD member 64
XMLSPACE dataset 63
XMLSPACE DDNAME 73
XMLSERV client
MAPDATA dataset 64
REXX runtime environment 26
XMLSPACE
dataset allocation dataset 63
DDNAME 73

Z

z/OS subsystem 32
ZDD XML Options 191
ZDDOPTS 191
and code pages 194
AUDIT member 196–199
BUILD member 199–204
COMMAND member 205–207
ddname 192–215
DEMOTE member 196–199
LIBTYPE member 207–209
members 192
multiple attribute values 193
PKGCREAT member 209–212
PKGPROP member 213–215
PROMOTE member 196–199
refreshing 192
translating exclamation point 194
troubleshooting 192
updating members 192
XML syntax 193
XML wildcard patterns 194