ChangeMan® ZMF for Eclipse™
Installation and Configuration Guide

Product version: 8.1.2

Publication date: September 2018 (rebranded only)

Table of Contents

Welcome to ChangeMan® ZMF for Eclipse™

Product Description
ChangeMan® ZMF for Eclipse™ provides plug-in integration between ChangeMan ZMF, the industry-leading software change management system for IBM z Systems mainframes, and the software development environments provided by the following products:

- Eclipse™ Workbench

- IBM® Rational® Developer for z Systems® (RDz)

These products include software developed by the Eclipse Project. See:

http://www.eclipse.org)

ZMF for Eclipse is an optional licensed feature of ChangeMan ZMF. It is distributed as part of the ChangeMan ZMF Client Pack.

Before You Begin

Readme
Refer to the Readme file for the latest updates on software compatibility, patches, known issues, workarounds, and other late-breaking information before you install ChangeMan ZMF for Eclipse. This information can change. If it does, the Readme file is updated and republished between updates to this manual.

The latest Readme file for ZMF for Eclipse can be downloaded from the **ChangeMan ZMF Client Pack** product download page on the Micro Focus SupportLine website at:

https://www.microfocus.com/support-and-services/documentation/

About This Book

Audience
This manual is intended for developers and systems engineers installing ZMF for Eclipse on the desktop, and for ChangeMan ZMF administrators responsible for configuring ZMF to support ZMF for Eclipse.

This document assumes that you are familiar with the Integrated Development Environment (IDE) provided by the Eclipse Workbench or IBM Rational Developer for z Systems (RDz) on a Microsoft Windows platform. It also assumes familiarity with software change control concepts and functions implemented in ChangeMan ZMF. If you are new to ChangeMan ZMF, refer to the "Related Documentation" table below for information not included in this manual.

Scope
This *Installation and Configuration* guide documents the installation prerequisites, install procedures, and post-install customization options for ZMF for Eclipse. Examples are provided for two common development environments:

- Eclipse Workbench with Apache Tomcat® on Windows

- IBM RDz with WebSphere® Application Server on Windows

You may use ZMF for Eclipse with other operating systems and Web application servers, as described in the Readme. The many combinations of supported IDEs, Web application servers, and client operating systems make it impossible to illustrate the installation or use of ZMF for Eclipse in all possible development environments.

Disclaimers   The instructions in this manual do not replace the documentation for:

- ***Eclipse Workbench from the Eclipse Project***

    Your primary source of information about the Eclipse Workbench is the Eclipse Project website:

    http://www.eclipse.org

- ***IBM Rational Developer for z Systems (RDz)***

    The documentation provided by IBM is your primary source of information regarding those products. IBM also provides technical reference information and an online library for RDz and related products on the Rational developerWorks website:

    http://www.ibm.com/developerworks/rational/

- ***Apache Tomcat, IBM WebSphere Application Server, or other web application servers***

    Platform-specific information related to your web application server environment should be obtained from your web application server vendor. For example, your primary source of information about Apache Tomcat is the Apache Software Foundation website:

    http://tomcat.apache.org/

    Your primary source of information regarding IBM WebSphere Application Server (WAS) is the documentation provided by IBM. IBM maintains an online documentation library for WAS at:

    http://www-01.ibm.com/software/webservers/appserv/was/library/

- ***Microsoft® Windows or other client operating systems***

    Platform-specific information about your development operating system environment should be obtained from your operating system vendor. For example, your primary source of information about Microsoft Windows is the Microsoft website:

    http://www.microsoft.com

Serena makes no representations or warranties that the instructions contained in this document are valid regarding Eclipse Workbench or IBM Rational Developer for z Systems. Refer to the product documentation supplied with these products for authoritative information.

Software Compatibility   The instructions included in this document have been verified only for the software versions specified in the Readme, as made available to Micro Focus. Other versions of the Eclipse Workbench and IBM Rational Developer for z Systems may integrate with ZMF for Eclipse erratically or not at all.

Change Bars   Change bars in the left margin identify changes to this manual since the last time it was published.

Online Help and Documentation

ZMF for Eclipse includes information resources in the form of built-in help and downloadable documentation on the Micro Focus SupportLine website at:

https://www.microfocus.com/support-and-services/documentation/

Accessing Online Help

The contents of the *User's Guide* for ZMF for Eclipse are available as online help within the workbench environment.

To access the help:

1   From the workbench **Help** Menu, click **Help Contents**, and then select **ChangeMan ZMF for Eclipse User's Guide**.

2   Use the **Search** option to find specific text strings within the online help.

Navigation through the help is provided by the workbench. Please see the Eclipse or RDz product documentation for more information on help navigation.

Product Documentation

The following documentation for ZMF for Eclipse can be downloaded in Adobe Portable Document Format (PDF) from the **ChangeMan ZMF Client Pack** product download page on the Micro Focus SupportLine website at:

https://www.microfocus.com/support-and-services/documentation/

Manual	Description
ChangeMan ZMF for Eclipse Installation and Configuration Guide	Installation, configuration, and customization options for the ZMF for Eclipse plugin.
ChangeMan ZMF for Eclipse User's Guide	User's guide for ZMF for Eclipse in Eclipse and RDz developer workbench environments.
ChangeMan ZMF Web Services Getting Started Guide	Reference information concerning the Web Services integration technology used by ZMF and ZMF for Eclipse.

Related Documentation

Refer to the following documents to learn more about ChangeMan ZMF functionality and its integration with the workbench environment. These documents can be downloaded from the **ChangeMan ZMF** product download page on the Micro Focus SupportLine website at:

https://www.microfocus.com/support-and-services/documentation/

Manual	Description
ChangeMan ZMF Installation Guide	Installation and setup instructions for ChangeMan ZMF — including XMLSPACE setup, a prerequisite for ZMF for Eclipse.

Manual	Description
ChangeMan ZDD Server Installation Guide	Installation and setup information for the ChangeMan ZDD server on the mainframe, including customization information for the ZDDOPTS client configuration library. Settings defined in ZDDOPTS apply to ZMF for Eclipse.
ChangeMan ZMF User's Guide	How-to instructions for the functions and facilities of ChangeMan ZMF. Developer functions documented here are also available through ZMF for Eclipse.
ChangeMan ZMF Messages Guide	Explanations and recovery tips for messages issued by ZMF. These messages are passed to ZMF for Eclipse.

Using the Manuals

PDF and
Adobe Reader

The documentation for ChangeMan ZMF and ZMF for Eclipse are provided in the Adobe Portable Document Format (PDF). To view PDF files, use Adobe® Reader®, which is freely available for download from:

http://get.adobe.com/reader/

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

Finding
Information

The PDF documents include the following features to help you find and use information:

- **Bookmarks.** All of the manuals contain predefined bookmarks that make it easy for you to quickly jump to a specific topic. By default, the bookmarks appear to the left of each online manual.

- **Links.** Cross-reference links within a manual enable you to jump to other sections within the manual and to other manuals with a single mouse click. These links appear in blue.

- **Advanced search.** Adobe Reader includes an advanced search feature that enables you to perform full-text keyword searches across multiple PDF files in a specified directory.

Multi-Document
Full Text Search

To search across multiple PDF documents at once for a text string, perform the following steps (requires Adobe Reader 6 or higher):

1   In Adobe Reader, select **Edit | Search** (or press CTRL+F).

2   Select the drive or folder containing the PDF documents you want to search.

   a   In the search window under **Where would you like to search?**, select the **All PDF Documents in** option.

   b   Scroll down to the bottom of the pull-down menu and select **Browse for Location**.

   c   When the browse window appears, navigate to the location containing the documents you want to search and click **OK**.

3   In the text box below your search location, enter the word or phrase for which you want to search.

4   Optionally, select one or more of the additional search options, such as **Whole words only** or **Case-Sensitive**.

5 Click the **Search** button.

Typographical Conventions

The following typographical conventions are used in the 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	Highlights window, panel, 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 in data entry fields, or results that you receive on screen.
`monospace italics`	Indicates placeholders for values you specify; for example, `filename`.
`monospace bold`	Indicates the default value in a list of mutually exclusive choices in a syntax description, such as {Y|N}. Also highlights key portions of a syntax example.
vertical line |	Indicates mutually exclusive choices in a syntax description, such as {Y|N}. Also separates menus and their associated commands. For example, select **File | Copy** means to select **Copy** from the **File** menu.

Chapter 1
Overview of ZMF for Eclipse

Features and Functions

ChangeMan® ZMF for Eclipse™ is a Java plug-in for the Eclipse™ Workbench and IBM® Rational® Developer for z Systems® (RDz) integrated development environments (IDEs). The plug-in integrates the mainframe software change management capabilities of ChangeMan ZMF directly into these workbench environments.

Benefits   Some benefits of Eclipse integration with ChangeMan ZMF are the following:

- Tool availability — including the latest GUI editors, debuggers, generators, and application programmer interface (API) toolkits for Java, COBOL, and assembler when working with mainframe software assets.

- A unified point of access through the workbench to multiple ChangeMan ZMF repositories. The Dimensions for Eclipse plug-in adds unified access through the Serena Explorer view to Dimensions CM resources as well.

- Full integration with the robust version control, build management, system test, and multisystem deployment capabilities of a software change control system — ChangeMan ZMF — that is optimized for enterprise-scale projects.

- Mainframe levels of security and disaster protection for the software repository.

Supported ZMF Functions   Nearly all the developer functions of ChangeMan ZMF are supported by ZMF for Eclipse. In addition, many ZMF tasks are performed automatically "under the covers" as you work with development resources in the workbench. For example, you can:

- Perform development lifecycle tasks from the desktop — including checkouts from baseline, checkin to a change package, component builds, package freeze, package audit, package promote/demote, package approvals, and more.

- Work with controlled software assets and personal development libraries in the native z/OS file system or the z/OS Unix Hierarchical File System (HFS).

- Perform mass downloads with automated mapping of library types to directories, automated checkout and lock when the resource is changed, library synchronization against multiple repository resources, and mass checkins with automatic package creation when working with desktop development libraries in the workspace.

- Submit predefined jobs or ZMF XML Services data streams to the mainframe for execution.

- Access the JES subsystem to review mainframe job output.

Components

Workbench Plug-in   ZMF for Eclipse consists of a Java plug-in that is compatible with both the open-source Eclipse Workbench and IBM's mainframe-oriented Rational Developer for z Systems, which is built on Eclipse technology. The plug-in is installed in the workbench environment.

Web Application Server   The ZMF for Eclipse plug-in invokes the ChangeMan ZMF Web Services to manage function-specific connectivity to ChangeMan ZMF. The ZMF Web Services reside on a Web application server that executes either on the development system where the workbench resides or at some network location accessible to the client. ZMF for Eclipse works with Web application servers that support the Apache Axis SOAP library.

Software
Repository

The Web application server and ZMF Web Services use TCP/IP to connect to the ChangeMan ZMF software repository on the mainframe. ChangeMan ZMF manages software assets to meet the demanding change control requirements of enterprises and organizations running large-scale, mission-critical applications.

Usage Scenarios

Because it is built on cross-platform technology such as Java and Web Services, ZMF for Eclipse has the flexibility to support a wide range of use cases and installation scenarios. The following are just two examples.

Enterprise e-Commerce Development in Java

The Problem

The sheer scale and complexity of enterprise e-commerce applications require robust build management and cross-platform software deployment capabilities at the level of the software repository. Moreover, e-commerce applications in industries such as banking, financial services, and insurance are highly regulated, with specific compliance requirements for software change control. ChangeMan ZMF has always satisfied these requirements for enterprise software repositories.

However, traditional enterprise programming languages are not ideally suited to the development of e-commerce applications. Java has established itself as the preferred language for large-scale e-commerce development. In addition, with so many prebuilt software development toolkits (SDKs), software libraries, code generators, debuggers, test tools, automated documentation tools, and other development tools available for numerous operating system platforms, Java is a highly productive language for cross-platform and Web-enabled application development. Mainframe e-commerce applications can be developed more quickly and cost-effectively by deploying Java components wherever appropriate.

The Solution

For these applications, ChangeMan ZMF and ZMF for Eclipse support the z/OS Unix Hierarchical File System (HFS) for baseline, staging, promotion, and production libraries. HFS libraries provide native support for long file names and Java classpaths.

Further Java productivity enhancements in ZMF for Eclipse includes native integration of ChangeMan ZMF functions with the Java perspective of the workbench through the "Team" contextual menu. Eclipse integration is important because the Java developer community has largely adopted the open-source Eclipse Workbench IDE as its development platform of choice.

The Use Case

The diagram below illustrates a usage scenario in which Java code for enterprise e-commerce is developed in a desktop workspace managed by the open-source Eclipse Workbench on Windows. Open-source development toolkits, code libraries, and plug-ins for Java can all be leveraged within the workbench, and unit testing occurs against the Java Virtual Machine running on the desktop.

For ZMF Web Services connectivity to ChangeMan ZMF, Apache Tomcat is chosen as the Web application server.

Enterprise
e-Commerce
Development
in Java
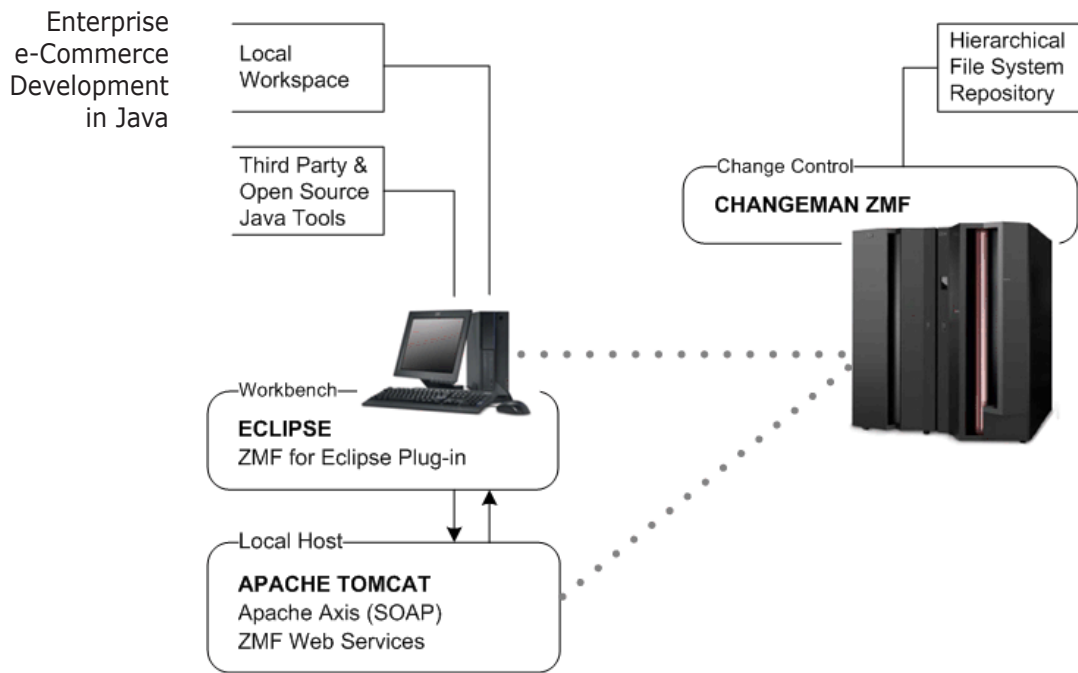
Using ZMF Web Services under the covers, the new, one-click application download feature of ZMF for Eclipse copies an entire ZMF application to the desktop at once. When a downloaded component is changed, ZMF for Eclipse automatically checks it out of baseline, stages it to development, and locks it in the ZMF change package automatically. The ZMF for Eclipse mass checkin feature uploads the entire application and checks in all components automatically to a newly created change package.

Mapping between ZMF libraries and application file types and directories on the workbench is automated. Library synchronization is supported by ZMF for Eclipse between local workspace projects and ZMF baseline or staging libraries. Side-by-side difference comparison and merge are supported in the workbench.

System testing and deployment into production are managed by ChangeMan ZMF.

Enterprise COBOL Application Maintenance

COBOL
Development
Requirements

For the maintenance of enterprise COBOL applications, including applications built over the DB2 relational database, IBM's Rational Developer for z Systems (RDz) workbench provides a powerful and wide-ranging set of development, debugging, and testing tools with GUI ease-of-use. Built on an Eclipse base, RDz's development tools complement the robust, enterprise-scale software change control, build management, and deployment features of ChangeMan ZMF.

The Use Case

In this use case, COBOL code, JCL, and associated DB2 plans and artifacts are checked out of a ZMF repository baseline into an RDz development project. Each RDz project maps to a personal development library in the native z/OS file system on the mainframe. Keeping development libraries on the mainframe under RDz project management maintains control of corporate software assets. The diagram below illustrates the basic features of this usage scenario.

For ZMF Web Services connectivity to ChangeMan ZMF, IBM WebSphere Application Server (WAS) is chosen as the Web application server.

Enterprise COBOL Application Maintenance
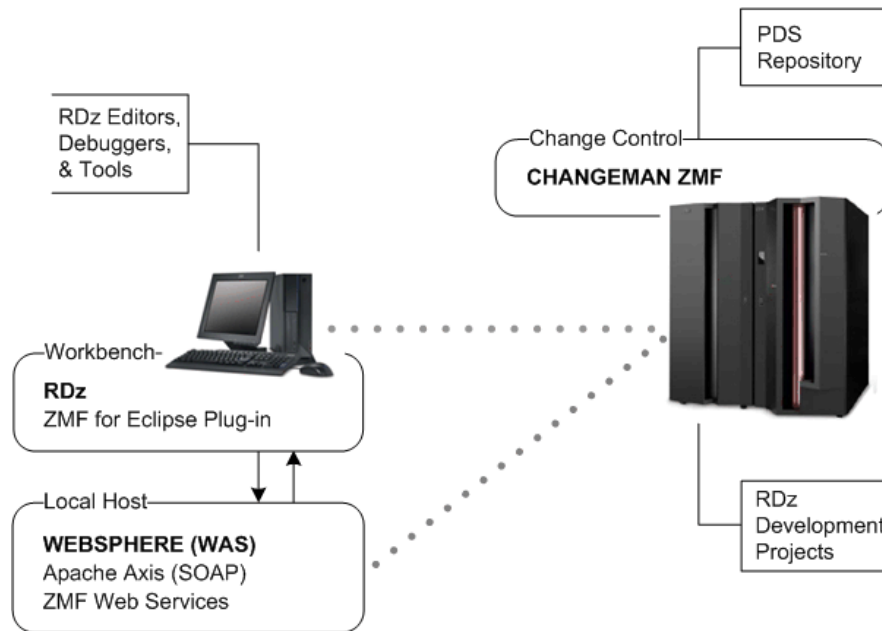
The Serena perspective and the Serena Explorer view provide a unified and configurable view of software resources in all personal development libraries on the mainframe, as well as baseline, staging, and promotion libraries in the ZMF repository. Compressed output listings can be read on the desktop in the Serena perspective thanks to a Serena-provided editor installed with the plug-in. The z/OS Explorer view of RDz provides access to project resources. Job output in the JES job queue can be accessed in either view.

Plug-in Comparison: Eclipse Versus RDz

ZMF for Eclipse plug-in behavior varies somewhat with the IDE in which it is installed. The following table compares plug-in behavior in the open-source Eclipse Workbench versus Rational Developer for z Systems.

	Eclipse Workbench	RDz
NAVIGATION	- Serena Explorer - Package Explorer (Java) - Remote System Explorer	- Serena Explorer - Package Explorer (Java) - z/OS Projects Explorer (WebSphere)
PROJECT MANAGEMENT	Eclipse projects on desktop	- Eclipse projects on desktop - RDz projects on mainframe
CHECK-OUT LOCATION	- Mainframe development libraries - ChangeMan ZMF change package	- Mainframe development libraries - ChangeMan ZMF change package - RDz projects (mainframe libraries)
LOCAL WORKSPACE	- Download z/OS data sets - Download HFS directories - Download of ZMF applications - Mass checkin to change package - Checkout from baseline - Checkout from promotion	- Download z/OS data sets - Download HFS directories - Download of ZMF applications - Mass checkin to change package - Checkout from baseline - Checkout from promotion

	Eclipse Workbench	RDz
z/OS DATA SET ACCESS	■ Create ■ Delete ■ Browse ■ Edit ■ Download	■ Create ■ Delete ■ Browse ■ Edit ■ Download
z/OS UNIX HFS OBJECT ACCESS	■ Create ■ Delete ■ Browse ■ Edit ■ Download	■ Create ■ Delete ■ Browse ■ Edit ■ Download
JES JOB ACCESS	■ Submit jobs ■ Cancel jobs ■ Display job output	■ Submit jobs ■ Cancel jobs ■ Display job output
XML SERVICES	■ Submit XML requests to ZMF ■ View service replies in editor	■ Submit XML requests to ZMF ■ View service replies in editor
JAVA INTEGRATION	■ Team menu of Java perspective ■ z/OS Unix HFS files ■ Open-source Java toolkits	■ Team menu of Java perspective ■ z/OS Unix HFS files ■ Open-source Java toolkits
COBOL SUPPORT		■ RDz development & test tools

Chapter 2
Installing ZMF for Eclipse

Installation Prerequisites

> **IMPORTANT!** The Readme file contains the latest information about supported software versions and other prerequisites for installation of the ZMF for Eclipse plug-in. Information in the Readme supersedes the guidance given in this document.

z/OS Server Requirements

Before you install ZMF4ECL, verify that requirements in the checklists below are met. Additional information on ZMF started task configuration can be found in the *ChangeMan ZMF Installation Guide*. The *ChangeMan ZMF Administrator's Guide* addresses parameter settings.

Licensing
- ❏ ZMF for Eclipse is a separately licensed option of ChangeMan ZMF. Each ZMF site that you plan to access from ZMF for Eclipse requires application of the appropriate license key to enable access. Contact Serena Customer Support to obtain a license.

Connectivity
- ❏ TCP/IP connectivity must be established between each instance of ChangeMan ZMF to be accessed and:
  - the Web application server where the ZMF Web Services reside
  - the workbench IDE (Eclipse or RDz) where the ZMF4ECL plug-in will reside
- ❏ To ensure successful operation of the ChangeMan ZMF for Eclipse interface, the XCH and ZMF ports both need to be open for access by both the Web application server and the workbench IDE.

ChangeMan ZMF Configuration
- ❏ A supported version of ChangeMan ZMF must be installed on each z/OS system that is accessed with ZMF for Eclipse 8.1.2. Refer to the Platform Matrix spreadsheet associated with the ZMF for Eclipse 8.1.2 product on the product download area of the Serena Support web site for a list of supported versions of ChangeMan ZMF. Also look at the latest Product News - at http://support.serena.com/ProductNews/prdNewsInfo.aspx?news=CHANGEMANZMF and observe the details for ZMF4ECL on the OEM Software Compatibility link on that page.
- ❏ Each ChangeMan ZMF started task that will be accessed by ZMF for Eclipse should be configured to support the following features:
  - XML Services and Web Services access to ZMF over a network
  - Unix System Services (USS) access and data areas (if you plan to work with Java)
  - Hierarchical File System (HFS) library type support (if you plan to work with Java)

Security
- ❏ The started task ID for the ChangeMan ZMF started task must be granted UPDATE access in your mainframe security system for all z/OS libraries that will be accessed by the plug-in — including all personal development libraries.

> **NOTE** If you do not know the started task ID for the ChangeMan ZMF started task, you can find it in the JESMSGLG output or ask your systems programmer.

User Permissions ❑ Each workbench user who accesses ChangeMan ZMF on the host must have a TSO user ID defined to the mainframe security system.

> **NOTE** ZMF for Eclipse is compatible with the RACF®, CA-ACF2®, and CA-Top Secret® security systems on the mainframe.

❑ Each workbench user who accesses ChangeMan ZMF must be given appropriate developer permissions by your ZMF administrator. Administrator permissions are not required.

Web Application Server Requirements

zmfws.war Web Archive File
A web application server is required to run the ZMF Web Services. This software manages interactions between ZMF4ECL on the PC and ChangeMan ZMF on the host. The code is distributed as zmfws.war and can be renamed to allow multiple versions if desired. The version desired can be set in the preferences for ZMF4ECL.

Axis SOAP Library ❑ A web application server that supports Version 1.4 of the Apache Axis SOAP library is required.

> **NOTE** Axis Version 1.4 is not the same as Axis2 Version 1.4.
>
> The appropriate Axis library is supplied by Serena with the ZMF Web Services API and need not be installed separately.

Web Application Server Software
Servers that support the Axis SOAP library include Adobe Cold Fusion, Apache Tomcat, IBM WebSphere Application Server (WAS), Oracle WebLogic, Red Hat JBoss, Sun Microsystems Java System Application Server (JSAS), and many others.

Apache Tomcat can be downloaded at no charge from the Apache Software Foundation at

http://tomcat.apache.org

Refer to the platform matrix on support.serena.com for a list of subsystems that are supported.

A trial version of WAS Express can be downloaded at no charge from IBM at

http://www-01.ibm.com/software/webservers/appserv/express/

> **IMPORTANT!** ZMF4ECL has not been tested with all the available web application servers that support Axis. It is possible that the plug-in will not work correctly with some servers.

Java ❑ A Java Runtime Environment is required by the ZMF Web Services software. It must be installed on the same machine where the web application server resides.

JRE can be downloaded at no charge from Oracle (formerly Sun Microsystems) at

http://java.com/en/download/manual.jsp.

Server Site ❑ The Web application server may be installed locally on the same PC where the workbench IDE and ZMF4ECL plug-in reside, or it may be accessed over a network.

HTTP Server
Support

❑ An HTTP transaction server must accessible to listen for SOAP requests issued by the web application server. As a general rule, the HTTP server should be installed on the same machine as the web application server. Only one HTTP server at a time may be listening for SOAP requests on that machine.

For demonstration purposes, this manual assumes that Tomcat or WebSphere Application Server (WAS) will be configured as the web server.

Workbench Requirements

Workbench
IDEs

❑ Either of the following IDE products are supported by the ZMF for Eclipse plug-in:

- *Eclipse IDE*

- *IBM Rational Developer for z Systems*

Refer to the Platform Matrix on the product documentation area of the support web site at

http://support.serena.com/

for the required versions of these two products.

The Eclipse Workbench can be downloaded at no charge from the Eclipse Foundation at

http://www.eclipse.org/downloads/

Platform

❑ Any operating system supported by the workbench is also supported by the plug-in software.

❑ 2 GB of RAM or higher are strongly recommended for the workbench.

Java

❑ A version of the Java Development Kit (JDK) compatible with your workbench is required by the plug-in and must be installed on the same machine as the workbench.

JDK can be downloaded at no charge from Oracle (formerly Sun Microsystems) at

http://java.sun.com/javaee/sdk/index.jsp

NOTE It is beyond the scope of this document to describe how to install and configure the prerequisite software mentioned here or in the Readme. In addition, the required software may have prerequisites of its own that are not listed. Refer to the documentation supplied with the prerequisite software for installation and configuration information.

Installing the ZMF Web Services

ZMF Web Services
Requirements

ZMF Web Services for ChangeMan ZMF must be installed as a web application on a web application server *prior* to using the ZMF for Eclipse plug-in on the IDE.

The ZMF Web Services stored in the zmfws.war file, must be upgraded to the 8.1.2 version prior to using ChangeMan ZMF for Eclipse 8.1.2.

The ChangeMan ZMF Web Services must be downloaded separately from ChangeMan ZMF for Eclipse on the Serena Support web site. Please read the current Readme file for the latest information.

Downloading the Web Services

`zmfws.war`
Web Archive

ZMF Web Services is distributed as a set of Java classes in a WAR (or Web ARchive) file called `zmfws.war`. This file must be downloaded from the Serena Support web site.

To download the ZMF Web Services, perform the following steps.

1 Logon to the Serena Support web site at
   http://www.serena.com/support

2 Download the most recent version of the `zmfws.war` library.

   a On the Case Manager page, click the **My Downloads** tab.

   b In the **Please Select Product** field, select **ChangeMan ZMF Client Pack** from the pull-down list.

   c A table will list the releases for which you have a license. In the table entry for the most recent version, find the **Available Operating Systems** column and click the **z/OS** link.

   d A table listing downloadable resources displays. Find the table entry for the ChangeMan ZMF Web Services file (`zmfws.zip`) and click **Download**.

3 Extract the `zmfws.war` file from the downloaded `zmfws.zip` archive.

Installing the WAR as a Web Application

Installation
Examples

ZMF Web Services for ChangeMan ZMF must be installed as a web application on a web application server. Step-by-step installation examples are given below for the following servers:

- Apache Tomcat 8.0.36

- IBM WebSphere Application Server (WAS) Express 8.5.5.9

> **NOTE** Web application installation steps vary by web application server. It is not possible to document all variations here. For more information about installing web applications on an application server, consult the documentation for your web application server.

Installing Web Services on Apache Tomcat

Tomcat is a simple, reliable, open-source web application server from the Apache Software Foundation. Because Tomcat does not build in much concurrency or session management support in the base product, the installation procedure for web applications is simple and direct.

The following installation instructions apply to Tomcat 8.0. A similar procedure may be followed with many other open-source web application servers.

Uninstalling Old Versions of Web Services

Uninstall
`zmfws.war`

Before installing the new ZMF Web Services, remove any prior versions from the web application server.

1   Stop the web application service.

  a   In Windows, go to **Start | Settings | Control Panel | Administrative Tools | Services**.

  b   From the list of services shown, select the entry for Apache Tomcat and click **Stop**.

2   Delete the `zmfws` folder and the `zmfws.war` file, if they exist, from the `webapps` directory managed by the web application server. For Apache Tomcat, a typical location is

```
C:\Program Files\Apache Software Foundation\
Tomcat 8.0\webapps
```

3   Do *not* restart the web application service at this time.

Installing New ZMF Web Services

Installing the new ZMF Web Services is a matter of copying it to the correct location and restarting the web application server.

1   Copy the `zmfws.war` file to the `webapps` directory in the web application server software directory. For example:

```
C:\Program Files\Apache Software Foundation\
Tomcat 8.0\webapps
```

2   Restart the web application server.

  a   In Windows, go to **Start | Settings | Control Panel | Administrative Tools | Services**.

  b   From the list of services, select the entry `Apache Tomcat`, then click **Start** or **Restart**. The server automatically expands the WAR file and installs the ZMF Web Services at startup.

Verifying Successful Installation

Verify
`zmfws.war`

To verify that the ZMF Web Services installed successfully, perform the following steps.

1   In your Web browser, browse to the home page for Tomcat. A typical location is

`http://RM8553:8080`

where RM8553 is the server name.

2   In the left navigation column of the Tomcat home page, in the **Administration** box, click the **Tomcat Manager** link.

3   Log on with the administrator user ID and password for Tomcat.

4   When the **Tomcat Web Application Manager** page displays (see next page), scroll to the bottom of the **Applications** table. Verify that the ZMF Web Services shows the following values:

- **Path**   — `/zmfws`

- **Display Name**— `ZMF Web Services`

- **Running**   — `true`

If the value in the **Running** column is `false`, go to the **Commands** column of the table and click **Start**.

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/zmfws	None specified	ZMF Web Services	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

As an alternative, type the following text into a browser to verify successful installation of the ZMF Web Services:

`http://server:port/zmfws/services/Version`

where:

- *server* is the name of the Web application server.

- *port* is the port to which it is connected.

For example:

`http://st6019:8080/zmfws/services/Version`

If successful, information about the currently installed version of the ZMF Web Services will be displayed. For example:

`ZMF Web Services Version 812 - Build 229 - created 9/2/2016`

Installing Web Services on WebSphere Application Server

IBM WebSphere Application Server (WAS) is a high-end web application server that builds in support for many levels of concurrency and virtualization, session management, security, single sign-on, and more in the base product. With extra function comes administrative complexity, however, so installing new web applications on WAS is not simple.

WebSphere Application Server Express addresses this complexity by packaging the most commonly co-installed WebSphere programs together — for example, IBM HTTP Server and IBM Rational Developer for z Systems are both included. WAS Express also provides a web application installation "Fast Path".

The following installation instructions apply to WebSphere Application Server Express 8.5.5.9.

Uninstalling Old Versions

Uninstall
`zmfws.war`

Before installing the new ZMF Web Services, unless you want to run multiple versions of web services, remove the prior versions from the web application server.

1   Open the administrator console for the WAS server where the old ZMF Web Services resides.

   a   In Windows, go to **Start | All Programs | IBM Websphere | IBM WebSphere Application Server V8.5 | Profiles | *app-server-name* | Administrative console**.

   b   Log on to the console with an administrator ID and password.

2   In the left navigation pane, expand the **Applications** node and go to **Application Types | WebSphere enterprise applications**.

3   From the **Enterprise Applications** panel, stop the ZMF Web Services application.

   a   Click the checkbox beside the entry for `zmfws_war`.

   b   In the row of control buttons above the list, click **Stop**.

Cell=RM8553Node01Cell, Profile=AppSrv01

Enterprise Applications

Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

☐ Preferences

| Start | Stop | Install | Uninstall | Update | Rollout Update | Remove File | Export | Export DDL | Export File |

Select	Name ↕	Application Status ↻
	You can administer the following resources:	
☐	DefaultApplication	➡
☐	ivtApp	➡
☐	query	➡
☐	zmfws_war	✖

Total 4

4 Uninstall the old ZMF Web Services application by selecting zmfws.war and clicking **Uninstall**.

5 Delete the `zmfws_war.ear` folder from the web application server software directory. A typical location is

```
C:\ProgramFiles\IBM\WebSphere\AppServer\profiles\
app-server-name\installedApps\node\zmfws_war.ear
```

Installing New Web Services

Install a New
`zmfws.war`
WebSphere Application Server (WAS) provides an installation wizard to assist with installation and configuration of the new ZMF Web Services.

1 Open the administrator console for the WAS server where the ZMF Web Services will reside.

a In Windows, go to **Start | All Programs | IBM Websphere | IBM WebSphere Application Server V8.5 | Profiles | *app-server-name* | Administrative console**.

b Log on to the console with an administrator ID and password.

2 In the left navigation pane, expand the **Applications** node and click **New Application**.

3 In the **New Application** page to the right, click **New Enterprise Application**.
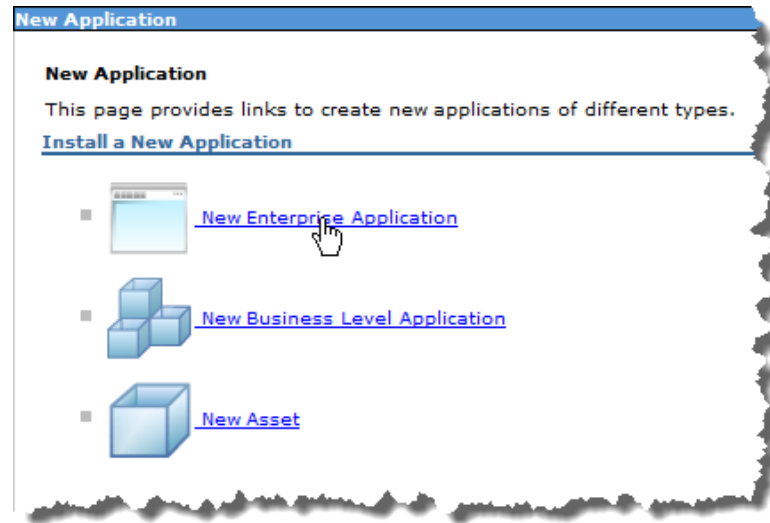
4 When prompted for the location of a **WAR module**, click **Local file system** and browse to the location where you unloaded `zmfws.war`. When selected, click Next.
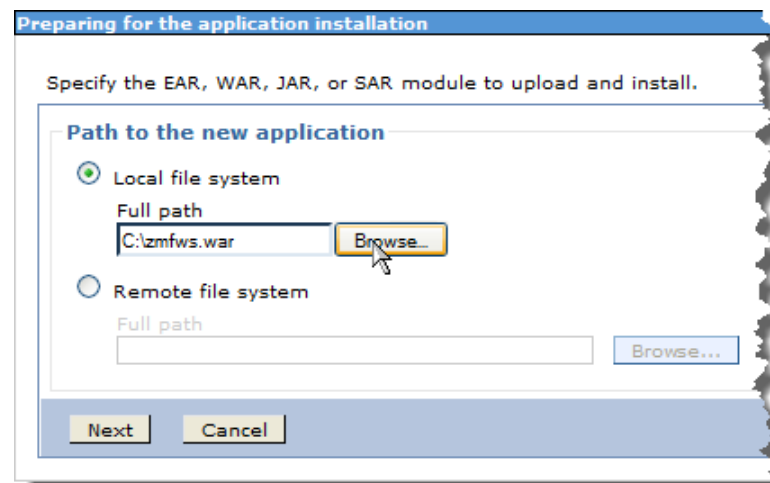
5   You will be asked **How do you want to install the application?** Choose the **Fast Path** installation method with default bindings and mappings. (These settings can be changed later if desired.)

    a   Click the radio button next to **Fast Path**.

    b   Expand the node **Choose to generate default bindings and mappings.**

    c   Check **Generate Default Bindings**.

    d   Check **Use default virtual host name**.

    e   Click **Next**.

6   In the **Select installations options** page click **Next** to accept defaults.

7   The **Map modules to servers** page asks you to associate particular web applications with particular WAS application servers (see below). Available servers are listed under **Clusters and servers**. Available applications are shown in the table below the server list.

In the simplest case, only one web application server will be defined. However, several web applications may be available to run on it.

    a   In the table of available web applications, check the box beside the ZMF Web Services module. Uncheck all other boxes.

    b   In the list of **Clusters and servers**, find the web application server you want to use to execute the ZMF Web Services. Select that server and click **Apply**.
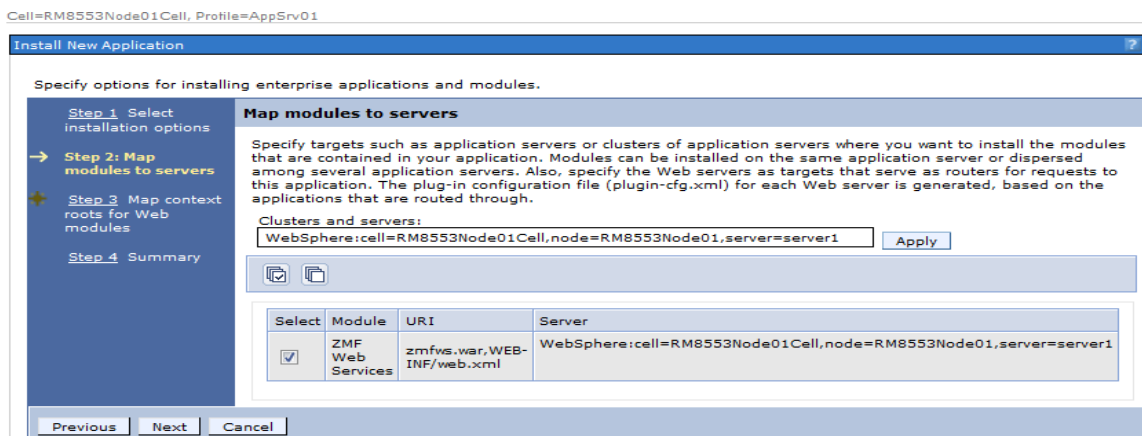
    c Click **Next**.

8 The **Map context roots** page lets you to define a custom root context (that is, a path to a directory location or folder) for execution of the ZMF Web Services. Specify the following in the Context Root field following the / character:

    zmfws
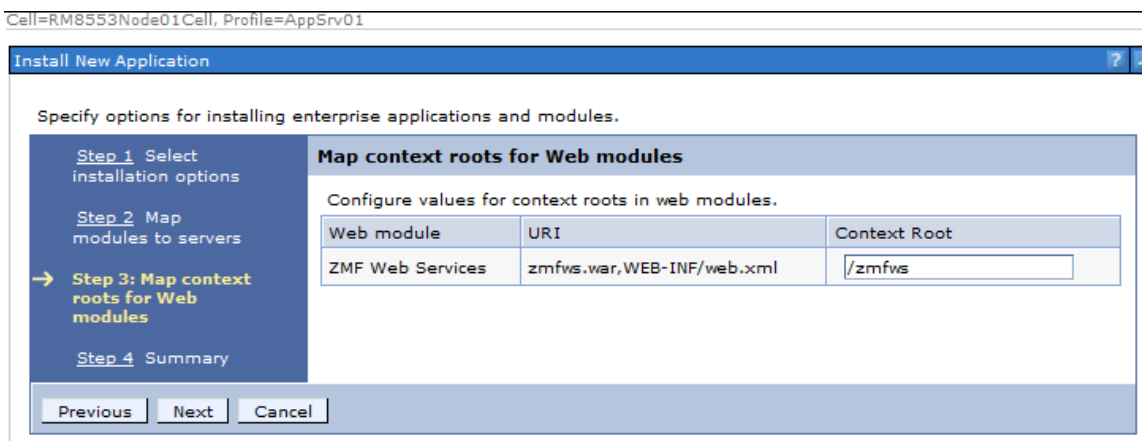
9 Click **Next.**

10 Review the **Summary** of the options chosen for the install (see below). Verify that the ZMF Web Services shows the following values:

- **Application name** — `zmfws_war`
- **Validate Schema** — No
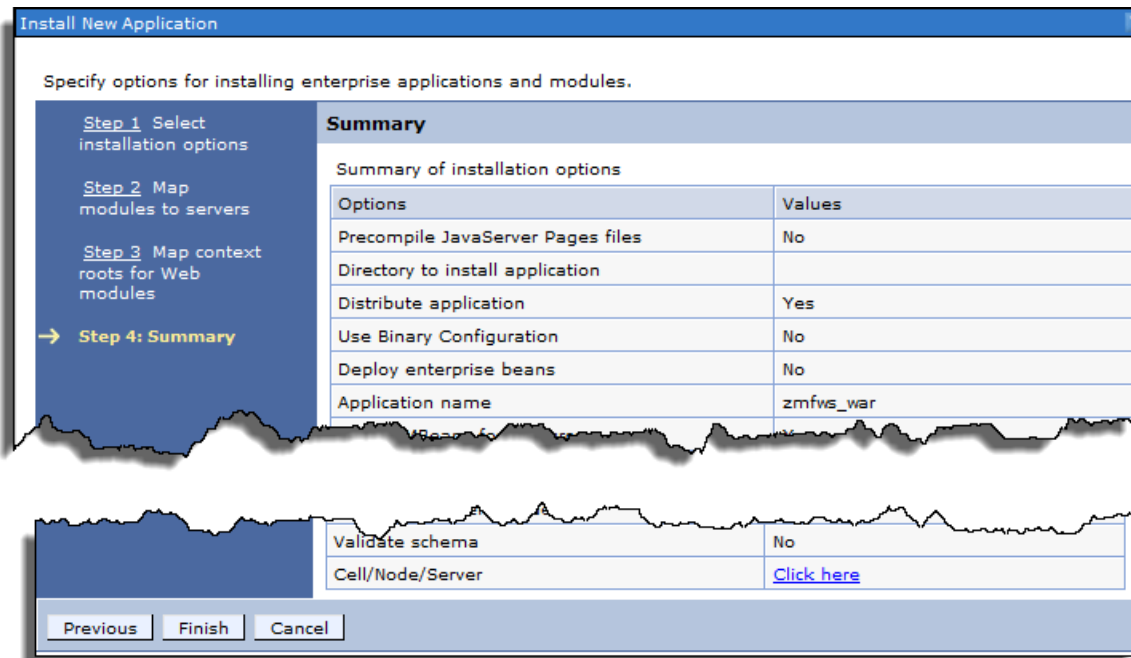
Click **Previous** to go back to a prior screen if you want to change a value.

Click **Finish** to generate the ZMF Web Services application as an Enterprise Application Archive (EAR).

11 A scrolling message log displays while the application is generated. Look for the following message at the bottom of the log:

`Application zmfws_war installed successfully.`

This message indicates that the ZMF Web Services binaries and bindings were created successfully in a *local, working* directory. The application configuration has not yet been saved in the master WAS directory. It cannot yet be started or managed by WAS.

12 Important hyperlinks are embedded in the message log text below the success message (see below).
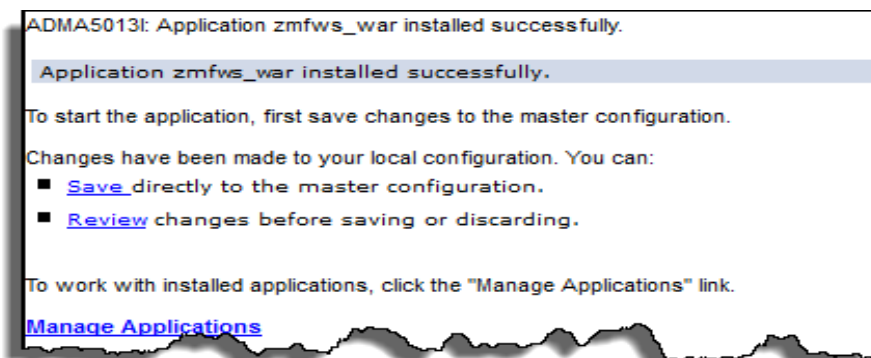
- Click **Review** to see the generated application configuration. If necessary, discard the configuration, go back and specify different settings, and regenerate the application.

■ Click **Save** to deploy the application into production under WAS.

Verifying Successful Installation

Verify Install of `zmfws.war`

To verify that the ZMF Web Services installed successfully, perform the following steps.

1 Expand the **Applications** node in the left navigation pane.

2 Under the **Applications** node, expand **Application Types** and click **WebSphere enterprise applications**.
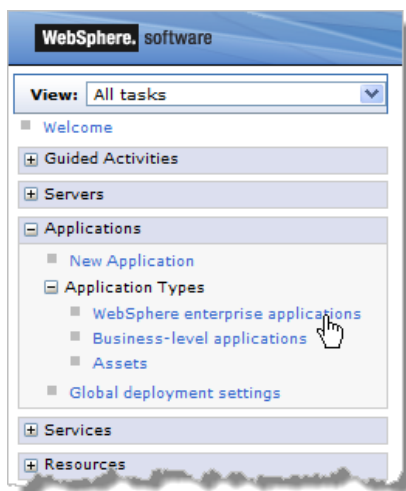
3 The page at right displays a table of all the web applications defined to WAS. The ZMF Web Services application should appear in this table with the application name `zmfws_war`.
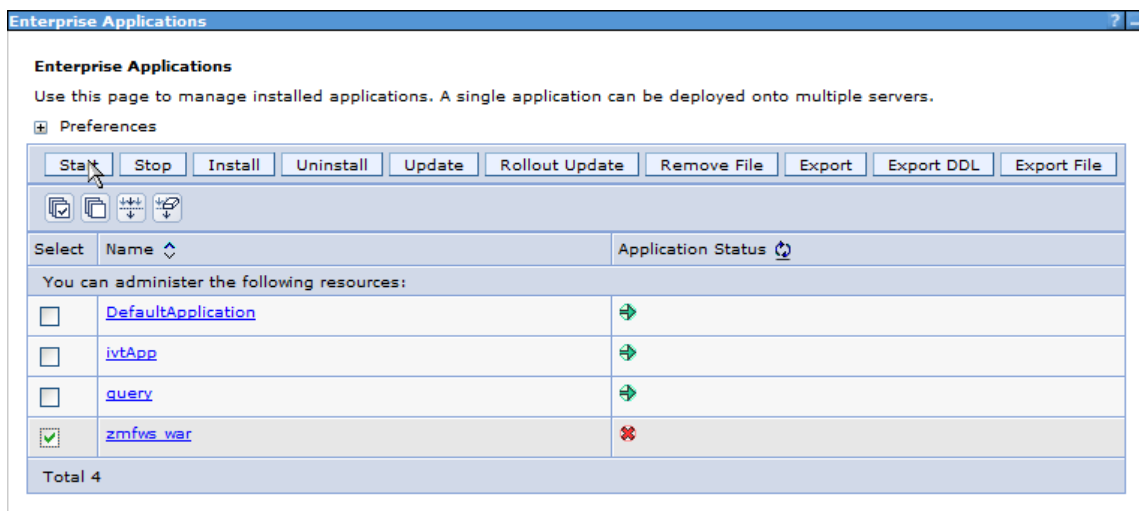
4 If `zmfws_war` has not yet been started, click the checkbox beside it to select it, then click **Start** in the row of control buttons above.

TIP To view or change configuration settings for `zmfws_war`, click the application name. A tabbed configuration page will display.

As an alternative, type the following text into a browser to verify successful installation of the ZMF Web Services:

```
http://server:port/zmfws/services/Version
```

where:

- *server* is the name of the Web application server.

- *port* is the port to which it is connected.

For example:

```
http://st6019:9080/zmfws/services/Version
```

If successful, information about the currently installed version of the ZMF Web Services will be displayed. For example:

```
ZMF Web Services Version 812 - Build 229 - created 9/2/2016
```

Installing Web Services in International Environments

Customers who install the Web Services that are distributed in the zmfws.war file for use by either ZMF for Eclipse or custom applications in non-US environments may need to take additional steps to complete the installation.

These steps are described in detail in `Solution S140158` in the knowledgebase on the Serena Support web site.

Preparing to Install the Workbench Plug-in

Uninstalling Prior Versions of the Plug-in

You must remove any previous installations of the ZMF4ECL plug-in (or its WD4ZMF predecessor) before you install the new version. This is only true if you have ZMF for Eclipse 7.1.2.01 or prior and have used the desktop installer to install the plug-in. Once you install version 7.1.2.01 or above using the new Eclipse-based installer, this step is no longer necessary, and you can use the updated install files to upgrade through the Eclipse-based installer.

> **IMPORTANT!** Be sure to uninstall prior versions of WD4ZMF as well as ZMF for Eclipse.

ZMF4ECL can be uninstalled using the installer you used to install it or from the Windows **Programs and Features** control panel in Windows. WD4ZMF can only be uninstalled this way.

Removing ZMF4ECL Using the Installer

1 Log on to the PC as a user with Windows administrator privileges.

2 Open the workbench and close the Serena perspective if it is open as follows:

  a Right-click on the **Serena perspective** tab.

  b Select **Close** from the contextual menu.

   If you don't close the perspective, the uninstall may leave a residual tab and perspective.

3 Close the workbench.

4 Start the installer program `ZmfEclipseVRM.msi`. When the **Welcome** dialog displays, click **Next**.

5 If ZMF4ECL is already installed, the installer prompts you to remove it. No other options are available.

  Click **Remove**.

Removing WD4ZMF Using the Windows Control Panel

1 Log on to the PC as a user with Windows administrator privileges.

2 Open the workbench and close the Serena perspective if it is open as follows:

  a Right-click on the **Serena perspective** tab.

  b Select **Close** from the contextual menu.

   If you don't close the perspective, the uninstall may leave a residual tab and perspective.

3 Close the workbench.

4 Use the Windows facility **Programs and Features** to uninstall any previous versions of WD4ZMF.

  a In Windows, go to **Start | Control Panel** and select **Programs and Features.**

  b From the list of installed programs, select **Serena WD4ZMF** and then click the **Uninstall** button.

  c When the InstallShield Wizard dialog box displays, select the **Remove** option and click **Next**.

  d Click **Finish** after the uninstall completes.

Removing ZMF4ECL Using the Windows Control Panel

1 Log on to the PC as a user with Windows administrator privileges.

2 Open the workbench and close the Serena perspective if it is open as follows:

  a Right-click on the **Serena perspective** tab.

  b Select **Close** from the contextual menu.

   If you don't close the perspective, the uninstall may leave a residual tab and perspective.

3 Close the workbench.

4 Use the Windows facility to **Programs and Features** to uninstall any previous versions of WD4ZMF.

 a In Windows, go to **Start | Control Panel** and select **Programs and Features.**

 b From the list of installed programs, select **ChangeMan ZMF Eclipse Integration** and then click the **Uninstall** button.

 c When the InstallShield Wizard dialog box displays, select the **Remove** option and click **Next**.

 d Click **Finish** after the uninstall completes.

Installing the Plug-in

> **IMPORTANT!** ZMF for Eclipse is a separately licensed option of ChangeMan ZMF. It will not display on the download site if you do not have a license for the product.

> **IMPORTANT!** Ensure that the Web Server has been updated to use the latest ZMF Web Services version (zmfws.war file) before you install or update the plug-in. See "Installing the ZMF Web Services" on page 22 for details.

You install ChangeMan ZMF for Eclipse with an Eclipse-based installer. You have two choices for installing the ChangeMan ZMF for Eclipse 8.1.2 plug-in:

- You can download the installation files from the Serena Support web site. If you choose this method, skip the rest of this section and follow the instructions in "Downloading the Plug-in from the Serena Support website" on page 36 and "Installing Using the Workbench" on page 37.

- You can invoke the Eclipse Installer from the Eclipse workbench, and point the installer to the one of the following URLs, and depending on the version of RDz that you are using, you can select the appropriate items to install. (Turn off 'Group Items by Category' in Eclipse and you will see two - ChangeMan ZMF Eclipse plugin and ChangeMan ZMF RDZ integration). You will need both if on RDz.

- On Eclipse only you can use either URL:

- For RDz 9.5 and/or Eclipse copy these URLs in blue:

http://help.serena.com/eclipseplugin/ZMF4ECLUpdateSite/RDz 9.5

- For RDz 9.1.x and below and/or Eclipse:

http://help.serena.com/eclipseplugin/ZMF4ECLUpdateSite/RDz 9.1 and below

- If Eclipse can't find the site, possibly it has inserted one or more %20 for the space character(s), then select Edit to remove any %20 and replace with a space.

> **NOTE** You can access this URL only from within the Eclipse Installer from the Eclipse workbench; you cannot access this URL with a browser. If you do try and use a browser, you will see a message like this:
>
> **Serena ChangeMan ZMF for Eclipse**
> This URL cannot be accessed with a browser, but only from within the Eclipse installer. Please check the ChangeMan ZMF for Eclipse Readme for more information regarding the URL.

If you choose this installation method, that's all there is to it. Skip the instructions in "Downloading the Plug-in from the Serena Support website" on page 36 and "Installing Using the Workbench" on page 37.

(You download and install the Web Services as described in "Installing Web Services on Apache Tomcat" on page 23 or "Installing Web Services on WebSphere Application Server" on page 26.)

Download the documentation pertaining to the Web Services and to ChangeMan ZMF for Eclipse from the Serena Support web site at h`ttp://support.serena.com`.

Downloading the Plug-in from the Serena Support website

You can download the ZMF for Eclipse installation files from the Serena Support website and post them locally or on your company's network. A user ID and password is required to log on to the Serena Support website.

1 From the `support.serena.com` case manager page, click the **My Downloads** tab.

2 From the **Please Select Product** pull-down menu on the downloads page, select `ChangeMan ZMF Client Pack`. A list of the ZMF client pack versions for which you are licensed displays.

3 Under **Available Operating Systems**, click **Z/OS** for the version of Eclipse for ZMF that you want to integrate with your workbench. A list of installer files displays.

4 In the table entry for the desired RDz version (9.5 or 9.1 and below) **ChangeMan ZMF for Eclipse 8.1.2 Installer**, click **Download**.

5 Click **Save** and select the directory location in which to save the file.

6 Extract the installer files from the zipped file. The installer files appear in the `ZMF4ECL Update Site` subdirectory.

The installation can be done using files residing on a local or network file directory. The installer performs the following tasks:

■ Installs the plug-ins for the workbench you are using (Eclipse or RDz), including the Eclipse for ZMF Help plug-in.

■ Upgrades or reverts the plug-ins to a selected version of the Eclipse for ZMF plug-in.

Installing Using the Workbench

After you download your ZMF4ECL installer files to a local file directory, use your workbench to install using the Local installation option.

> **IMPORTANT!** The exact steps to install local files from the workbench vary from one version of Eclipse and RDz to another. For the exact steps to use for your workbench, please see your Eclipse or RDz documentation.

1 In your Eclipse or RDz workbench, select the option to install or update software. For example, **Help | Install New Software** or **Help | Software Updates**.

2 Select the option to add the site on which the install repository is located. For example, click **Add** or **Add Site**.

> **NOTE** The installation files for ZMF for Eclipse are not hosted in a Web application repository by default. However, you choose the add sites option to browse to and select installation files that reside on a local or network directory.

3 Click **Local** and browse to the `ZMF4ECL Update Site` directory that contains the downloaded and extracted install files.

4 Click **OK** to select the directory.

5 Click **OK** on the **Add Repository** dialog to accept the location.

The plug-ins that are available appear. If the list of plug-ins does not appear, make sure **Group items by category** is unchecked.

6 If using Eclipse only, select **ChangeMan ZMF Eclipse Plugin**.

If using RDZ, select both **ChangeMan ZMF Eclipse Plugin** and **ChangeMan ZMF RDZ Integration**.

7 Click **Next**.

The **Install Details** dialog appears.

8 Click **Next**.

The **Review Licenses** dialog appears.

9 Review the license text and select **I accept the terms of the license agreement**.

10 Click **Finish**.

You may receive a security warning. Click **OK** to continue with the installation, and the **Selection Needed** dialog appears asking **Do you trust these certificates?**

11 Select **Serena Software, Development; Serena Software** and click **OK**.

A message appears letting you know that you need to restart your workbench to complete the installation.

Verifying the Installation

After you restart the workbench:

1 Select **Window | Preferences**.

2 Select **ZMF/Eclipse** in the list.

The plug-in version and build number appear in the content view.

Enabling Plug-in Support in ChangeMan ZMF

Applying the License for ZMF for Eclipse

ZMF for Eclipse is a separately licensed option of ChangeMan ZMF. You must apply the appropriate license key to each ZMF server that you plan to access using the workbench plug-in. To obtain a license key, contact Customer Support.

A ChangeMan ZDD server license is required on the host to enable ZMF4ECL to work with ChangeMan ZMF.

Complete instructions for applying license keys for product options appear in the *Serena SER10TY User's Guide*. This manual is included with the documentation for ChangeMan ZMF. It may also be downloaded from the Serena Support web site.

ChangeMan ZMF 7.1 and 8.1 Considerations

The following steps are required to apply your license key for ZMF 8.1.x:

- The `SERLIC DD` statement must be updated in your ZMF JCL to activate the license for ZMF for Eclipse.

- The ZMF server (ChangeMan ZMF on the mainframe) parameter, `ECLIPSE=Y`, in the startup parameters for SERNET is obsolete in 8.1.1 and above, and in 8.1.0.01 and earlier, but it is required (down to 7.1.x). Refer to Appendix D as well as Chapter 5, ZMF 8.1 Installation Guide for details of SERNET parameters and SERNET keyword options.

ChangeMan ZMF 6.1 Considerations

No special ZMF license is required to support the ZMF4ECL option when using ZMF4ECL with ZMF 6.1.x.

IMPORTANT! ZMF 6.1.x systems should ***not*** use the `ECLIPSE=Y` parameter. If this parameter is set, it will direct SERNET to search for a license that does not exist in the ZMF 6.1.x environment.

Installing the ZDDOPTS Parameter Library

ZDDOPTS Parameter Library

The ZDDOPTS parameter library for ChangeMan ZMF resides on the mainframe and customizes the behavior of ZMF when accessed by ZMF for Eclipse. The same parameter

library is used with all Web-enabled client integration products for ChangeMan ZMF, including Serena ChangeMan ZDD. Consequently:

- One ZDDOPTS file can configure multiple clients that access the same ZMF server.

- Different ZMF servers can enforce different repository behavior with the same client.

A ZDDOPTS library must be added to your ChangeMan ZMF configuration on the z/OS host to enable ZMF support for ZMF4ECL. This must be done for each ZMF server that will be accessed from the workbench.

Creating a Default ZDDOPTS Library

To add a ZDDOPTS library to an existing ChangeMan ZMF server:

1   Allocate a PDS(E) library on the host for ZDDOPTS.

This library can have any record length. However, if you wish to use the sample ZDDOPTS configuration members provided with ZMF for Eclipse (or in the SERCOMC SAMPXML library provided with ChangeMan ZMF on the host), allocate the ZDDOPTS library with these characteristics:

```
DSN=somnode.CMNZMF.ZDDOPTS
DCB=(RECFM=VB,LRECL=255,BLKSIZE=0) SPACE=(CYL,(1,10,30))
```

2   Copy the files from sample ZDDOPTS directory provided with the ZMF for Eclipse (or the corresponding members from the SERCOMC SAMPXML library provided with ChangeMan ZMF on the host) into your new ZDDOPTS library.

3   Add the DDNAME for ZDDOPTS to the ChangeMan ZMF server JCL and code it with the data set name of the ZDDOPTS library you just allocated.

Library Mapping

Desktop Workspace Requirements

If you want to use a desktop workspace as a personal development library under ZMF for Eclipse, you must configure library mapping between mainframe library types and desktop file extensions in ZDDOPTS. This is done in the ECLIPSE member of ZDDOPTS. Without library mapping, the **Add to Workspace** option in the Serena Explorer contextual menus will be grayed out.

Configuring ZDDOPTS Parameters

See Appendix A, "ZDDOPTS Parameter Reference" for a complete description of the ZDDOPTS library members and the parameter settings for each. Customizing these parameters should be reserved for the ZMF administrator.

Managing the Plug-in Software Updates

Serena may provide a new build to address fixes and enhancements to the plug-in version 8.1.2 and above. If you installed ChangeMan ZMF for Eclipse 8.1.2 by pointing the Eclipse Installer to this URL:

```
http://help.serena.com/eclipseplugin/ZMF4ECLUpdateSite/RDz 9.5
```

or if on the older RDz:

> `http://help.serena.com/eclipseplugin/ZMF4ECLUpdateSite/RDz 9.1 and`
> `below`

1   Ensure that the Web Server has been updated to use the latest version of the ZMF Web Services (zmfws.war file) prior to using the plug-in.

2   In your Eclipse or RDz workbench, select the option to update software. For example, **Help | Check for Updates**.

The system checks for any updates and if any are found, it asks you to check the updates that you wish to install. If you check them, click **Next** then **Finish** and it will update automatically.

3   If prompted, select the Serena Software certificate as trusted and click **OK**.

A message appears letting you know that you need to restart your workbench to complete the upgrade.

4   Restart your workbench.

Verifying the Upgrade

1   Select **Window | Preferences**.

2   Select **ZMF/Eclipse** in the list.

The plug-in version and build number appear in the content view.

Uninstalling Using the Eclipse-based Installer

If you have installed ZMF for Eclipse 7.1.2.01 or higher and want to uninstall the software completely from a workstation, you can do so using the Eclipse-based installer.

> ⚠️ **CAUTION!** You cannot use the Eclipse-based installer to uninstall ZMF for Eclipse 7.1.2 or prior. If you installed your plug-in with the desktop installer, you must uninstall using the desktop installer or the Windows remove programs option. If you are using 7.1.2 or prior, see "Uninstalling Old Versions of Web Services" on page 24.

To uninstall ZMF for Eclipse 7.1.2.01 or above:

1   In your Eclipse or RDz workbench, select the option to see installation details. For example, **Help | Install New Software | What is already installed** or **Help | About Eclipse SDK | Installation Details**.

2   Select the **Installed Software** tab.

3   Select **ChangeMan ZMF for Eclipse Plugin**.

4   Click **Uninstall**.

The **Uninstall Details** form appears.

5   Click **Finish**.

A message appears letting you know that you need to restart your workbench to complete the uninstall.

6   Restart your workbench.

Verifying the Eclipse-based Uninstall

To verify the uninstall of 7.1.2.01 or above:

1   Select **Window | Preferences**.

You should no longer see **ZMF/Eclipse** in the list.

2   Select **Window | Open Perspective | Other**.

You will no longer see **Serena** in the list.

Chapter 3
Server Setup and Connectivity

Connecting to a Web Application Server

When you first bring up the workbench after installing the ZMF for Eclipse plug-in, you need to create a connection to the Web application server where you installed the ZMF Web Services. This is a prerequisite for connecting to a ChangeMan ZMF software repository.

The plug-in connects to only one Web application server at a time.

Creating or Changing a Connection

To create or change the plug-in connection to a Web application server, perform the following steps.

1 Open the Serena perspective in the workbench.

   a From the workbench menu bar, select **Window | Open Perspective | Other**.

   b When the **Open Perspective** window displays, select **Serena** and click **OK**.

2 From the workbench menu bar, select **Window | Preferences**. The **Preferences** window displays.

3 In the left pane of the **Preferences** window, expand the **ZMF/Eclipse** node and then click **Settings**.
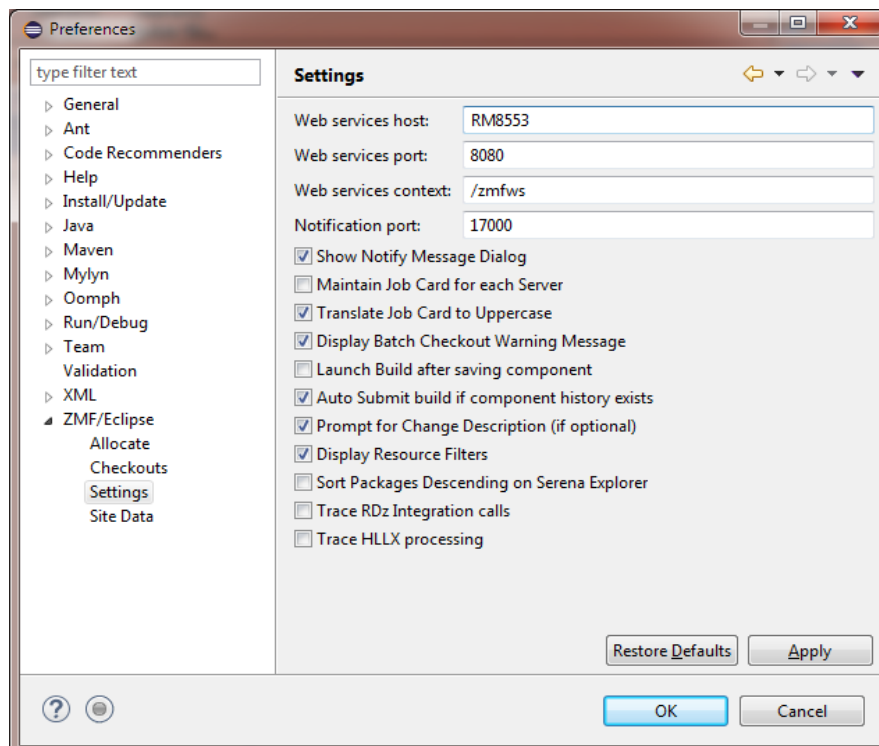
4 Enter the Web services host name and port number in the **Web services host** and **Web services port** text fields.
If you do not know these values, contact your network administrator.

NOTE By default, the plug-in assumes the Web application server is listening on port 8080 of the local server. However, the port number should identify the HTTP port that is configured for the Web application server that is being connected to.

A remote server on the network can also be used as the Web application server supporting ZMF for Eclipse.

5 Enter the location of the .war file into the Web Services context field.

6 Enter the Notification port.

7 Select or deselect preferred settings for this Web application server as follows:

- Select **Show Notify Message Dialog** (selected by default) to show each notification message as it is received from ZMF. These messages inform you of progress on work initiated in ZMF, such as build jobs submitted and promotions. You can view the messages at any time in the **ZMF Notifications** view. For information on this view, see the *ChangeMan ZMF for Eclipse User's Guide*.

- Select **Maintain Job Card for each Server** (deselected by default) if you run several different ZMF systems and need custom job cards for each server. All functions that use a job card, such as build, promote, and audit, will use the server-specific job card if this option is set. If it is not set, only one job card is used for all servers.

- Select **Translate Job Card to Uppercase** (selected by default) to convert all job cards to uppercase**.**

- Select **Display Batch Checkout Warning Message** (selected by default) to see these messages.

- Select **Launch Build after saving component** (deselected by default) to launch the build dialog after a save.

- Select **Auto Submit build if component history exists** (deselected by default) **.**

- Select **Prompt for Change Description (if optional)** (selected by default) to prompt for a change description upon checkin even if the ZMF SSV, or Staging Versions, option for a particular library type is configured as optional.

- Select **Display Resource Filters** (selected by default)**.**

- Select **Sort Packages Descending on Serena Explorer** (deselected by default)**.**

- Deselect **Trace RDz Integration calls** (deselected by default) - This should only be necessary if requested by Serena Support.

- Deselect **Trace HLLX processing** (deselected by default) **-** This should only be necessary if requested by Serena Support.

-

8 Click on **Restore Defaults** to go back to the product defaults, note this is including the four text fields at the top.

9 Click **OK** or **Apply** to save the settings.

Verifying the Connection

A direct method does not exist for testing the connectivity between the ZMF for Eclipse plug-in and a particular Web application server. However, you can verify that the host DNS name and port number for your Web application server are correct as follows:

1   Verify that the Web application service is started in Windows.

   a   Go to **Start | Control Panel | Administrative Tools | Services** and scroll to the name or your application server (for example, `Apache Tomcat` or `IBM WebSphere Application Server`).

   b   If the service is not listed as `Running` in the services panel, select it and click **Start** or **Restart**.

2   In your Web browser, type the URL you defined in the ZMF for Eclipse preferences for the Web application server. A typical location is

   `http://RM8553:8080/`

   If the server is running at this location, the home page for that server should display in the browser.

   For Apache Tomcat, that home page looks something like this:
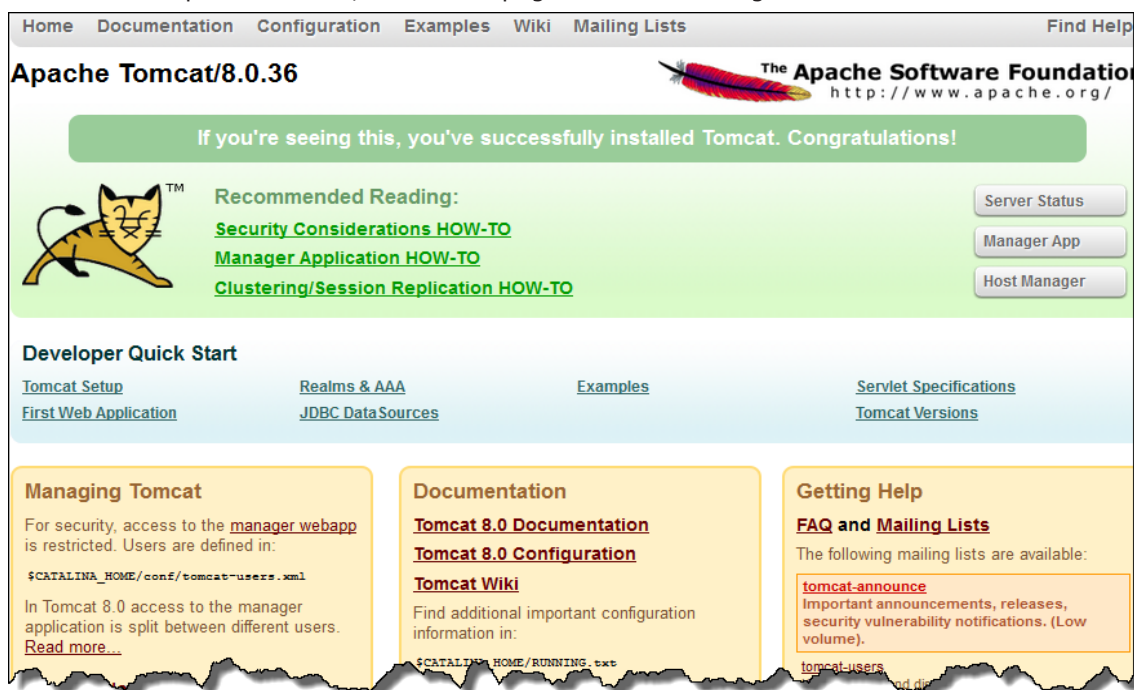
Connecting to a ChangeMan ZMF Server

For each ZMF repository that you want to work with in the workbench, you must define a connection in ZMF for Eclipse. A single workbench may connect to multiple ZMF sites or repositories concurrently.

ZMF Server Connection Prerequisites

Workbench Connectivity to the Web Application Server

Before you attempt to connect to a ZMF software repository, be sure you have a working connection between the workbench and the Web application server where the ZMF Web Services reside.

> **IMPORTANT!** Start the Web application server from the Windows **Start | Control Panel | Administrative Tools | Services** panel if it is not already started.

TCP/IP Connectivity to z/OS

A TCP/IP connection must be established between the Web application server system and the z/OS host where ChangeMan ZMF resides before you can create a ZMF connection in ZMF for Eclipse.

On the Web application server host you can use the Windows `ping` command to verify TCP/IP connectivity with z/OS.

1   Select the Windows **Start | Run** option and type the `cmd` command to request a command-line interface window. Press Enter or click **OK**.

2   When the command-line window displays, type the following command at the prompt:

    ping lpar.company.com

where

    lpar = the network name of the z/OS host or LPAR

    company.com = the domain name for the network where the
                  z/OS mainframe resides

Press Enter.

3   Verify that the `ping` command returns a response similar to the following:

```
Pinging sys1.serena.com [11.22.33.444] with 32 bytes of data:

Reply from 11.22.33.444: bytes=32 time=107ms TTL=61
Reply from 11.22.33.444: bytes=32 time=93ms TTL=61
Reply from 11.22.33.444: bytes=32 time=101ms TTL=61
Reply from 11.22.33.444: bytes=32 time=103ms TTL=61

Ping statistics for 11.22.33.444:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milliseconds:
    Minimum = 93ms, Maximum = 107ms, Average = 101ms
```

> **IMPORTANT!** If the `ping` command is unsuccessful, contact your network administrator and correct the problem before attempting to create a connection between ZMF for Eclipse and ChangeMan ZMF.

Creating a New ZMF Connection

A ZMF connection in the Serena perspective is all you need to work with mainframe resources from the workbench. However, users of Rational Developer for z Systems (RDz) will typically want to use RDz projects to manage personal development libraries on the mainframe as well. A separate RDz connection to the mainframe is required for this purpose.

See "Creating a New z/OS Connection in RDz" in the topic on "Linking Development Libraries to RDz Projects" in Chapter 4, "Development Work Area Setup", for more information about RDz connection setup.

To create a new connection to a ZMF repository in ZMF for Eclipse, perform the following steps.

1   Open the Serena perspective in the workbench.

   a   From the workbench menu bar, select **Window | Open Perspective | Other**.

   b   When the **Open Perspective** window displays, select **Serena** and click **OK**.

2   In the **Serena Explorer** navigation panel, right-click on the **ChangeMan ZMF** node to bring up its contextual menu.

3   From the contextual menu, select **Tools | New ZMF Server**.

Create a New ZMF Server Connection

4   When the **ZMF Server Properties** window displays, enter the following information:

ZMF Server Properties Window

- **Name** — Node name to display in Serena Explorer for the ZMF server site.

- **Address** — IP address or DNS name of mainframe LPAR where ZMF resides. The DNS server name takes the form

    `lpar.company.com`

    where

`lpar`	= network name of the z/OS system or LPAR
`company.com`	= domain name for the network where the z/OS LPAR resides

- **ZMF Port** — Port number for the particular ZMF instance where the desired software repository resides.

    This is the value entered as CMN=nnnn when ChangeMan ZMF was installed. You will log on to this port when you access libraries managed by ChangeMan ZMF.

- **XCH Port** — Port number for the SERNET network server that routes messages among ZMF instances and z/OS system services.

    This is the value entered as XCH=nnnn when ChangeMan ZMF was installed. You will log on to this port when you access your personal development libraries or other data sets on z/OS that are *not* managed by ChangeMan ZMF.

- **Connections Secured via TLS** — Select if you wish to use Transport Layer Security (TLS) for an encrypted connection, and it is enabled on the z/OS server. Connection to the server will fail if the TLS setting for ZMF4ECL does not match the TLS configuration on the z/OS server.

- **Overide Default Web Services Address** — Select if you wish to override the supplied defaults.

    If you do not know these values, contact your ZMF administrator.

5   Click **Finish** to create the connection in ZMF for Eclipse. Skip the remaining setup steps at this time.

> **NOTE** If you click **Next** at the **ZMF Server Properties** window, you will be asked to define a series of optional filters for viewing ZMF applications, data sets, HFS files, and jobs associated with this ZMF server. These filters can be added, changed, or deleted at any time and do not need to be configured now.

6   Verify that the name of the new ZMF repository is listed under the **ChangeMan ZMF** node when you expand it in **Serena Explorer**.

7   Repeat these steps as needed to add more ZMF sites, such as a test site or a production site, to the **Serena Explorer** view.

> **IMPORTANT!** If you are using RDz, a z/OS connection must be configured in the Remote Systems Explorer perspective of RDz for each ZMF server connection created in the Serena perspective. This is necessary to integrate ZMF checkouts with RDz projects. See your RDz documentation for more information.

Viewing or Changing Connection Properties

To display the properties of an existing connection (such as port numbers) to a ChangeMan ZMF server, perform the following steps.

1   In the **Serena Explorer** view, expand the **ChangeMan ZMF** node to show all ZMF sites with defined connections.

2   Right-click on the desired ZMF site to bring up its contextual menu, then select **Properties** to see the **ZMF Server Properties window**.
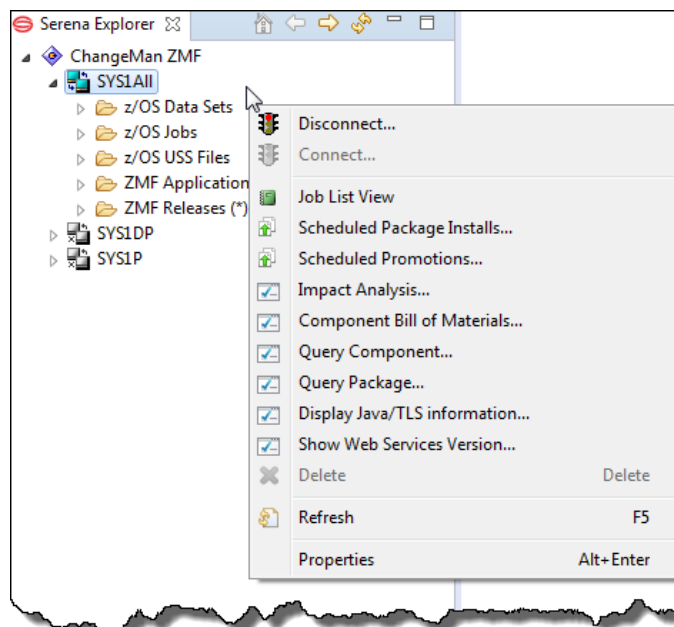
3   The **ZMF Server Properties** window displays connection properties.

Use this dialog to change connection properties if needed, as previously described in "Creating a New ZMF Connection" (Step 4 to Step 5).

Deleting a ChangeMan ZMF Connection

To delete an existing ZMF connection from ZMF for Eclipse:

1   In the **Serena Explorer** view, expand the **ChangeMan ZMF** node to show all ZMF sites with defined connections.

2   Right-click on the ZMF site you want to delete to bring up its contextual menu, then select **Delete ZMF Server**.

Exporting and Importing ZMF Connection Settings

So that you can create and deploy a common ZMF server configuration across workstations, configure one workbench with the ZMF server settings you want, export the configuration, and import the configuration into each additional workbench.

Exporting Connection Settings

To export workbench connection settings, perform the following steps.

1   Open the Serena perspective in the workbench.

 a   From the workbench menu bar, select **Window | Open Perspective | Other**.

 b   When the **Open Perspective** window displays, select **Serena** and click **OK**.

2   In the **Serena Explorer** navigation panel, right-click on the **ChangeMan ZMF** node to bring up its contextual menu.

3   From the contextual menu, select **Tools | Export Server Definitions**…

4   When the **Select Servers for Import/Export** window displays, do the following:

 a   Select the servers for which you want to export the settings.

 b   Browse to the directory where you want to save the settings.

 c   Click **OK**.

5   Upon Export Complete to *fullpath\server.xml* confirmation, click **OK** to close the dialog.

Importing Connection Settings

To import previously exported workbench connection settings, perform the following steps.

1   In the **Serena Explorer** navigation panel, right-click on the **ChangeMan ZMF** node to bring up its contextual menu.

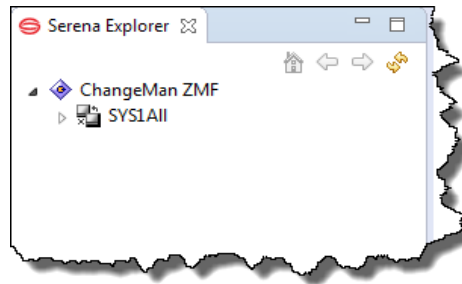**2**   From the contextual menu, select **Tools | Import Server Definitions**.

3   When the **ZMF Server Import** window displays, do the following:

    a Browse to the directory where you saved the exported settings.

    b Select and open the `server` file.

    c Select the **Replace duplicate server definitions** option if you want any existing server definitions of the same name to be replaced.

    d Click **OK**.

4 Upon confirmation, click **OK** to close the dialog.

Refreshing ZMF Connection Settings

If you update the ZDDOPTS parameter library on a ZMF server while using ZMF for Eclipse, you must disconnect and reconnect to that server for those settings to take effect.

> **NOTE** It is *not* necessary to restart the ZMF started task when you make changes to ZDDOPTS.

Logging On to a ChangeMan ZMF Server

Automatic Logon Prompts   ZMF for Eclipse automatically prompts you to log on to the selected ZMF server when you first open its node in Serena Explorer. You may also be prompted to log in again when switching between the z/OS native and z/OS Unix file systems.

Contextual Menu Logon   You can also log on from a contextual menu at any time. To log on to a ZMF server from a contextual menu:

1 Open the Serena perspective in the workbench.

    a From the workbench menu bar, select **Window | Open Perspective | Other**.

    b When the **Open Perspective** window displays, select **Serena** and click **OK**.

2 In the **Serena Explorer** navigation view, right-click on the desired ZMF site to bring up its contextual menu, and then select **Connect**.

3 The **ZMF Logon** dialog displays the name of the server you are accessing and prompts for your mainframe logon credentials.

    Type your mainframe user ID and password in the **Userid** and **Password** fields. Password phrases (as well as standard passwords) are supported.

> **NOTE** Do not enter a value in the **New Password** and **Verify New Password** fields unless you wish to change your password on the mainframe.

If you wish to save your mainframe logon credentials on your local computer for automatic reuse with this dialog check **Save Userid** or **Save Password**.

IMPORTANT! Do NOT save your mainframe logon credentials on a shared machine. Check **Save Userid** or **Save Password** only if you are the sole authorized user of your PC and access to the PC is password-protected.
The ability to save credentials can be allowed or disallowed through the corresponding ZDDOPTS field commands. See Appendix A, "COMMAND Member Parameters" on page 80.

Click **OK** to log on to the host.

4 A message dialog displays your logon status.

Logging Off from a ChangeMan ZMF Server

Logging off from a ChangeMan ZMF repository is similar to logging on.

1 In the **Serena Explorer** navigation view, right-click on the desired ChangeMan ZMF site to bring up its contextual menu.

2 Select **Disconnect**.

Chapter 4
Development Work Area Setup

z/OS Personal Development Libraries and ZMF

Work Area Types and Locations	ZMF for Eclipse requires development activities to be performed in a set of personal development libraries on the z/OS server. However, Java developers working in the workbench Java perspective can dynamically link mainframe development libraries to libraries in a desktop workspace associated with the workbench. In this way, development activities take place on the desktop but software assets remain controlled.
Projects	If your workbench client is Eclipse, you can manage development libraries in your desktop workspace using Eclipse projects. If your workbench client is RDz, z/OS development libraries can be associated with RDz projects.
Repository Services	For both clients, and regardless of personal development library location, ZMF for Eclipse ensures that versioning and other repository services are performed by ChangeMan ZMF.

Requirements and Best Practices for z/OS Personal Development Libraries

Best Practices Guidelines	The following guidelines apply to z/OS personal development libraries when used with ZMF for Eclipse:

- Development libraries should be set up on the same z/OS system where the associated ZMF repository and change packages reside.

- COBOL, PL/I, or assembly language developers should use PDS data sets in the z/OS native file system as personal development libraries on the mainframe. By default, ZMF for Eclipse assumes a naming convention for these data sets in which the High Level Qualifier (HLQ) is equal to your TSO user ID.

- Java developers should use the z/OS Unix Hierarchical File System (HFS) for personal development libraries on the mainframe. By default, ZMF for Eclipse assumes a path naming convention for these libraries that is rooted in the directory `/u/userid/`, where `userid` is your TSO user ID.

- The file system used by a personal development library (PDS or HFS) should match the file system used by the baseline libraries from which components are checked out *and* the file system used by the staging libraries associated with the change packages to which components are checked in.

- Java developers should verify with their ZMF administrator that their Java applications in ZMF have been customized to support HFS library types, and that their baseline libraries, staging libraries, promotion libraries, and production libraries are all created in advance using the z/OS Unix System Services (USS).

TIP Serena recommends that staging versions be enabled for your staging libraries when using ZMF for Eclipse. The staging versions (SSV) feature ensures that development artifacts are automatically versioned whenever you check in a component from a development library to a change package.

Creating a z/OS Development Library

PDS Libraries
In the z/OS native file system, personal development libraries are created using ISPF. These libraries may be created as PDS (Partitioned Data Set) or PDS/E (Partitioned Data Set/Extended) data sets.

HFS Libraries
In the z/OS Unix Hierarchical File System (HFS), personal development libraries can be created using whichever of the following tools you prefer:

- **ISHELL** provides an ISPF panel interface to the z/OS Unix system shell and is the interactive tool of choice for mainframe developers unfamiliar with Unix.

- **OMVS** starts the z/OS Unix system shell from the TSO command line. OMVS is the interactive tool of choice for Java developers familiar with Unix shell commands.

More information about z/OS Unix is available from IBM at
`http://www.ibm.com/support/knowledgecenter/zosbasics/`
`com.ibm.zos.zconcepts/zconc_whatiszunix.htm`

If you use the context menu for z/OS Data Sets in Eclipse you can create new datasets, also you can create new datasets using the Remote Systems perspective of RDz.

Viewing Personal Development Libraries

Multiple personal development libraries on multiple z/OS systems can be accessed from the workbench.

To view your z/OS personal development libraries from the workbench:

1   Open the Serena perspective in the workbench.

2   In the **Serena Explorer** view, select the ZMF server where your personal libraries reside and log on.

3   Expand the selected ZMF server node in the **Serena Explorer** navigation tree.
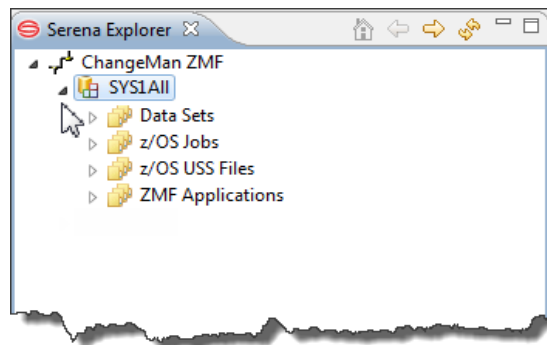
4   Expand the node associated with the type of file system — native z/OS PDS or z/OS Unix System Services (USS) HFS — where your development libraries reside.

- To access native z/OS PDS libraries, expand the **Data Sets** node.

- To access Unix-style HFS libraries, expand the **z/OS USS Files** node.

5   Expand the default filter node, or another relevant viewing filter node that you may have created. Your personal development libraries are displayed as folders under the relevant object type and filter in the **Serena Explorer** navigation tree.

6 Expand a library folder to show its members or components. These display as terminal nodes in the tree.

> **TIP** If the list of z/OS libraries shown is too large to work with or is incomplete, you can customize your Serena Explorer viewing filters at any time. By default, the filters show all libraries with a top-level data set node or directory name that matches your TSO user ID.
>
> Refer to the *ZMF for Eclipse User's Guide* for more information.

Deleting a z/OS Development Library

To delete a z/OS personal development library on the mainframe from within the workbench, perform the following steps.

1 Open the Serena perspective in the workbench

2 In the **Serena Explorer** view, select the ZMF server where your personal libraries reside and log on.

3 Expand the selected ZMF server node in the in **Serena Explorer** navigation tree.

4 Expand the node associated with the type of file system — native z/OS PDS or z/OS Unix System Services (USS) HFS — where your development libraries reside.

- To access native z/OS PDS libraries, expand the **Data Sets** node.

- To access Unix-style HFS libraries, expand the **z/OS USS Files** node.

Your personal development libraries are displayed as folders under the relevant object type node in the **Serena Explorer** navigation tree. Library members or components appear as files (terminal nodes) in the tree.

5 Right-click on the library you want to delete in **Serena Explorer** to bring up its contextual menu. Select the **Delete** option.

> **TIP** An alternative is to select a library and press the **Delete** key.

6 A dialog box asks you to confirm the delete request. If the selected library is correct, click **OK**.

Integrating a Desktop Workspace with ZMF

Java Perspective Integration

Mass Download and Mass Checkin

ZMF for Eclipse integrates ChangeMan ZMF with the Java perspective of the workbench and Java projects in the desktop workspace using the **Team** menu of the **Package Explorer** view. As part of the support for this integration, mass download and mass upload with automatic change control at the application level have been added to ZMF for Eclipse. This enables development of mainframe Java code on the desktop, where you can

take advantage of powerful GUI Java editors and the many open-source code libraries, code generators, developer toolkits, debuggers, documentation generators, and more that are available for Java. At the same time, ChangeMan ZMF retains control of software assets.

Automated
Library Mapping

Dynamic integration between the Java perspective and ChangeMan ZMF relies on automated library mapping between desktop workspace directories and mainframe software repository libraries. Customers configure the desired library mappings in the ECLIPSE member of the ZDDOPTS parameter library for ChangeMan ZMF. These mappings ensure that file types and desktop filename extensions correspond in a one-to-one fashion with the ZMF library types defined for an application.

Desktop workspaces must be structured in accordance with the relevant library mappings for automated component checkout, checkin, and lock to work as intended.

Best Practices for ZMF Desktop Workspaces

Any folder on the desktop can be defined as a workspace to Eclipse or RDz. To use such as folder as a personal development library with ChangeMan ZMF, the following best practices are recommended.

- ***Dedicate a workspace*** to the exclusive use of ZMF applications. Always switch to the ZMF workspace before working with components managed by ChangeMan ZMF.

> **TIP** If you frequently work with ZMF applications, you may wish to make the ZMF workspace the default workspace for the workbench.

- ***Organize the workspace by ZMF application, then by file type*** (binary, source, HTML, and so on) within application. For example:
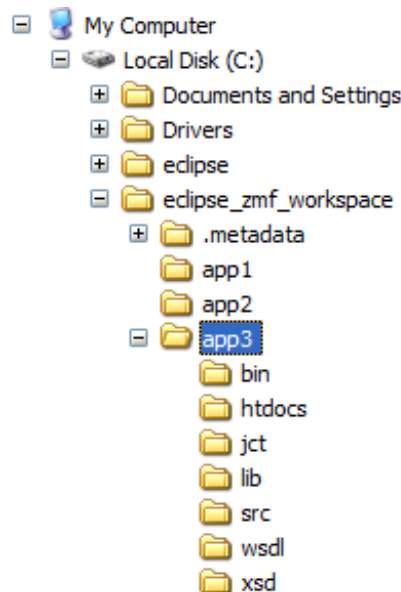
- ***All files in the same folder must be of the same file type.*** This is necessary so that conversions between ASCII and EBCDIC character formats proceed properly across systems (for example, excluding binaries but including text files), and so that ZMF build jobs treat different object types correctly.

Enabling Library Mapping in ZDDOPTS

Library Mapping
Prerequisites

Library mapping between mainframe library types in the ZMF repository and their corresponding desktop file types and folders is managed automatically by ZMF for Eclipse if the necessary library maps exist on the ZMF server. Library maps must be defined in the `ECLIPSE` member of ZDDOPTS for each ZMF server. Different library type-to-folder mappings may be specified for different ZMF applications.

For details about all ZDDOPTS members, see Appendix A, "ZDDOPTS Parameter Reference" on page 65.

Add to Workspace
Option

If library mapping is not enabled in ZDDOPTS, the **Add to Workspace** option will be grayed out in the contextual menus for the relevant applications in Serena Explorer.

Library Mapping Parameter Syntax

ZDDOPTS
Syntax

All parameters in ZDDOPTS are defined using XML. The `ECLIPSE` member of ZDDOPTS contains one or more application mapping profiles. Each mapping is defined by a `<profile>` element that identifies the relevant application in its `application` attribute. Individual library mappings within an application are defined using the `<libtype>` element, which may be repeated.

A library map for a given application has the following general form:

```
<profile application="aaaa">

   <libtype name="bbb"
            {eclipsefolder="bbb"|"c...c"|""}
            {binary="Y|N"                  }
            {root="Y|N"                    } />
   . . .

</profile>
```

where

- one `<profile>` tag is required for each ZMF application that allows checkout to a desktop workspace development library

- any number of `<libtype>` subtags are allowed within a `<profile>` tag

ZDDOPTS Library Mapping Example

```
<profile application="APP1">
   <libtype name="ECL" eclipsefolder=""
                       binary="N" root="Y" />
   <libtype name="JV1" eclipsefolder="src"
                       binary="N" root="N" />
   <libtype name="JIN" eclipsefolder="lib"
                       binary="Y" root="N" />
   <libtype name="WSD" eclipsefolder="wsdl"
                       binary="N" root="N" />
   <libtype name="WIN" eclipsefolder="htdocs"
                       binary="Y" root="N" />
   <libtype name="JCT" eclipsefolder="JCT"
                       binary="N" root="N" />
   <libtype name="WCT" eclipsefolder="WCT"
                       binary="N" root="N" />
</profile>
```

XML Elements and Attributes for Library Mapping

Element or Attribute Name	Usage	Data Type	Allowed Values & Comments
`<profile>`	1 - n Required	Complex	■One per ZMF application ■Unbounded per ZMF site
`application`	Required	1-4 bytes, character	ZMF application name
`<libtype>`	1 - n Required	Complex	■ One per ZMF library type ■ Unbounded within ZMF application
`name`	Required	3 bytes, character	Name of ZMF library type to be mapped to a desktop workspace folder
`binary`	Optional	1 byte, character	Y = Yes, binary objects (executables, graphics, compressed data) should be excluded from ASCII/EBCDIC conversion N = No, source or text object should not be excluded from conversion (default)
`eclipsefolder`	Optional	0-n bytes, character	Name of desktop workspace folder to associate with ZMF library type. All objects in a desktop folder will have the same filename extension as the folder name. NOTES: ■Defaults to same value as `name` attribute. ■Must be set to null if `root="Y"`.
`root`	Optional	1 byte, character	Y = Yes, objects in ZMF library should be copied to root directory (Java only) N = No, objects in ZMF library should not be copied to root directory (default) NOTES: ■Non-Java applications must use "N". ■If "Y", must set value of `eclipsefolder=""`.

Linking Development Libraries to RDz Projects

Overview of RDz Work Area Setup

Users of Rational Developer for z Systems (RDz) will typically want to use RDz projects to manage their personal development libraries on the mainframe. In general, setting up a development work area on the mainframe for use with RDz requires the following:

- Creating a z/OS connection in RDz for each system where a ZMF repository and any related personal development libraries reside.

- Creating an RDz project for each set of related development libraries.

  - An MVS subproject is needed to contain any PDS development libraries associated with a project.

  - A z/OS Unix subproject is needed to contain any HFS libraries associated with a project.

- Creating or modifying viewing filters for your personal development libraries in the Remote Systems Explorer perspective of RDz.

- Assigning development libraries to specific RDz subprojects.

RDz Projects and
ChangeMan ZMF

You can link your personal development libraries on z/OS to any desired z/OS project in RDz. At the same time, ZMF for Eclipse provides access in RDz to z/OS personal development libraries and to resources under change control in ChangeMan ZMF repositories. This permits integrated ZMF checkouts to RDz projects and lets you take advantage of the mainframe development tools in RDz while working with ZMF-managed software assets.

Desktop
Workspaces
and RDz

Java developers can also set up a desktop workspace to support local development of mainframe Java applications. The local workspace can be integrated with ZMF for versioning and building management (see "Integrating a Desktop Workspace with ZMF"). At the same time, it can be associated with a z/OS Unix project in RDz to take advantage of mainframe-specific development and debugging tools — for example, to support a DB2 relational database back end.

Creating a New z/OS Connection in RDz

A z/OS connection must be configured in RDz for each mainframe system hosting a ZMF repository that you want to work with using RDz. The RDz connection to z/OS is independent from the connection to z/OS and ZMF used by ZMF for Eclipse.

> **NOTE** The z/OS connection in RDz is used by the **Remote System Explorer** perspective and the **z/OS Projects** perspective of RDz. The z/OS and ZMF connection in ZMF for Eclipse is used by the **Serena** perspective and by the **Team** integration menu in the **Java** perspective in RDz and in the open-source Eclipse Workbench.

Follow the instructions that are provided in the RDz user manuals that IBM supplies to create a new z/OS connection in RDz.

Creating an RDz Project

RDz Project
Checkout

Personal development libraries on the z/OS server must be connected to an RDz project before you can work with them in RDz. Once this is set up, you can use ZMF for Eclipse to check out components from a ZMF baseline library into a personal development library and its associated RDz project at the same time.

Follow the information provided in the RDz user manuals that IBM provides to create an RDz project.

The new project should appear in the **z/OS Projects** navigation view of the z/OS Projects perspective.

Adding an RDz Subproject to a Project

Projects,
Subprojects, and
File Systems

RDz uses different subproject types to manage resources in different mainframe file systems. PDS development libraries in the native z/OS file system must be linked to an MVS subproject. HFS development libraries in the z/OS Unix System Services (USS) file system must be linked to a z/OS Unix subproject.

Separation of file systems at the subproject level supports different build procedures for different file systems. At the same time, related resources in different file systems can be managed together at the project level.

Follow the instructions given in the RDz user manuals that IBM provides to create an MVS subproject or a z/OS Unix subproject.

The new subproject should display under the node for the parent project in the **z/OS Projects** navigation view.

Creating Remote Systems Viewing Filters

View Filters and Project Filters

Filters can be created in RDz for two different purposes: (1) managing your view of mainframe resources in the **Remote Systems** navigation view; and (2) dynamically associating selected resources with projects and subprojects in the **z/OS Projects** navigation view. There need not be any correspondence between the two sets of filters.

Generally you will want to be able to view many more development libraries and other files in the **Remote Systems** view than you will actually want to associate with particular projects. For purposes of coordinating your work across perspectives, it is helpful to have a filter in the **Remote Systems** view that shows all your PDS development libraries and another that shows all your HFS libraries in the same way they are shown in the Serena perspective. More narrow filters can be added to these as needed.

Follow the instructions given in the RDz user manuals that IBM provides to create viewing filters.

Adding a Library to an RDz Subproject

Development resources are added to subprojects in different ways, depending on the subproject type.

- *MVS Subprojects* — PDS libraries or individual members are explicitly and statically linked to a subproject. In the **z/OS Projects** navigation view, PDS library and member names are shown directly beneath the node for the subproject to which they are linked.

- *z/OS Unix Subprojects* — HFS directories or files are linked dynamically to a subproject by means of selection filters. In the **z/OS Projects** navigation view, filter names are shown below the node for the subproject to which HFS directories and files are linked. Expand the filter node to see the HFS directories and files that currently meet the selection criteria for inclusion in the subproject.

The libraries added to a subproject must reside in the type of file system that the subproject supports.

IMPORTANT! Every MVS subproject must include a PDS data set (not just members), and every z/OS Unix subproject must include a directory (not just files), to enable ZMF checkout.

Follow the instructions given in the RDz user manuals that IBM provides to add a PDS library or member to an MVS subproject or add an HFS library or file to a z/OS Unix subproject.

In the **z/OS Projects** view, expand the named project and subproject nodes. The libraries or files you added should display.

Refer to the ChangeMan ZMF for Eclipse 8.1.2 User's Guide for information about the Smart Editor.

Appendix A
ZDDOPTS Parameter Reference

ZDDOPTS Functions and File Organization

ChangeMan ZMF administrators manage ZMF server interactions with Web Services clients by setting parameters in the ZDDOPTS parameter library on the mainframe. Clients affected by ZDDOPTS settings include ChangeMan ZDD and ZMF for Eclipse.

> **NOTE** The content of ZDDOPTS members is validated at the time the user logs on to ZMF for Eclipse. If an error is found, error messages are written to the .log file and are also displayed in a message box. The parsing of a ZDDOPTS member stops when an error, if any, is found."

Configurable Functions

Many ZDDOPTS parameters provide feature-specific, field-level customization of Web client data entry options. Customizations can vary by ZMF application. For example, you can:

- Specify data validation requirements for any field

- Add ZMF user-defined variables to data entry windows

- Require an entry in a normally optional field

- Make an editable field read-only for Web client users

In addition, ZDDOPTS provides three sets of options that are particularly relevant to the configuration of ZMF for Eclipse. These options are configured in the following library members:

- `ECLIPSE` — Configures automated mapping between ZMF library types and desktop workspace file types and directories in ZMF for Eclipse.

- `COMMAND` — Selectively enables or disables individual Web Services commands for all web clients that access the ZMF server. For example, a ZMF administrator could turn off the display of client menu items referencing ZMF options that are not installed on the mainframe.

- EOPTIONS — Defines enterprise-wide preferences for ZMF4ECL.

> **IMPORTANT!** If you update a ZDDOPTS member on the server, you must disconnect from and reconnect to the server for the new settings to take effect.'
>
> It is *not* necessary to restart the ZMF started task.

ZDDOPTS Members

Data Set Members

ZDDOPTS is actually a PDS/E (Partitioned Data Set/Extended) library containing multiple members. Each member is an XML text file that specifies parameter values for a specific ZMF function. The following members can be included in a ZDDOPTS library.

Member	Description
AUDIT	Specifies options for the ZMF package audit function.
BUILD	Specifies field labels, edit rules, and default values for ZMF component build, recompile, and relink functions.

Member	Description
COMMAND	Enables or disables ZMF menu commands in the client.
DEMOTE	Specifies options for ZMF change package demotion from a test environment.
ECLIPSE	Defines mappings between ZMF library types and desktop directories & file extensions for ZMF for Eclipse.
EOPTIONS	Defines enterprise-wide preferences for ZMF4ECL. For example, you can choose to have ZMF4ECL get a list of applications at logon and determine if the user has update access to these applications. This list is subsequently used when check-in processing is executed.
LIBTYPE	Specifies ZMF library types to display or hide in the client.
PKGCREAT	Specifies options for the ZMF package create function.
PKGPROP	Selects change package properties that a Web Services client may update.
PROMOTE	Specifies options for ZMF change package promotion to a test environment.
SYSLIB	Permits the customization of library paths for RDz-generated JCL and RDz Editors.

ZDDOPTS Allocation

DDNAME Requirements

The ZDDOPTS parameter library must be allocated to the ZMF started task using DDNAME ZDDOPTS:

```
//ZDDOPTS DD DSN=somnode.ZDDOPTS,DISP=SHR
```

Multiple ZDDOPTS Files

A ZDDOPTS file must be configured for each ZMF server that allows Web Services client access. If you want to use the same options for all ZMF servers, allocate the same ZDDOPTS library to all servers. If you want unique behavior for a server, create a different physical library with the desired parameter settings and code its name in the ZDDOPTS DDNAME statement in your JCL.

Shared Parameter Syntax

XML Syntax

Each member in the ZDDOPTS library must be a well-formed XML page. However, due to issues with case handling in ZMF on the mainframe, the rules for case in ZDDOPTS are not strictly compliant with the XML standard.

Case Sensitivity

In ZDDOPTS:

- *The names of elements (tags) and attributes* are case sensitive. They must be coded as shown in this document (usually all lower case) for the parser to recognize them.

- *Attribute values* are not case sensitive, as the parser forces them to upper case before passing them to ZMF.

> **NOTE** The XML syntax used in ZDDOPTS is not related in any way to the syntax used with the ZMF XML Services.

Wildcard Patterns

Some attributes in ZDDOPTS XML tags accept wildcard patterns as values. Use the following wildcard characters in a pattern string:

Character	Function
*	Matches zero or more characters
?	Matches a single character

For example:

- The pattern "A*" matches any string that starts with the letter A.

- The pattern "*Z" matches any string that ends in the letter Z.

- The pattern "A*Z" matches any string that starts with A and ends with Z.

- The pattern "A??D" matches any string that starts with A, followed by exactly two characters, followed by a string-final 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 string-final DE.	ABCDE AXYDE	ABCDEF AXYZDE
*	Match any string.	Z ABCDEFGH	

Multiple Attribute Values

Some ZDDOPTS XML tags have attributes that allow multiple values to be passed through as a parameter string to ZMF. From an XML perspective, there is only one attribute value — a single character string containing spaces. However, the spaces are treated as parameter delimiters by ZMF. The entire parameter string must be contained within one set of double quotes in ZDDOPTS.

Most attributes of the `<profile>` element accept parameter strings as values. For example:

```
<profile application="TST* DEMO X*">
```

<options> Element

The top-level XML element (or tag) for each ZDDOPTS XML page is the `<options>` element.

<options> Syntax

The `<options>` element has the following general form:

```
<options name="member" strict="Y|N">
 ...
</options>
```

where

member   is the ZDDOPTS member name (such as AUDIT or ECLIPSE) corresponding to the ZMF function configured

<options> Attributes

The `<options>` element always takes the `name` and `strict` attributes, regardless of the member in which it appears. Attributes for the `<options>` element have the following requirements:

Attribute Name	Usage	Data Type	Multiple Values	Description & Values
name	Required	1-8 char	No	ZDDOPTS member name. Corresponds to ZMF function configured. Allowed values: ■ AUDIT ■ BUILD ■ COMMAND ■ DEMOTE ■ ECLIPSE ■ LIBTYPE ■ PKGCREAT ■ PKGPROP ■ PROMOTE
strict	Optional	1 char	No	Parser strictness. Values: Y = Yes, strictly enforce tag names and issue error if name not recognized **N** = No, ignore unrecognized tag names & do not issue error if found (**default**)

<options> Subtags

Every `<options>` element contains at least one subordinate `<profile>` element whose contents vary by ZDDOPTS member name (which is to say, by ZMF function configured). Some members also allow additional elements within the `<options>` tag.

The following subordinate elements or subtags may be nested within an `<options>` element.

Subtag Name	Usage	Description & Values
`<multiple>`	0-1 in BUILD	Sets defaults for components in build jobs based on saved history. Optional, used in BUILD only.
`<profile>`	1 - n, required	Configuration options for a ZMF application. Used in all ZDDOPTS members. Requirements: ■ One per ZMF application ■ Unbounded per ZMF site

<options> Example

```
<options name="PROMOTE">

  <profile application="X* Y* Z*">
    <field name= .../>
    <field name= .../>
  </profile>

  <profile application="*">
    <field name="userVariable05"
           label="Turnover"
           required="Y">
  </profile>

</options>
```

<profile> Element

The second-level XML element (or tag) for each ZDDOPTS XML page is the `<profile>` element. This element serves as a container for a set of Web client options that apply to a particular ZMF application or set of applications. At least one `<profile>` element is required within an `<options>` element.

<profile> Syntax

The `<profile>` element has the following general form:

```
<profile [application=["app1 {app2 app3 ...}"|"*"] |
         user=["user1 {user2 user3 ...}"|"*"    ] ]
         {language=["lang1 {lang2 lang3 ...}"|"*"] }
         {libtype=["lib1 {lib2 lib3 ...}"|"*"    ] }
         {procname=["proc1 {proc2 proc3 ...}|"*"  ] }
 ...
</profile>
```

where

   `appn`   = *n*th ZMF application ID or wildcard pattern

   `langn`   = *n*th compiler/translator language or wildcard pattern

   `libn`   = *n*th ZMF library type or wildcard pattern

   `procn`   = *n*th ZMF build procedure name or wildcard pattern

> usern = *n*th TSO user ID or wildcard pattern

Either the `application` attribute or the `user` attribute is required, but they are never used together. Other attributes are optional, depending on the ZDDOPTS member (that is, the ZMF function configured).

<profile> Attributes

The `<profile>` element takes one or more of the following attributes, depending on the member in which it appears. All attributes accept multiple values delimited by spaces, with wildcard patterns allowed for each value. All attribute values default to the asterisk (*) wildcard, which matches any value.

Attributes for the `<profile>` element have the following requirements:

Attribute Name	Member Usage	Data Type	Multiple Values	Description & Values
`application`	AUDIT, BUILD, DEMOTE, ECLIPSE, LIBTYPE, PKGCREAT, PROMOTE	1-4 char, wildcard	Yes	ZMF application(s) to which profile applies. Default value is **"*"**.
`language`	BUILD	1-8 char, wildcard	Yes	Language(s) or compiler(s) to which profile applies. Default value is **"*"**.
`libtype`	BUILD	3 char, wildcard	Yes	ZMF library type(s) to which profile applies. Default value is **"*"**.
`procname`	BUILD	1-8 char, wildcard	Yes	ZMF build procedure(s) to which profile applies. Default value is **"*"**.
`user`	COMMAND, PKGPROP	1-8 char, wildcard	Yes	TSO user ID(s) to which profile applies. Default value is **"*"**.

<profile> Subtags

Subordinate elements or subtags contained within the `<profile>` element vary by ZDDOPTS member name (that is, by ZMF function configured). The following subordinate elements or subtags may be nested within an `<profile>` element.

Subtag Name	Usage	Description & Values
`<command>`	1-n required in COMMAND. 0 elsewhere	
`<field>`	1-n required in AUDIT, BUILD, DEMOTE, PKGCREAT, PKGPROP, PROMOTE. 0 elsewhere	Sets defaults for components in build jobs based on saved history. Used in BUILD member only.

Subtag Name	Usage	Description & Values
`<libtype>`	1-n required in ECLIPSE & LIBTYPE. 0 elsewhere	Configuration options for a ZMF application. Used in all ZDDOPTS members. Requirements: ■ One per ZMF application ■ Unbounded per ZMF site

<profile> Example

```
<options name="PROMOTE">

    <profile application="X* Y* Z*">
       <field name= .../>
       <field name= .../>
    </profile>

    <profile application="*">
       <field name="userVariable05"
              label="Turnover"
              required="Y">
    </profile>

</options>
```

AUDIT Member Parameters

The AUDIT member of the ZDDOPTS library specifies field-level options for ZMF audit requests submitted by a Web Services client. These options may restrict, but may not expand, the more general settings defined for mainframe users connected to ZMF using the ISPF interface.

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.

Syntax

The XML parameter syntax for an AUDIT member is as follows:

```
<options name="AUDIT" strict="Y|N">

  <profile application="app1 app2 app3 ...">
    <field name= .../>
    <field name= .../>
     ...
  </profile>

   ...

  <profile application="... appn">
     <field name= .../>
     <field name= .../>
      ...
```

```
            </profile>

        </options>
```

Elements

This table shows the elements for the AUDIT member.

Element	Description	Usage	Attributes
`<options>`	Top-level document element	1	`name` `strict`
`<profile>`	Container for field specifications for particular ZMF application(s). Nested within `<options>` element.	1 - n	`application`

`<field>`	Properties for a particular field. Nested within `<profile>` element	1 - n	`name`
			`length`
			`default`
			`readonly`
			`label`
			`required`
			`uppercase`
			`validation`

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.	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

Attribute	Description / Values	Multiple Values	Length	Default
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 "NUMERIC" 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

Name Value	Description	User Defined	Values	Length
UserVariable06 - UserVariable10	User options	Yes	Any text, subject to validation	0-72

Example

Sample ZDDOPTS members are provided with the ZMF for Eclipse product media. After installing ZMF for Eclipse, you can view these samples at the following default location on your PC:

```
C:\Program Files\Serena\ZMF\710\ZEclipse\zddopts
```

BUILD Member Parameters

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

Element	Description	Attributes
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	"*"

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 .	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

Attribute	Description / Values	Multiple Values	Length	Default
length	The maximum number of characters allowed for entry in a UserOption field. Cannot exceed the field size defined for that particular UserOption. See 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 ChangeMan 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"
validation	Validation performed on data entered in this field. **NOTE** This is an attribute for the UserOption fields only. "YN" Y or N "ALPHA" A-Z or national (#, $, @) "ALPHANUM" A-Z, national (#, $, @), or 0-9 "DSNAME" A valid data set name "NUMERIC" 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 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
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

Name Value	Description	User Defined	Values	Length
UserOption7201 – UserOption7205	User options	Yes	Any text, subject to validation	0-72

Example

A sample BUILD member is delivered with the ChangeMan 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 Member Parameters

The COMMAND member of the ZDDOPTS library allows certain menu commands to be disabled in the user interface of remote clients such as ChangeMan ZDD and ZMF for Eclipse. 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 user="userid">
    <command name="someCommand" enable="Y"/>
    <command name= .../>
  </profile>

  <profile user="userid2">
    <command name= .../>
    <command name= .../>
  </profile>

</options>
```

Elements

This table shows the elements for the COMMAND member.

Element	Description	Attributes
options	Top-level document element.	name strict

Element	Description	Attributes
`profile`	Complete set of command definitions.	`user`
`command`	Enable or disable menu commands.	`name` `enable`

Attributes for <options> Element

These are 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

These are 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 <command> Element

The table below lists the attributes for the <command> element. Commands are always enabled by default.

Attribute	Description	Values	Multiple Values	Length	Default
name	Command to be enabled or disabled in the client. Values are NOT case sensitive. NOTE: If any specific commands are enabled, the "*" attribute value should be omitted or disabled.	"APPROVEPACKAGE" "AUDITPACKAGE" "BACKOUTPACKAGE" "BILLOFMATERIALS" "BROWSE" "BROWSECOMPONENT" "BROWSELISTING" "BUILDCOMPONENT" "CHECKINCOMPONENT" "CHECKOUTCOMPONENT" "CHECKOUTTOPERSONAL" "COMPONENTHISTORY" "COMPONENTWORKLIST" "CREATEPACKAGE" "DATASETVIEW" "DELETE" "DELETECOMPONENT" "DELETEPACKAGE" "DEMOTE" "DOWNLOAD" "EDIT" "EDITCOMPONENT" "FREEZEPACKAGE" "HMIGRATE" "HRECALL" "IMPACTANALYSIS" "JOBVIEW" "LOCKCOMPONENT" "NEW" "NEWCOMPONENT" "PACKAGEPROPERTIES" "PACKAGEQUERY" "PROMOTE" "PROMOTIONHISTORY" "REBUILDINSTALLJCL" "RECOMPILECOMPONENT" "REFREEZECOMPONENT" "RELINKCOMPONENT"	No	1 – 18	Value required

Attribute	Description	Values	Multiple Values	Length	Default
name (continued)		`"RENAMECOMPONENT"` `"RESETAUDITLOCK"` `"REVERTPACKAGE"` `"SAVEPASSWORD"` `"SAVEUSERID"` `"SCRATCHCOMPONENT"` `"SHOWBASELINE"` `"SHOWPROMOTION"` `"SOURCETOLOAD"` `"STAGINGVERSIONS"` `"SUBMITJOB"` `"SUBMITXML"` `"UNFREEZECOMPONENT"` `"UNLOCKCOMPONENT"` `"UPLOAD"` `"USSVIEW"` `"ZMFVIEW"` `"*"`			
enable	Indicates whether command is enabled or disabled.	"Y" "N"	No	1	"Y"

DEMOTE Member Parameters

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 <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 .	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

Attribute	Description / Values	Multiple Values	Length	Default
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 "NUMERIC" 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 ChangeMan 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.

ECLIPSE Member Parameters

The `ECLIPSE` member of ZDDOPTS defines the default mappings between mainframe library types in ChangeMan ZMF and Eclipse workbench folder names on the desktop. These mappings are used to manage mass downloads of ZMF application components to an Eclipse Java project or mass checkins from an Eclipse project into a ZMF change package.

Different library type-to-folder mappings may be specified for different ZMF applications.

Each ECLIPSE member contains one or more application mapping profiles. Each mapping is defined by a `<profile>` element that identifies the relevant application in its `application` attribute. Individual library mappings within an application are defined using the `<libtype>` element.

Syntax

A library map for a given application has the following general form:

```
<profile application="aaaa">

   <libtype name="bbb"
           {eclipsefolder="bbb"|"c...c"|""}
           {binary="Y|N"                 }
           {root="Y|N"                   }
   />
   . . .

</profile>
```

where

- one `<profile>` tag is required for each ZMF application that allows checkout to a desktop workspace development library

- any number of `<libtype>` subtags are allowed within a `<profile>` tag

Library Mapping <profile> Example

```
<profile application="APP1">
   <libtype name="ECL" eclipsefolder=""
                    binary="N" root="Y" />
   <libtype name="JV1" eclipsefolder="src"
                    binary="N" root="N" />
   <libtype name="JIN" eclipsefolder="lib"
                    binary="Y" root="N" />
   <libtype name="WSD" eclipsefolder="wsdl"
                    binary="N" root="N" />
```

```
          <libtype name="WIN" eclipsefolder="htdocs"
                             binary="Y" root="N" />
          <libtype name="JCT" eclipsefolder="JCT"
                             binary="N" root="N" />
          <libtype name="WCT" eclipsefolder="WCT"
                             binary="N" root="N" />
      </profile>
```

XML Elements and Attributes for Eclipse Library Mapping

Element or Attribute Name	Usage	Data Type	Allowed Values & Comments
`<profile>`	1 - n Required	Complex	▪ One per ZMF application ▪ Unbounded per ZMF site
application	Required	1-4 bytes, character	ZMF application name
`<libtype>`	1 - n Required	Complex	▪One per ZMF library type ▪Unbounded within ZMF application
name	Required	3 bytes, character	Name of ZMF library type to be mapped to a desktop workspace folder
binary	Optional	1 byte, character	Y = Yes, binary objects (executables, graphics, compressed data) should be excluded from ASCII/EBCDIC conversion N = No, source or text object should not be excluded from conversion (default)
eclipsefolder	Optional	0-n bytes, character	Name of desktop workspace folder to associate with ZMF library type. All objects in a desktop folder will have the same filename extension as the folder name. NOTES: ▪ Defaults to same value as `name` attribute. ▪ Must be set to null if `root="Y"`.
root	Optional	1 byte, character	Y = Yes, objects in ZMF library should be copied to root directory (Java only) N = No, objects in ZMF library should not be copied to root directory (default) NOTES: ▪ Non-Java applications must use "N". ▪ If "Y", must set value of `eclipsefolder=""`.

EOPTIONS Member Parameters

The `EOPTIONS` member of ZDDOPTS defines enterprise-wide preferences for ZMF4ECL. It consists of *keyword=value* directives.

Two keywords are available:

```
      CheckinEnq="Y/N"
```

The default is N. Specifying Y for this option directs ZMF4ECL to obtain an ENQ on the user's personal library member during checkin from personal. This is to ensure that the data set is not being updated during the checkin process.

```
CheckApplSecurity="Y|N"
```

The default is N. Specifying Y for this option will cause ZMF4ECL to get a list of applications at logon and determine if the user has update access to ensure that only applications/packages to which the user has update authority are presented for selection.

This list is subsequently used when check-in processing is executed.

> **NOTE** In ZMF instances that contain a large number of applications, setting this flag to Y will increase the time that it takes a user to connect.

LIBTYPE Member Parameters

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 ChangeMan ZMF Exit 35 you have made to suppress the display of "hidden" library types.

NOTE ChangeMan ZMF Exit 35 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 ChangeMan 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

Element	Description	Attributes
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
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 ChangeMan 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 Member Parameters

- The PKGCREAT member of the ZDDOPTS library specifies the attributes of user-defined package fields. The PKGCREAT member is used to customize the user interface for both the "New Package" dialog boxes, as well as the "Package Properties" dialog boxes.

- You can customize the user-defined field labels, field validation rules, default values, and designate fields as read-only. To place security rules on which users can alter package fields after a package is created, use the PKGPROP member. Unlike PKGCREAT, the PKGPROP member can be used for any package field, and not just user-defined fields.

- The PKGCREAT member includes one or more profiles. Each profile is a complete set of user-defined field definitions.

- Profiles are specified by application. You can specify one or more applications associated with each profile. Applications can be specified with wildcard patterns. The first matching profile will be 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 .	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

Attribute	Description / Values	Multiple Values	Length	Default
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 "NUMERIC" 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
UserVarLen16-UserVarLen98	Unavailable.			
UserVarLen101-UserVarLen115	User options.	Yes	Any text, subject to validation	0-1
UseVarLen199	Reserved field.			
UserVarLen201 - UserVarLen211	User options	Yes	Any text, subject to validation	0-2

Name Value	Description	User Defined	Values	Length
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 ChangeMan 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 Member Parameters

- The PKGPROP member of the ZDDOPTS library can be used to restrict which users are allowed to change fields within the package properties. Different permissions can be specified for each field.

- 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. Security 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)	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	Field name	"PackageTitle" "PackageLevel" "RequestorName" "RequestorPhone" "WorkRequestId" "Department" "UnplannedReason" "TemporaryDays" "Release" "PackageDescription" "ProblemAction" "Scheduler" "InstallInstructions" "PredessorJobs" "SuccessorJobs" "AffectedApplications" "InstallSite" "InstallDateTime" "ContactName" "ContactPhone" "ParticipatingPackages" "*"	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 ChangeMan 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
```

PROMOTE Member Parameters

The AUDIT, DEMOTE, and PROMOTE members of the ZDDOPTS library specify 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 <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 .	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. Field can be changed. "N"	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 "N" upper case. Field will remain as entered.	No	1	"N"

Attribute	Description / Values		Multiple Values	Length	Default
validation	Validation performed on data entered in the field.		No	0 – 8	"NONE"
	"YN"	Y or N			
	"ALPHA"	A-Z or national (#, $, @)			
	"ALPHANUM"	A-Z, national (#, $, @), 0-9			
	"NUMERIC"	0-9			
	"SYMBOL"	Same as ALPHANUM, except first character must be alphabetic or national.			
	"DSNAME"	Valid data set name			
	"NONE"	No validation performed			

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 ChangeMan 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.

SYSLIB Member Parameters

The ZDDOPTS SYSLIB member permits customization of the SYSLIB property value. SYSLIB is used when you select "Add to RDz subproject" and "Override RDZ Syslib Property" during the check-out process.

SYSLIB permits the customization of library paths for RDz-generated JCL and RDz Editors.

Here is a sample SYSLIB member:

```
<profile application="DEMO" language="COBOL">
<syslib type="PACKAGE" />
<syslib type="PARTICIPATING" />
<syslib type="PROMOTION" />
<syslib type="BASELINE" />
<syslib type="WALLPAPER" dataset="SYS1.COBLIB" />
<syslib type="WALLPAPER" dataset="SYS1.COBLIB2" />
</profile>
```

where:

- Application - identifies the ZMF application.

- Language - identifies the programming language. COBOL, PLI, and HLASM are the supported RDz languages.

- Type - identifies the type of syslib entry. Valid types and their definitions are:
  - PACKAGE - Package LIKE-CPY staging libraries.
  - PARTICIPATING - Package LIKE-CPY libraries in any participating packages.
  - PROMOTION - LIKE-CPY Promotion libraries.
  - BASELINE - LIKE-CPY Baseline libraries.
  - WALLPAPER - The library specified in the data set attribute.

If no matching SYSLIB member is found, these default entries will be used:

- `<syslib type="PACKAGE" />`

- `<syslib type="PARTICIPATING" />`

- `<syslib type="PROMOTION" />`

- `<syslib type="BASELINE" />`

Appendix B

Error Messages and Troubleshooting

Component Browse Function Fails

Error Message   When attempting to browse ChangeMan ZMF components, the following error occurs:

```
Could not initialize the editor
```

Explanation   This error occurs when a network firewall blocks access to the XCH (Sernet Connect) and/or ZMF (CMN) ports for the target ChangeMan ZMF instance on the mainframe.

Resolution   To resolve this problem, first determine the XCH and ZMF port numbers for the ZMF instance you are trying to access. These can be found on the ZMF server information tab in the Serena perspective of ZMF for Eclipse.

Provide the XCH and ZMF port numbers to your network administrator and request that these be opened for client access over the TCP/IP network where your web application server resides.

Checkout to RDz Project Fails

Error Message   After selecting **ZMF-RDz | Checkout** from the **Serena Explorer** contextual menu and providing a valid RDz project and subproject name, you may encounter a message like the following:

```
Project undefined or no data set
```

Explanation   This error occurs when no PDS libraries or HFS directories have been linked to the subproject targeted by the checkout. An RDz project or subproject is a logical collection, not a physical container for checked out files. A physical library or directory from the RDz **Remote Explorer** view must be linked to an RDz subproject before ZMF checkout.

Resolution   Link a Component Browse Function Fails102physical library or directory in the RDz **Remote Explorer** view to an appropriate RDz subproject. Then retry the ZMF checkout.

Exception When Sharing an Eclipse Project

Error Message   If the ZDDOPTS configuration library is not installed with ChangeMan ZMF on the mainframe, or if the ECLIPSE member of that library is missing or incorrect, you will encounter the following message when you attempt to share an Eclipse project with ZMF:

```
'Share Eclipse project with ZMF' has encountered a problem -
Exception running background job Share Eclipse with ZMF
```

Explanation   Each ChangeMan ZMF instance on the mainframe must be configured to support access from ZMF for Eclipse. To do this, your ZMF administrator must install the ZDDOPTS configuration library on each ZMF instance that will be accessed using ZMF for Eclipse. In addition, the ECLIPSE member of each library must be appropriately configured for library type mapping between the mainframe and your Eclipse project.

Resolution   Instructions for ZDDOPTS setup are provided in this manual. See "Enabling Plug-in Support in ChangeMan ZMF" on page 38 for installation instructions. Detailed syntax specifications for the ECLIPSE member (and others, if desired) are provided in Appendix A, "ZDDOPTS Parameter Reference".

Code Page Issues with Comments in ZDDOPTS

Error Message   The first time you execute a ZMF process that uses the XML specification in a ZDDOPTS configuration library, you may encounter an error such as the following:

```
ChangeMan [ChangeMan ZMF instance] on server [ChangeMan ZMF server]
has invalid XML data specified for ZDDOPTS in [library(member)].

Missing equals sign between attribute and attribute value.
```

Explanation   This message occurs when in an XML comment in ZDDOPTS is processed using a non-English character code page.

Serena supplies sample ZDDOPTS members on the product media for ChangeMan ZMF and ZMF for Eclipse. The sample members were created for use with English-language character code pages on the mainframe. If you are using different code pages, you may need to modify the XML comments in these sample ZDDOPTS members.

All ZDDOPTS members use standard XML syntax for comments, as follows:

```
<!-- Some comment text -->
```

In English-language code pages, the hexadecimal code for the exclamation point (!) is x'5A'. This hexadecimal code translates into other characters in certain code pages you may use on the mainframe. This can cause XML parsing errors.

The following table shows the character translation for the English exclamation character code (x'5A') in common mainframe code pages:

Code Page	Language	Character 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	Ð

Code Page	Language	Character Translation
01156	Latvian, Lithuanian	!
01157	Estonian	_
01158	Ukrainian	]
01164	Vietnamese	]

Resolution   To correct this problem, do either of the following:

- ■ **COPY ZDDOPTS from the ChangeMan ZMF sample library** to your production library. Then manually edit all the comments in all ZDDOPTS members to ensure that all comments begin with `<!--` when rendered using your currently installed code page.

- ■ **UPLOAD the ZDDOPTS sample delivered with ZMF for Eclipse** from the client to the mainframe. The upload process will replace character code `x'5A'` with the correct exclamation character code for your currently installed code page.

Refer to <u>Solution S140158</u> in the Knowledgebase on the Serena Support web site for more information about the use of code pages for Serena's international customers.

ZDDOPTS Updates Do Not Affect Workbench

Symptom   Changes are made to a ZDDOPTS member on the mainframe, but the behavior of ZMF for Eclipse does not change.

Explanation   You must refresh the ZMF server settings on the client for the new settings to take effect in ZMF for Eclipse.

Resolution   Disconnect from and reconnect to the ZMF Server. It is **not** necessary to restart the ZMF started task.

Index

A

C

D

E

R

Rational Developer for z Systems (RDz) requirements 22
RDz projects
     creating 62
     work area setup 61
RDz subprojects
     creating 62
     file systems and 62
     relation to file systems 62
     relation to projects 62
repository 15
repository services 56

S

security requirements 20, 21
security, z/OS 21
Smart Editor 63
software compatibility 8
staging versions 56
subprojects
     adding development libraries 63
     MVS 63
     z/OS Unix (USS) 63

T

TCP/IP connection, verifying 47
TCP/IP requirements 20
     XCH port 20
     ZMF (CMN) port 20
Team menu 58
troubleshooting
     code page issues 103
     component browse function fails 102
     editor won't initialize 102
     exception when sharing Eclipse project with ZMF 102
     ZDDOPTS changes don't show 104
     ZDDOPTS issues 103
typographical conventions 11

U

uninstalling
     workbench plug-in 33
     ZMF Web Services 24, 26
use cases
     COBOL application maintenance 16, 17
     enterprise e-commerce with Java 15, 16

W

X

Z