SERENA®

ChangeMan® ZMF 8.1

Web Services Getting Started Guide

Serena Proprietary and Confidential Information

Trademarks

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

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1850 Gateway Drive, San Mateo California, 94404.

Publication date: November 2015

Table of Contents

Welcome to the *Serena ChangeMan ZMF Web Services Getting Started Guide*

ChangeMan ZMF Summary
Serena® ChangeMan® ZMF is a comprehensive and fully integrated solution for software change management in z/OS environments. It provides reliable and streamlined implementation of software changes from development into production. ChangeMan ZMF manages and automates the application life cycle, ensures current and complete software builds, protects the integrity of the code migration process, and delivers higher quality code to test and production environments.

Web Services API Summary
The ChangeMan ZMF Web Services Application Program Interface (API) is designed for customers who want to develop their own intranet, extranet, or Internet client applications for ChangeMan ZMF in a Service Oriented Architecture (SOA) environment.

Because the Web Services API is built on the ChangeMan ZMF XML Services, Web clients have programmatic access to the full range of ZMF functionality. Because access is enabled through the industry-standard Simple Object Access Protocol (SOAP) messaging protocol over the HTTP transport protocol, client applications can be developed in any programming language that supports these standards, including Java and C++ for open-source clients and .NET and C# for Windows clients.

ChangeMan ZMF services are described with the Web Service Description Language (WSDL), so you can take advantage of the many development tools that enable rapid development of Web client applications from WSDL specifications.

Web-enabled ZMF integration products, such as ChangeMan ZMF for Eclipse, are not required in order to use the ZMF Web Services API.

About This Book

Objective
The *ChangeMan ZMF Web Services Getting Started Guide* provides installation and quick-start programming guidance for the Web Services API to functions and facilities of ChangeMan ZMF. More general aspects of Web Services technology, tools, or implementation are not addressed.

> **NOTE** This document discusses the ChangeMan ZMF Web Services API only. See the documentation provided with your particular Web Services application server for information about installing and configuring that server.

Audience
This document is intended for software developers, systems engineers, and test engineers who will install, develop, and test Web Services applications that access the functions of ChangeMan ZMF. Familiarity with the features of ChangeMan ZMF and its XML Services API are assumed. Familiarity with open-source Web Services tools and technologies is also assumed.

Before You Begin
See the ChangeMan ZMF Readme file for the latest updates and corrections to this manual.

Change bars in the left margin (like that shown at left) identify substantive changes to this publication in ChangeMan ZMF release 8.1.

Related Documentation

This manual provides only the essential information needed to take advantage of the Web Services API for ChangeMan ZMF. It does not duplicate generic information about the SOAP protocol, WSDL, or other information that applies to all SOA Web Services applications. This manual does not duplicate information already described in detail in other ChangeMan ZMF documentation. Customers are referred to the following related documentation for such information.

Web Service Standards Documentation

The following reference information about SOAP and Web Services standards applies to most SOA Web Services environments.

- *Web Services Basic Profile Version 1.0* — Available from the Web Services Interoperability Organization (WS-I) at

  http://www.ws-i.org/Profiles/BasicProfile-1.0.html

- *Simple Object Access Protocol (SOAP) Version 1.1* — Available from the World Wide Web Consortium (W3C) at

  http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

- *Web Services Description Language (WSDL) Version 1.1* — Available from the World Wide Web Consortium (W3C) at

  http://www.w3.org/TR/wsdl

- *Apache Axis Version 1.3* — Available from the Apache open source Web site at

  http://www.apache.org/dyn/closer.cgi/ws/axis/1_3

NOTE Refer to the *Readme* file for the latest information on supported versions and required enabling technology.

ZMF Web Services Sample Code

Sample code and instructions for running it are attached to solution number S139564 in the Serena knowledgebase.

The sample code includes 2 Java source files that illustrate:

- Logging on to ZMF
- Getting a copy of the ZMF global parameters
- Getting a list of ZMF package components
- Logging off

You can download the zip file containing the code and instructions from the following link:

http://knowledgebase.serena.com/InfoCenter/index?page=content&id=S139564

ChangeMan ZMF Documentation Suite

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

Manual	Description
Administrator's Guide	Describes ChangeMan ZMF features and functions with instructions for choosing options and configuring global and application administration parameters.
ChangeMan ZMF Quick Reference	Provides a summary of the commands you use to perform the major functions in the ChangeMan ZMF package life cycle.
Customization Guide	Provides information about ChangeMan ZMF skeletons, exits, and utility programs that will help you to customize the base product to fit your needs.
DB2 Option Getting Started Guide	Describes how to install and use the DB2 Option of ChangeMan ZMF to manage changes to DB2 components.
ERO Concepts	Discusses the concepts of the ERO Option of ChangeMan ZMF for managing releases containing change packages.
ERO Getting Started Guide	Explains how to install and use the ERO Option of ChangeMan ZMF to manage releases containing change packages.
ERO Messages	Describes system messages and codes produced by ChangeMan ZMF ERO.
ERO XML Services User's Guide	Documents ERO functions and services available for general customer use. These services are also known as the green services and provide primarily search and query functions.
High-Level Language Functional Exits Getting Started Guide	Provides instructions for implementing and using High-Level Language (Cobol, PL/1, and REXX) exits, driven consistently by all clients to enforce local business rules in ZMF functions.
IMS Option Getting Started Guide	Provides instructions for implementing and using the IMS Option of ChangeMan ZMF to manage changes to IMS components.
INFO Option Getting Started Guide	Describes two methods by which ChangeMan ZMF can communicate with other applications: ■ Through a VSAM interface file. ■ Through the Tivoli Information Management for z/OS product from IBM.
Installation Guide	Provides step-by-step instructions for initial installation of ChangeMan ZMF. Assumes that no prior version is installed or that the installation will overlay the existing version.

Manual	Description
Java / HFS Getting Started Guide	Provides information about using ZMF to manage application components stored in UNIX System Services (USS) file systems, especially Java application components.
Load Balancing Option Getting Started Guide	Explains how to install and use the Load Balancing Option of ChangeMan ZMF to connect to a ZMF instance from another CPU or MVS image.
M+R Getting Started Guide	Explains how to install and use the Merge and Reconcile (M+R) Option of ChangeMan ZMF to consolidate multiple versions of source code and other text components.
M+R Quick Reference	Provides a summary of M+R Option commands in a handy pamphlet format.
Messages	Explains messages issued by ChangeMan ZMF, SERNET, and System Software Manager (SSM) used for the Staging Versions feature of ZMF.
Migration Guide Versions 6.1 and 7.1 to Version 8.1	Gives guidance for upgrading ChangeMan ZMF from Versions 6.1.*x* and 7.1.*x* to Version 8.1.
OFM Getting Started Guide	Explains how to install and use the Online Forms Manager (OFM) option of ChangeMan ZMF.
SER10TY User's Guide	Gives instructions for applying licenses to enable ChangeMan ZMF and its selectable options.
User's Guide	Describes how to use ChangeMan ZMF features and functions to manage changes to application components.
XML Services User's Guide	Documents the most commonly used features of the XML Services application programming interface to ChangeMan ZMF.
ZMF Web Services User's Guide	Documents the Web Services application programming interface to ChangeMan ZMF.

Using the Manuals

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

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

The PDF manuals include the following features:

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

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

- **Comments.** All PDF documentation files that Serena delivers with ChangeMan ZMF have enabled commenting with Adobe Reader. Adobe Reader version 7 and higher has commenting features that enable you to post comments to and modify the contents of

PDF documents. You access these features through the Comments item on the menu bar of the Adobe Reader.

- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.

- **Advanced search.** Starting with version 6, Adobe Reader includes an advanced search feature that enables you to search across multiple PDF files in a specified directory.

Searching the ChangeMan ZMF Documentation Suite

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

1   Download the ZMF All Documents Bundle ZIP file and the *ZMF Readme* to your workstation from the **Downloads** tab on the Serena Support Web site.

2   Unzip the PDF files in the ZMF All Documents Bundle into an empty folder. Add the *ZMF Readme* to the folder.

3   In Adobe Reader, select **Edit** | **Advanced Search** (or press **Shift+Ctrl+F**).

4   Select the **All PDF Documents in** option and use **Browse for Location** in the drop down menu to select the folder containing the ZMF documentation suite.

5   In the text box, enter the word or phrase that you want to find.

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

7   Click **Search**.

8   In the **Results**, expand a listed document to see all occurrences of the search argument in that PDF.

9   Click on any listed occurrence to open the PDF document to the found word or phrase.

Typographical Conventions

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

Convention	Explanation
italics	Introduces new terms that you may not be familiar with, and occasionally indicates emphasis.
bold	Emphasizes important information and field names.
UPPERCASE	Indicates keys or key combinations that you can use. For example, "press the ENTER key".

Convention	Explanation
monospace	Indicates syntax examples, values that you specify, or results that you receive.
monospaced italics	Indicates names that are placeholders for values you specify; for example, *filename*.
monospace bold	Indicates the results of an executed command.
vertical rule |	Separates menus and their associated commands. For example, "select File | Copy" means to select the Copy option from the File menu. Alternatively, indicates mutually exclusive choices in a command syntax line.

Chapter 1
Installation and Setup

Components of the ZMF Web Services Environment

The ZMF Web Services API is intended for "roll your own" use by Web-savvy customers across a variety of environments. Some standard components of the ZMF Web Services environment are installed on the host with the base ChangeMan ZMF product. Others vary greatly across sites according to the requirements of the customer. Because few assumptions about the application environment can be made in advance, the installation and use of the ZMF Web Services are intended to be tailored extensively by the customer.

The following components are required by the ChangeMan ZMF Web Services API:

- **ZMF XML Services** — The XML Services API to ChangeMan ZMF must be enabled on the host before installing ZMF Web Services support. See the *ChangeMan ZMF Installation Guide* for instructions on enabling external access to the XML Services.

- **SERNET** — The mainframe communications front-end for ChangeMan ZMF manages TCP/IP connectivity between ChangeMan ZMF and the ZMF Connector plug-in that resides on your Web Services application server. (See the description of the ZMF Connector plug-in on page 12.) SERNET also supervises the ZMF XML Services request/response cycle and routes requests across multiple ZMF started tasks. SERNET is automatically installed with ChangeMan ZMF.

- **Web application server** — A Web application server, such as Apache Tomcat or IBM WebSphere Application Server, services requests and shares resources among multiple Web clients. The application server may include a native HTTP server or be front-ended by an external HTTP server, as needed to optimize security and performance. It invokes the ZMF Connector plug-in to provide service connectivity between the mainframe and the Web. Any Web Services application server that supports the Apache Axis SOAP library should in principle be compatible with the ChangeMan ZMF Web Services API. You must install and configure the application server before you install the ZMF Web Services API.

- **Apache Axis SOAP engine** — The Apache eXtensible Interaction System (Axis) is an open-source SOAP message processing engine and Web Services function library based on J2EE (Java 2 Enterprise Edition). It is installed as a plug-in to the Web Services application server. In addition to a SOAP parser, the Axis library includes utilities for generating Java classes from Web Service Description Language files (WSDLs) and for monitoring TCP/IP packets. Axis is supported by many Web Services

application servers, including Apache Tomcat, BEA WebLogic, Borland Enterprise Server, JBoss, Macromedia Cold Fusion MX, Oracle Web Services Manager, and (as an alternate SOAP engine) IBM WebSphere Application Server. The ZMF Connector plug-in requires Axis. It is supplied by the ChangeMan ZMF installer and is installed on the Web Services application server platform. **==>??Question to reviewer: Not clear. Does this last sentence indicate that the ChangeMan ZMF installer supplies Axis or supplies the ZMF Connector plug-in? That is, what does the word "It" refer to??<==**

- **ZMF Connector** — The ZMF Connector plug-in is a Java servlet that manages connectivity between a Web Services application server on the network and SERNET on the mainframe. On inbound requests, it extracts ZMF XML Services payloads from their SOAP envelope, validates them, then passes them on to the chosen ZMF instance on the mainframe. On outbound replies, it encapsulates ZMF XML Services payloads within a SOAP envelope and passes them to a particular Web Services client application. The ChangeMan ZMF installer supplies the ZMF Connector plug-in and installs it on the Web Services application server platform.

- **WSDL (Web Service Description Language) files** — The ZMF Web Services API WSDLs describe the SERNET and ChangeMan ZMF services. They provide XML Schema specifications for data types and allowed values, as well as SOAP XML extensions that specify bindings of the abstract services to specific SOAP messages and Web application servers. A variety of code generators and development tools take WSDLs as input to assist Web client development. The ChangeMan ZMF installer supplies the ZMF WSDLs and installs them on the Web application server or on a separate development system, as needed.

- **Web client developer tools** — Standalone tools or integrated development environments (IDEs) and plug-ins of your choice can take WSDL specifications as input, parse the XML, and generate Java code stubs and/or interactive fill-in-the-blank Web client prototypes as output. Open-source Web Services tools are available at no charge, such as **Eclipse** (for Java development).

> **NOTES** Because the ZMF Web Services do not build in a session management mechanism, Web Services development and testing tools such as `soapUI` will not execute ZMF Web Services calls successfully from the basic API WSDLs supplied with the ChangeMan ZMF installer.
>
> You can find further information about Web Services tools for the **Eclipse** IDE online at http://www.eclipse.org/webtools/ws/.

Component Locations

Of the environment components required by the Web Services API, the ChangeMan ZMF installer supplies ChangeMan ZMF itself, the XML Services API, SERNET, and the ZMF Connector plug-in for your Web Services application server.

WAR File A Web Application Archive (WAR) file is used to distribute the Web Services WSDL files, the Apache Axis library, and a set of Java classes associated with the ZMF Connector. The file name is `zmfws.war` and is contained in a zip file named `zmfws.zip`, which you can download from the Serena Support Web site under Serena ChangeMan ZMF Client Pack.

Installation Prerequisites

Your environment must be set up to satisfy the requirements listed below before you install the ChangeMan ZMF Web Services plug-in on the Web Services application server. It is not within the scope of this document to describe how to install or configure the software listed in this section.

Web Application Server Environment

The Web Services application server may run on any desired operating system platform, including z/OS, but most commonly runs on Windows, Linux, or Solaris. On this platform, the ChangeMan ZMF Web Services API requires:

- Oracle® Corporation's J2SE™ Development Kit (JDK) version 1.5 or higher, which you can download free of charge from

  http://www.oracle.com/technetwork/java/index.html

- The ZMF Web Services API has been tested with Apache Tomcat. You can download the latest GA version free of charge from

  http://tomcat.apache.org/

IMPORTANT! The Web Services application server must support Apache Axis version 1.3. Axis version 1.3 should not be confused with Axis 2 version 1.3. The ChangeMan ZMF Web Services API does not support Axis 2.

The Axis 1.3 library is supplied in the zmfws.war file. It does not conflict with other versions of Axis installed on the server.

Web Services Development Environment

A Web Services client development tool that supports the WSDL 1.1 standard is desirable. For Java development, Eclipse is recommended.

ZMF for Eclipse Compatibility
ZMF for Eclipse (formerly WD4ZMF), a Serena plug-in that integrates the open-source Eclipse IDE with ChangeMan ZMF, uses the ChangeMan ZMF Web Services API.

If you plan to install the ChangeMan ZMF Web Services API on the same platform where the ZMF for Eclipse or an earlier ZMF Eclipse integration already resides, be sure that both ZMF for Eclipse and the Web Services API you plan to install are at the same maintenance level before you proceed.

CAUTION! If a previously installed Eclipse plug-in for ZMF uses an older version of the Web Services API than the one you are about to install, proceeding with the installation may cause ZMF for Eclipse or WD4ZMF to fail.

z/OS Server Environment

On the z/OS host, the ChangeMan ZMF 8.1 Web Services API requires the following:

- ChangeMan ZMF version 6.1.0 (or above) on every z/OS LPAR you wish to access with ZMF Web Services. (HFS and Java functionality require ZMF 7.1.0 or higher.)

==>??Reviewers: Please validate the ChangeMan ZMF versions stated above??<==

- TCP/IP connectivity between the Web Services application server and the ChangeMan ZMF server(s).

In addition, Serena recommends that you install the XMLSERV prototyping tool on your ChangeMan ZMF system to assist with debugging Web Services calls to the ZMF XML Services. (See the *ChangeMan ZMF Installation Guide* for setup instructions.)

Security Considerations

Host Security   The ChangeMan ZMF Web Services API is compatible with RACF®, CA-ACF2®, and CA-Top Secret® on the mainframe. In addition, all messages between ChangeMan ZMF on the host and the ZMF Connector plug-in on the Web Services application server are compressed.

Installing the Web Services API

NOTE Installation steps are illustrated here using Microsoft Windows XP as the client operating system and Apache Tomcat for the Web Services application server. You may use other client operating systems and Web Services application servers, as described in "Installation Prerequisites".

Uninstall Prior API Versions

Remove any previous installation of the ChangeMan ZMF Web Services API before upgrading to a new version. In particular, delete any previous ZMF Web Services WAR file and expansions of the WAR file from the `webapps` directory of your Web Services application server.

CAUTION! If you have previously installed the ZMF for Eclipse or WD4ZMF plug-ins on your Web application server, some version of the ChangeMan ZMF Web Services API resides on this system. Verify the compatibility of your installed plugin with the version of the Web Services API you plan to install before deleting the existing WAR file.

Apache Tomcat Example   If you are using Apache Tomcat on the Windows XP platform as your Web Services application server, perform the following steps:

1   Stop the Tomcat service:

   a   Go to **Start > Settings > Control Panel > Administrative Tools > Services**.

   b   Select the Apache Tomcat entry and then click **Stop** to stop the service.

2   Go to the Tomcat `webapps` directory at:

   `Program Files\Apache Software Foundation\Tomcat x.x\webapps`

3   Delete the following if they exist in the `webapps` directory:

   `zmfws` folder (and all its contents, which are expansions of the WAR file)

   `zmfws.war` file

Download the WAR File

WAR File Download

Download the ChangeMan ZMF Web Services API WAR file from the Serena Support Web site.

1 Under **My Downloads**, select **ChangeMan ZMF Client Pack** from the **Product** drop-down list.

2 Locate the desired version and click on **Z/OS** under **Available Operating Systems**.

3 Download the ChangeMan ZMF Web Services file (zmfws.zip) and save it to an appropriate directory. This zip file contains the WAR file, `zmfws.war`.

Install the Web Application Server Plug-in

WAR File Installation

To install the Web Services application server software plugin, do the following:

- Stop the Web Services application server software.

- Copy `zmfws.war` (from the directory where it was saved) to the `webapps` directory for the Web Services application server that will serve ChangeMan ZMF requests on behalf of Web Services clients.

- Restart the Web Services application server software.

Apache Tomcat Example

If you are using Apache Tomcat installed as a service on the Windows XP platform as your Web Services application server, perform the following steps:

1 Stop the Tomcat service:

   a Go to **Start > Settings > Control Panel > Administrative Tools > Services**.

   b Select the Apache Tomcat entry and then click **Stop** to stop the service.

2 Copy `zmfws.war` (from the directory where it was saved) to the Tomcat `webapps` directory at:

   ```
   Program Files\Apache Software Foundation\Tomcat x.x\webapps
   ```

3 Restart the Tomcat service:

   a Go to **Start > Settings > Control Panel > Administrative Tools > Services**

   b Select the Apache Tomcat entry and then click **Start** or **Restart** to restart the service.

   c Tomcat automatically expands the ZMF Web Services WAR file on restart. This creates the `zmfws` directory and all its contexts.

TIP If the `zmfws.jar` file is not expanded into the `zmfws` directory automatically when you start Tomcat, use Tomcat's `Activate` administrator function to expand the archive and start the ZMF Connector servlet.

Verifying the Install

To verify the API installation, use the Web application server's manager app to verify that the zmfws application is running.

Chapter 2

Concepts

Web Services API Concepts

The ChangeMan ZMF Web Services API is a Web-enabled messaging layer built over the ChangeMan ZMF XML Services API. It enables developers to create applications that access ZMF functionality over an intranet, an extranet, or the World Wide Web.

Web Services Message Flow
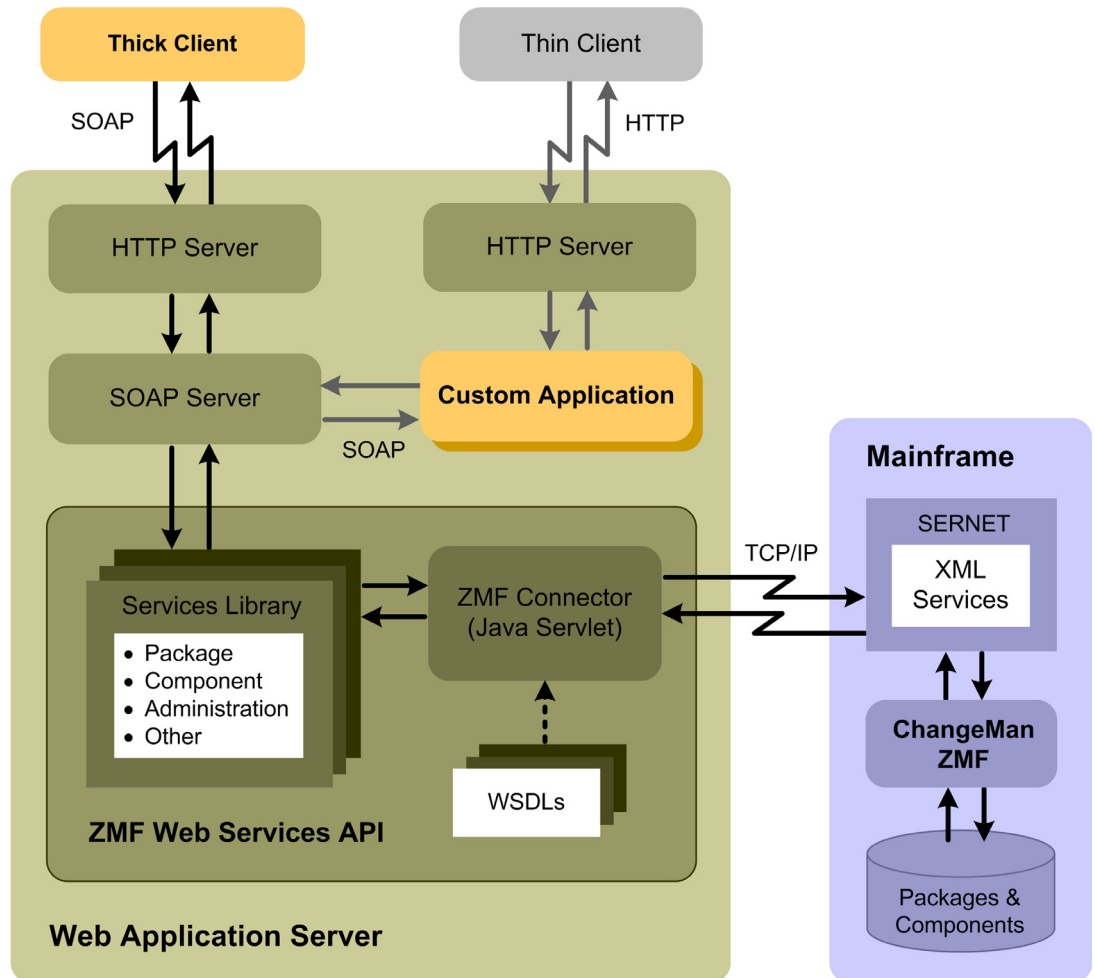
SOAP Protocol

The Web Services API works by organizing existing ChangeMan ZMF XML Services into named classes (Web Services) and methods (functions). Inbound calls to these services and methods are wrapped in an XML-defined, open-standard messaging syntax called SOAP (Simple Object Access Protocol). The SOAP message is transmitted over an HTTP connection to a Web Services application server, such as Apache Tomcat or IBM Websphere. The application server directs messages to the appropriate ZMF Web Service, which in turn invokes a server-side Java servlet that acts as a connector to ZMF.

ZMF Connector Java Servlet

The ZMF Connector servlet strips off the SOAP wrapper, then validates the data in the request against a Web Service specification defined in a Web Services Definition Language (WSDL) file. (WSDL is also an open, XML-based standard.) If the data in the request is valid, the ZMF Connector forwards it over a TCP/IP link as a straightforward XML Services request to ChangeMan ZMF. ZMF XML Services replies are received by the ZMF Connector, encapsulated in a SOAP wrapper, then forwarded by the Web application server back to the Web client.

Thick Clients and Thin Clients

The ZMF Web Services API supports both thick and thin client types.

Thick Client

A ***thick client*** is an application in which most of the custom-coded function executes locally on the client computer. This approach is preferred when computational workload can profitably be offloaded from the server to the client; when the high bandwidth demands of a rich user interface or multimedia processing can be offloaded from the network to the client; or when security considerations demand independent processing on both sides of the connection. In these kinds of situations, a ZMF Web Services application would be coded to reside on the client and send SOAP messages via HTTP to the application server where the ZMF Web Services API is hosted. In the preceding diagram, the gold box labeled "Thick Client" shows where your Web Services code would reside in a thick-client application and illustrates the kinds of Web Services message flows involved when a thick client connects to ChangeMan ZMF.

Thin Client

A ***thin client*** is an application in which most of the custom-coded function executes on a server, with the results supplied on demand to the client computer. In a Web Services context, a thin client is often simply a Web browser, which sends a standard HTTP request to the Web application server, which invokes a custom server-side application and then serves the result to the client. In the background, the custom application sends SOAP-formatted messages as needed to the ZMF Web Services API, which converts each message to XML, passes it to ZMF, receives the reply, reformats the XML reply in a SOAP wrapper, and returns it to the custom application. In the previous figure, the gold box labeled "Custom Application" shows where your Web Services code would reside in a thin-client application and illustrates the kinds of Web Services message flows involved when a thin client connects to ChangeMan ZMF.

Request/Response Cycle

Session Management

Like the ZMF XML Services transactions they are built over, ZMF Web Services transactions consist of a request message issued by the client, followed by a reply message returned by the host. However, HTML and SOAP are stateless protocols that do not automatically associate a particular reply with a particular request. It is the Web client developer's responsibility to maintain state and session information to properly associate requests and replies.

Array Results

The SOAP protocol does not support "boxcar" chaining of multiple requests or responses in a single transaction. However, several ChangeMan ZMF XML Services build in a

workaround for this problem in tasks such as searches or requests for parameter lists, which typically return a large number of identically formatted XML strings. The relevant XML Service packages such results in an array, with the number of elements in the array noted in the first array element. The ZMF Web Services API accepts such array results and passes them to the Web client in a single response message.

Supported ChangeMan ZMF Functions

Developer
Functions

All ZMF developer functions are accessible through the ZMF Web Services API. These include package creation, freeze, approval, promotion, revert, and backout; component checkout and checkin; package and component search functions; impact analysis and audit; user notifications; and others. In addition, the Web Services API supports SERNET file and dataset services for basic file manipulations such as download and upload, as well as retrieval of DDname allocations and performing ISPF file tailoring.

Administrator
Functions

ZMF administrator functions are supported in read-only mode. For example, you can retrieve a list of package parameters using the Web Services API, but you cannot change those parameters.

ERO

The Web Services API does not support ZMF Enterprise Release Option (ERO) functions.

Mapping ZMF Web Services to XML Services

ChangeMan ZMF Web Services tag names are not identical to ZMF XML Services tag names, but they do map to XML Services tag names in a systematic way.

Service Tag Names

The XML Services `<service/>`, `<scope/>`, and `<message/>` tags are general-purpose tags, each of which includes a `name` attribute specifying the ZMF functionality desired. The three tags must be used in the correct combination to specify a function, and an eight-byte length restriction on the value of the `name` attribute leads to error-prone abbreviations such as `name="CMPONENT"` for certain services.

In contrast, the Web Services API uses a single named service tag, such as `<newPackage/>` or `<checkinComponent/>`, to fully identify the functionality desired in a request. Each top-level service name maps to one and only one XML Services service.scope.message triplet.

Within the service tag, request tags are always named `<request>`. Reply messages, however, can take one of two forms, depending on whether they are standalone host status messages or if they also return data from the host. Reply messages that return only host status information are always called `<response>`. Reply messages that return data in addition to host status information have a different structure and take names specific to each service.

Within each request or reply tag are a number of parameter tags specific to the service. The names of these parameter tags are identical to their XML Services counterparts. Data types, allowed values, and dependencies are also identical to those in XML Services. Users are referred to the *ChangeMan ZMF XML Services User's Guide* for parameter tag names and allowed values. That information is not duplicated in this document.

Request Messages

ZMF Web Services request messages take the general form

```
<serviceTag>
   <request>
      <parameterTag1/>
      <parameterTag2/>
                 .
                 .
                 .
      <parameterTagn/>
   </request>
</serviceTag>
```

where

serviceTag	is a Web Services-specific tag name that maps to a specific XML Services service/scope/method tag triplet and which varies with the ZMF service requested
request	is the literal tag name of the request method
parameterTag	takes the same name and values as the corresponding XML Services parameter tag

Reply Messages

ZMF Web Services reply messages take one of two general forms, depending on whether or not data is retrieved from the host and returned to the client as part of the reply.

Standard `<response>` Message

The simplest reply format contains no returned data, but only an acknowledgement of host status as a result of the requested action. The general form of a simple reply is

```
<serviceTag>
   <response>
      <statusMessage>Some message</statusMessage>
      <statusReturnCode>nn</statusReturnCode>
      <statusReasonCode>nnnn<statusReasonCode>
   </response>
</serviceTag>
```

where

serviceTag	is a Web Services-specific tag name that maps to a specific XML Services service/scope/method tag triplet and which varies with the ZMF service requested
response	is the literal tag name of the response method
nn and *nnnn*	are numeric values returned by the host

Replies With Host Data

Reply messages that return data include the `<response>` tag above as the final element of a more complex reply. The general form of a complex reply is

```
<serviceTag>
   <serviceTagResults>
      <parameterTag1/>
      <parameterTag2/>
```

```
                    .
                    .
                    .
            <parameterTagn/>
            <response>
               <statusMessage>Some message</statusMessage>
               <statusReturnCode>nn</statusReturnCode>
               <statusReasonCode>nnnn<statusReasonCode>
            </response>
         <serviceTagResults>
      </serviceTag>
```

where

serviceTag	maps to a specific XML Services service/scope/method tag triplet and varies with the ZMF service requested
*serviceTag*Results	is the name of the response method, where the literal suffix Results is appended to the *serviceTag* name
parameterTag	takes the same name and values as the corresponding XML Services parameter tag
response	is the literal name of the standard complex tag used to return the host status resulting from the request

ZMF and SERNET Namespaces

All Web Services tag names must be qualified by a namespace prefix. Two proprietary namespaces have been defined for the ZMF Web Services:

- **ser** — This namespace defines SERNET connectivity and data transfer services.

- **cmn** — This namespace defines ChangeMan ZMF change management services.

SERNET Namespace SERNET service tag names take the following general form in production code:

    `<ser:`*tagname*`/>`

where

ser:	is the literal namespace prefix for SERNET tags
tagname	is any SERNET tag name

ChangeMan ZMF Namespace All ChangeMan ZMF developer and administrator service tag names take the following general form in production code:

    `<cmn:`*tagname*`/>`

where

cmn:	is the literal namespace prefix for ChangeMan ZMF tags
tagname	is any ChangeMan ZMF tag name

Tag syntax descriptions in this manual may omit namespace prefixes for clarity, but they are not optional in practice.

> **IMPORTANT!**   All Web Services tags must be qualified by the appropriate namespace prefix or the function call will fail. For SERNET tags, the qualified tag name format is `<ser:`*`tagname`*`/>`. For ZMF tags, the qualified tag name format is `<cmn:`*`tagname`*`/>`.

ZMF Web Services Sample Code

Sample code and instructions for running it are attached to solution number S139564 in the Serena knowledgebase.

The sample code includes 2 java source files which illustrate:

- Logging on to ZMF
- Getting a copy of the ZMF global parameters
- Getting a list of ZMF package components
- Logging off

The zip file containing the code and instructions can be downloaded using the following link:

http://knowledgebase.serena.com/InfoCenter/index?page=content&id=S139564

Chapter 3
Connectivity and Session Management

SERNET Web Services

Function Summary
Connectivity between your Web client application and ChangeMan ZMF is managed on the mainframe side by SERNET, Serena's communication and networking front-end for ChangeMan ZMF. SERNET must be running on the host before ChangeMan ZMF can start and Web Services connectivity to ChangeMan ZMF can be established.

SERNET manages Web Services logon and logoff to z/OS.

Logon Service

The SERNET logon service takes as input a valid TSO user ID and password for the user of the Web client application. It also requires an IP address for the desired host system where the ChangeMan ZMF subsystem resides and a port ID where SERNET is listening for requests to this subsystem. The subsystem ID for the desired ChangeMan ZMF instance is not required, as it is redundant with the IP address and port.

The logon service returns status information stating whether or not logon was successful.

Logon Request

`<logon>` Service Tag
A SERNET logon request contains the `<logon>` service tag as the top-level tag in the SOAP envelope body. The `<logon>` service tag encloses a `<request>` method tag, which in turn encloses a set of parameter tags.

Syntax Example
The following example logon request shows fully qualified ZMF Web Services tag names with values, wrapped in a SOAP envelope. Deprecated parameter tags are omitted.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:ser="http://wsdl.zmf.serena.com/services/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:logon>
         <ser:request>
```

```
                    <ser:host>111.222.333.444</ser:host>
                    <ser:portid>9999</ser:portid>
                    <ser:user>USER01</ser:user>
                    <ser:password>OLDPASS</ser:password>
                    <ser:newpassword>NEWPASS</ser:newpassword>
                    <ser:version>6.1.0</ser:version>
                </ser:request>
            </ser:logon>
        </soapenv:Body>
    </soapenv:Envelope>
```

`<request>` Parameter Tags

The following parameter tags are supported within the logon `<request>` tag:

Parameter Tag	Use	Occurs	Data Type & Length	Values & Dependencies
`<host>`	Required	1	String (255)	TCP/IP address of LPAR where desired ZMF subsystem resides
`<newpassword>`	Optional	0 - 1	String (8)	New password for `<user>`
`<password>`	Required	1	String (8)	Current password for `<user>`
`<portid>`	Required	1	String (4)	TCP/IP port ID for desired ZMF subsystem
`<subsystemid>`	Deprecated	1	String (1)	ZMF subsystem ID. Blank subsystem ID not supported.
`<user>`	Required	1	String (8)	TSO user ID for Web client
`<version>`	Optional	0 - 1	String (10)	ZMF API version supported by Web client. Example value for ZMF 6.1.0 patch 01: `6.1.0.01`

Logon Response

A SERNET logon response contains the `<logon>` service tag as the top-level tag in the SOAP envelope body. Within this service tag, the status of the logon action is reported in the `<response>` method tag, which in turn encloses a set of parameter tags.

Syntax Example

The logon `<response>` method tag returns a host-originated status message, return code, and reason code describing the host status of the logon request. The same syntax is used by all SERNET Web Services `<response>` methods.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:ser="http://wsdl.zmf.serena.com/services/">
    <soapenv:Header/>
    <soapenv:Body>
        <ser:logon>
            <ser:response>
                <ser:statusMessage>Logon successful</ser:statusMessage>
                <ser:statusReturnCode>00</ser:statusReturnCode>
                <ser:statusReasonCode>0000</ser:statusReasonCode>
            </ser:response>
        </ser:logon>
    </soapenv:Body>
</soapenv:Envelope>
```

**<response>
Parameter Tags**

The following parameter tags are supported within the logon `<response>` tag:

Parameter Tag	Use	Occurs	Data Type & Length	Values & Dependencies
`<statusMessage>`	Required	1	String (255)	Free-format status message returned by host or ZMF
`<statusReasonCode>`	Required	1	String (4)	Diagnostic reason code
`<statusReturnCode>`	Required	1	String (4)	Status severity code. Typical values: `00` - Normal completion `04` - Completed using default overrides for minor error conditions `08` - Some job steps failed `12` or higher - Severe error

Logoff Service

The SERNET logoff service takes as input the TSO user ID for the Web client user to be logged off the host. The logoff service returns status information stating whether or not logoff was successful.

Logoff Request

**<logoff>
Service Tag**

A SERNET logoff request contains the top-level `<logoff>` service tag, which in turn encloses a `<request>` method tag. The method tag in turn encloses parameter tags.

**Syntax
Example**

The following example logoff request shows fully qualified ZMF Web Services tag names with values, wrapped in a SOAP envelope.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:ser="http://wsdl.zmf.serena.com/services/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:logoff>
         <ser:request>
            <ser:user>USER01</ser:user>
         </ser:request>
      </ser:logoff>
   </soapenv:Body>
</soapenv:Envelope>
```

**<response>
Parameter Tags**

The following parameter tags are supported within the logoff `<request>` tag:

Parameter Tag	Use	Occurs	Data Type & Length	Values & Dependencies
`<user>`	Required	1	String (8)	TSO user ID for Web client

Logoff Response

Syntax Example

The logoff `<response>` method tag returns a host-originated status message, return code, and reason code describing the status of the host logoff request. The same syntax is used by all SERNET Web Services `<response>` methods.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:ser="http://wsdl.zmf.serena.com/services/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:logoff>
         <ser:response>
            <ser:statusMessage>Logon successful</ser:statusMessage>
            <ser:statusReturnCode>00</ser:statusReturnCode>
            <ser:statusReasonCode>0000</ser:statusReasonCode>
         </ser:response>
      </ser:logoff>
   </soapenv:Body>
</soapenv:Envelope>
```

`<response>` Parameter Tags

The following parameter tags are supported within the logoff `<response>` tag:

Parameter Tag	Use	Occurs	Data Type & Length	Values & Dependencies
statusMessage	Required	1	String (255)	Free-format status message returned by host or ZMF
statusReasonCode	Required	1	String (4)	Diagnostic reason code
statusReturnCode	Required	1	String (4)	Status severity code. Typical values: 00 - Normal completion 04 - Completed using default overrides for minor error conditions 08 - Some job steps failed 12 or higher - Severe error

Maintaining State and Managing Sessions

It is the Web client developer's responsibility to maintain state and manage sessions with the host when using the ChangeMan ZMF Web Services API. This is because ChangeMan ZMF assumes that sessions are managed externally (for example, by TSO or by ZMF for Eclipse), but HTML and SOAP are stateless protocols.

Chapter 4
Developer Services

Mapping Table Conventions

The tables in this chapter summarize the mappings between ZMF Web Services API tag names and ZMF XML Services tag names for ChangeMan ZMF developer services. The following conventions apply:

- The Web Services API service tag name shown in these tables evokes the corresponding XML Services service.code.message triplet in the leftmost column.

- All Web Services requests use the `<request>` method tag, which encloses service-specific parameter tags.

- Web Services reply messages use the method tag names shown in the table for the particular service. Method tags named `<response>` have the standard status response format. Method tags with other names enclose service-specific parameter tags, followed by the standard `<response>` tag.

- Parameter tag names are identical to those shown for the corresponding XML Service request or result in the *ChangeMan ZMF XML Services User's Guide*.

- All tag names for developer services must be qualified with the **cmn** namespace prefix.

See Chapter 2, "Concepts", for more information about ZMF Web Services tag syntax.

This chapter describes the following ChangeMan ZMF developer Web Services:

Package Search and Summary Services

The package search and summary services of ChangeMan ZMF are supported by the Web service `PackageSearchSummaryServices` and are described in the WSDL file `ZMFPackageSearchSummaryServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
package.general.search	`<queryPackages>`	`<queryPackages Results>`	Query all packages
package.limbo.search	`<queryLimboPackages>`	`<queryLimboPackages Results>`	Query limbo packages
package.service.summary	`<calculatePackage Summary>`	`<calculatePackage SummaryResults>`	Calculate package summary statistics

Package Life Cycle Services

The change package life cycle services of ChangeMan ZMF are supported by the Web service `PackageLifeCycleServices` and are described in the WSDL file `ZMFPackageLifeCycleServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
package.check.promote	`<checkPackagePromotion>`	`<response>`	Check package eligibility for promotion
package.cleanup.demote	`<demoteCleanupPackage>`	`<response>`	Clean up after package demote
package.service.approve	`<approvePackage>`	`<response>`	Approve package
package.service.backout	`<backoutPackage>`	`<response>`	Back out package
package.service.create	`<newPackage>`	`<newPackageResults>`	Create package
package.service.delete	`<deletePackage>`	`<response>`	Delete package
package.service.demote	`<demotePackage>`	`<response>`	Demote package
package.service.freeze	`<freezePackage>`	`<response>`	Freeze package
package.service.promote	`<promotePackage>`	`<response>`	Promote package
package.service.revert	`<revertPackage>`	`<response>`	Revert package
package.service.submit	`<submitPackageJCLBuild>`	`<response>`	Build JCL install job for package
schedule.service.hold	`<holdPackageInstallJob>`	`<response>`	Hold package install job
schedule.service.release	`<releasePackageInstallJob>`	`<response>`	Release package install job

Package Information Management Services

The package information management services of ChangeMan ZMF are supported by the Web service `PackageInfoManagementServices` and are described in the WSDL file `ZMFPackageInfoManagement.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
approver.pkg.list	`<getApprovers>`	`<getApproversResults>`	List package approvers & approval actions
libtype.pkg.list	`<getLibraryTypes>`	`<getLibraryTypes Results>`	List library types for package
package.aff_apls.list	`<getAffectedAppls>`	`<getAffectedAppls Results>`	List applications affected by pkg
package.gen_desc.list	`<getDescriptions>`	`<getDescriptions Results>`	List package description(s)
package.gen_parms.list	`<getGeneralParameters>`	`<getGeneralParameters Results>`	List general parameters for package
package.gen_parms.refreeze	`<refreezeGeneral Parameters>`	`<response>`	Refreeze general parameters
package.gen_parms.unfreeze	`<unfreezeGeneral Parameters>`	`<response>`	Unfreeze general parameters
package.imp_inst.list	`<getImplementation>`	`<getImplementation Results>`	List implementation instructions
package.prm_cmp.list	`<getPackageComponent PromoHistory>`	`<getPackageComponent PromoHistoryResults>`	List promotion history for pkg component
package.prm_hist.list	`<getPromotionHistory>`	`<getPromotionHistory Results>`	List promotion history for pkg
package.pkg_link.list	`<getControlLinks>`	`<getControlLinks Results>`	List linked packages
package.prt_pkgs.list	`<getParticipating Packages>`	`<getParticipating PackagesResults>`	List participating packages
package.sch_recs.list	`<getDependencies>`	`<getDependencies Results>`	List install job dependencies
package.sites.refreeze	`<refreezeInstallSites>`	`<response>`	Refreeze install sites for pkg
package.sites.unfreeze	`<unfreezeInstallSites>`	`<response>`	Unfreeze install sites for pkg
package.usr_recs.list	`<getUserVariables>`	`<getUserVariables Results>`	List user variables for pkg
schedule.service.list	`<getInstallSchedule>`	`<getInstallSchedule Results>`	List package install schedule
site.pkg.list	`<getInstallSites>`	`<getInstallSites Results>`	List install sites for package

Package Configuration Services

The package configuration management services of ChangeMan ZMF are supported by the Web service `PackageConfigurationServices` and are described in the WSDL file `ZMFPackageconfigurationServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
package.aff_appl.mcreate	`<newAffectedAppls>`	`<response>`	Add new affected applications to package
package.aff_appl.mdelete	`<deleteAffectedAppls>`	`<response>`	Delete affected application from package
package.aff_appl.update	`<updateAffectedAppls>`	`<response>`	Update affected applications for package
package.gen_desc.update	`<updateDescriptions>`	`<response>`	Update general pkg description
package.gen_parms.update	`<updateGeneral Parameters>`	`<response>`	Update general pkg parameters
package.imp_inst.update	`<updateImplementation>`	`<response>`	Update package implementation instructions
package.ims_acb.update	`<updateIMSApplControl Block>`	`<response>`	Update IMS ACB
package.ims_crgn.update	`<updateIMSControlRegion>`	`<response>`	Update IMS control region
package.prt_pkgs.mcreate	`<newParticipating Packages>`	`<response>`	Add new participating pkg
package.prt_pkgs.mdelete	`<deleteParticipating Packages>`	`<response>`	Delete participating pkg
package.prt_pkgs.update	`<updateParticipating Packages>`	`<response>`	Update participating pkg
package.sch_recs.mcreate	`<newDependencies>`	`<response>`	Add new install job dependencies
package.sch_recs.mdelete	`<deleteDependencies>`	`<response>`	Delete install job dependencies
package.sch_recs.update	`<updateDependencies>`	`<response>`	Update install job dependencies
package.usr_recs.update	`<updateUserVariables>`	`<response>`	Update user variables for pkg
schedule.service.update	`<updateInstallSchedule>`	`<response>`	Update install schedule
site.pkg.create	`<newInstallSite>`	`<response>`	Add new install site to package
site.pkg.delete	`<deleteInstallSite>`	`<response>`	Delete install site from package

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
site.pkg.update	`<updateInstallSite>`	`<response>`	Update install site for package

Package Validation Services

Validation services for packages are supported by the `PackageValidationServices` Web service and are described in the file `ZMFPackageValidationServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.pkg_lod.list	`<getSourceToLoad Dependencies>`	`<getSourceToLoad DependenciesResults>`	List source to load member dependencies
package.cmponent.integrty	`<checkComponentIntegrity>`	`<checkComponent IntegrityResults>`	Check integrity of package components
package.service.audit	`<auditPackage>`	`<auditPackageResults>`	Audit package

Package Component Management Services

The package-level component management services of ZMF are supported by the Web service `PackageComponentManagementServices` and are described in the WSDL file `ZMFPackageComponentManagementServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.pkg_comp.list	`<getComponents>`	`<getComponentsResults>`	List components in package
cmponent.pkg_comp.delete	`<deleteComponent>`	`<response>`	Delete package component
cmponent.pkg_util.list	`<getUtilityComponents>`	`<getUtilityComponents Results>`	List scratch/ rename utility records in pkg
cmponent.pkg_util.delete	`<deleteUtility Component>`	`<response>`	Delete scratch/ rename utility record from pkg
package.cmp_desc.list	`<getChangeDescriptions>`	`<getChangeDescriptions Results>`	List change descriptions for components in package
package.forms.refreeze	`<refreezeForms>`	`<response>`	Refreeze previously unfrozen forms in package

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
package.forms.unfreeze	`<unfreezeForms>`	`<response>`	Unfreeze forms in frozen package
package.non_src.refreeze	`<refreezeNonSource Component>`	`<response>`	Refreeze nonsource component in package
package.non_src.unfreeze	`<unfreezeNonSource Component>`	`<response>`	Unfreeze nonsource component in package
package.prm_ovly.list	`<getPromotionOverlays>`	`<getPromotionOverlays Results>`	List components overlaid by promotion
package.scr_ren.refreeze	`<refreezeUtility Component>`	`<response>`	Refreeze scratch/rename utility record in package
package.scr_ren.unfreeze	`<unfreezeUtility Component>`	`<response>`	Unfreeze scratch/rename utility record in package
package.src_lod.refreeze	`<refreezeSource Component>`	`<response>`	Refreeze source component in package
package.src_lod.unfreeze	`<unfreezeSource Component>`	`<response>`	Unfreeze source component in package

Component Life Cycle Services

The `ComponentLifeCycleServices` Web service supports the component life cycle services of ZMF. It is described in the file `ZMFComponentLifeCycleServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.apl_dprc.check	`<checkBuildProcedures>`	`<response>`	Check component build procedures
cmponent.apl_dprc.list	`<getBuildProcedures>`	`<getBuildProcedures Results>`	List component build procedures
cmponent.service.browse	`<browseComponent>`	`<browseComponent Results>`	Browse package component
cmponent.service.build	`<buildComponent>`	`<response>`	Build package component
cmponent.service.checkin	`<checkinComponent>`	`<response>`	Check component in to package

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.service.checkout	`<checkoutComponent>`	`<response>`	Check component out of package
cmponent.service.compare	`<compareComponents>`	`<compareComponents Results>`	Compare package components
cmponent.service.copy **CAUTION!** Misuse may cause serious errors and out-of-sync conditions.	`<copyComponents>`	`<response>`	Copy components.
cmponent.service.lock	`<lockComponent>`	`<response>`	Lock component
cmponent.service.recomp	`<recompileComponent>`	`<response>`	Recompile component
cmponent.service.relink	`<relinkComponent>`	`<response>`	Relink component
cmponent.service.rename	`<renameComponent>`	`<response>`	Rename component
cmponent.service.scratch	`<scratchComponent>`	`<response>`	Scratch component
cmponent.service.unlock	`<unlockComponent>`	`<response>`	Unlock component

Component Version Management Services

The `ComponentVersionManagementServices` Web service supports ZMF's component staging versions facility. WSDL file `ZMFComponentVersionManagementServices.wsdl` describes this service.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.ssv_ver.list	`<getStagingVersions>`	`<getStagingVersions Results>`	List staged versions of components in package
cmponent.ssv_ver.retrieve	`<retrieveStagingVersion>`	`<retrieveStagingVersion Results>`	Retrieve staged version of component

Component Information Management Services

The component information management services of ZMF are supported by the `ComponentInfoManagementServices` Web service. It is described by the WSDL file `ZMFComponentInfoManagementServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.apl_cdsc.list	`<getComponent Descriptions>`	`<getComponentDescriptions Results>`	List application component descriptions
cmponent.chg_desc.list	`<getComponentChange Descriptions>`	`<getComponentChange DescriptionsResults>`	List component change descriptions
cmponent.prm_hist.list	`<getComponent PromoHistory>`	`<getComponentPromotion HistoryResults>`	List component promotion history
cmponent.history.list	`<getHistory>`	`<getHistoryResults>`	List complete component history
cmponent.history.listbase	`<getBaselineHistory>`	`<getBaselineHistory Results>`	List history for baselined version of component
cmponent.history.listconc	`<getConcurrentUser History>`	`<getConcurrentUser HistoryResults>`	List history for components in concurrent development
cmponent.history.listcurr	`<getCurrentHistory>`	`<getCurrentHistory Results>`	List history for active components
cmponent.history.listshrt	`<getShortHistory>`	`<getShortHistoryResults>`	List short component history
cmponent.pkg_wrkl.list	`<getWorklist>`	`<getWorklistResults>`	List component user work list

Component Security Services

The ComponentSecurityServices Web service supports the component-level security features of ZMF. It is described in the ZMFComponentSecurityServices.wsdl file.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
cmponent.apl_secr.check	`<checkSecurity>`	`<response>`	Check user authorization to edit component
cmponent.apl_secr.list	`<getAuthorizedUsers>`	`<getAuthorizedUsers Results>`	List authorized users for component

Data Set Management Services

The `DatasetManagementServices` Web service supports the dataset management features of ZMF. It is described in the `ZMFDatasetManagementServices.wsdl` file.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
dss.service.allocate	`<newDataset>`	`<response>`	Allocate dataset
dss.service.delete	`<deleteDataset>`	`<response>`	Delete dataset
dss.service.info	`<getDatasetProperties>`	`<getDatasetProperties Results>`	List allocation properties of dataset
dss.service.list	`<getDatasetMembers>`	`<getDatasetMembers Results>`	List dataset members
dss.service.mbrdel	`<deleteDatasetMember>`	`<response>`	Delete dataset member
dss.ispfile.info	`<getIspfInformation>`	`<getIspfInformatio Results>`	List ISPF properties of dataset

Database Management Services

The `DatabaseManagementServices` Web service supports the DB2 and IMS database development features of ChangeMan ZMF. It is described in the WSDL file `ZMFDatabaseManagementServices.wsdl`.

DB2 Database Management Services

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
db2admin.apl_actv.list	`<getApplDB2Active Libraries>`	`<getApplDB2Active LibrariesResults>`	List active DB2 libraries for application
db2admin.apl_logl.list	`<getApplDB2Logical Subsystem>`	`<getApplDB2Logical SubsystemResults>`	List DB2 logical subsystem for application
db2admin.gbl_logl.list	`<getGlobalDB2Logical Subsystem>`	`<getGlobalDB2Logical SubsystemResults>`	List global DB2 logical subsystem
db2admin.gbl_phys.list	`<getGlobalDB2Physical Subsystem>`	`<getGlobalDB2Physical SubsystemResults>`	List global DB2 physical subsystem

IMS Database Management Services

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
imsovrd.apl_dbd.list	`<getApplDBDOverrides>`	`<getApplDBDOverrides Results>`	List IMS DBD overrides for application
imsovrd.apl_psb.list	`<getApplPSBOverrides>`	`<getApplPSBOverrides Results>`	List IMS PSB overrides for application
imsovrd.gbl_dbd.list	`<getGlobalDBDOverrides>`	`<getGlobalDBDOverrides Results>`	List global IMS DBD overrides
imsovrd.gbl_psb.list	`<getGlobalPSBOverrides>`	`<getGlobalPSBOverrides Results>`	List global IMS PSB overrides
imsovrd.pkg_dbd.list	`<getPackageDBDOverrides>`	`<getPackageDBD OverridesResults>`	List IMS DBD overrides for package
imsovrd.pkg_psb.list	`<getPackagePSBOverrides>`	`<getPackagePSB OverridesResults>`	List IMS PSB overrides for package
package.ims_acb.list	`<getPackageACBBuild Statements>`	`<getPackageACBBuild StatementsResults>`	List IMS ACB build statements for package
package.ims_crgn.list	`<getPackageControl Region>`	`<getPackageControl RegionResults>`	List IMS control region for package

Impact Analysis Services

The `ImpactAnalysisServices` Web service supports the impact analysis features of ZMF. It is described in the `ZMFImpactAnalysisServices.wsdl` file.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
impact.bun.list	`<getImpactBunList>`	`<getImpactBunList Results>`	List information about application/ libtype/baseline correlations
impact.component.list	`<getImpactComponentList>`	`<getImpactComponent ListResults>`	List information in I/A table about baselined components
impact.table.list	`<getImpactTableList>`	`<getImpactTableList Results>`	List information about I/A relationships

Schedule Services

The `ScheduleServices` Web service supports the scheduling features of ZMF. It is described in the `ZMFScheduleServices.wsdl` file.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
schedule.service.delete	`<deleteSchedule>`	`<deleteSchedule Results>`	Delete scheduler records
schedule.service.list	`<getSchedule>`	`<getScheduleResults>`	List scheduler records for package
schedule.service.hold	`<holdSchedule>`	`<holdScheduleResults>`	Hold package installation job
schedule.service.release	`<releaseSchedule>`	`<releaseSchedule Results>`	Release previously held package installation job
schedule.service.update	`<updateSchedule>`	`<updateSchedule Results>`	Update scheduler records

Chapter 5
Administrator Services

Mapping Table Conventions

The tables in this chapter summarize the mappings between ZMF Web Services API tag names and ZMF XML Services tag names for ChangeMan ZMF administrator services. The following conventions apply:

- The Web Services API service tag name shown in these tables evokes the corresponding XML Services service.code.message triplet in the leftmost column.

- All Web Services requests use the `<request>` method tag, which encloses service-specific parameter tags.

- Web Services reply messages use the method tag names shown in the table for the particular service. Method tags named `<response>` have the standard status response format. Method tags with other names enclose service-specific parameter tags, followed by the standard `<response>` tag.

- Parameter tag names are identical to those shown for the corresponding XML Service request or result in the *ChangeMan ZMF XML Services User's Guide*.

- All tag names for ChangeMan ZMF administrator services must be qualified with the `cmn` namespace prefix.

See Chapter 2, "Concepts", for more information about ZMF Web Services tag syntax.

This chapter describes the following ChangeMan ZMF administrator Web Services:

Approver Notifications

The `ApproverNotificationAdminServices` Web service supports the user notification features of ZMF in read and execute modes. Updates and deletions are not allowed from a Web client. The WSDL file `ZMFApproverNotificationAdminServices.wsdl` describes this service.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
approver.apl.list	`<getApplApprovers>`	`<getApplApproversResults>`	List application approvers

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
notyfile.service.download	`<downloadNotification File>`	`<downloadNotification FileResults>`	Download approver notification file
user.service.notify	`<notifyUser>`	`<response>`	Send user notification

Change Library Administration

The `ChangeLibraryAdminServices` Web service provides read-only access to the change library administration features of ZMF. The service is described in the WSDL file `ZMFChangeLibraryAdminServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
baselib.service.list	`<getBaselineLibraries>`	`<getBaselineLibraries Results>`	List baseline libraries
promlib.library.list	`<getPromotionLibraries>`	`<getPromotionLibraries Results>`	List promotion libraries
prodlib.service.list	`<getProductionLibraries>`	`<getProductionLibraries Results>`	List production libraries

Developer Environment Administration

The `DeveloperEnvironmentAdminServices` Web service provides access to the developer environment administration features of ZMF. Most functions are read-only. However, build parameters may be updated at the application and global level from a Web client. The service is described in the `ZMFSiteAdminServices.wsdl` file.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
language.apl.list	`<getApplLanguageParsers>`	`<getApplLanguageParsers Results>`	List language parsers for application
language.gbl.list	`<getGlobalLanguage Parsers>`	`<getGlobalLanguage ParsersResults>`	List global language parsers
libtype.apl.list	`<getApplLibraryTypes>`	`<getApplLibraryTypes Results>`	List application library types
libtype.gbl.list	`<getGlobalLibraryTypes>`	`<getGlobalLibraryTypes Results>`	List global library types
parms.apl.list	`<getApplParms>`	`<getApplParmsResults>`	List build parameters for application

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
parms.apl.update	`<updateApplParms>`	`<response>`	Update build procedures for application
parms.gbl.list	`<getGlobalParms>`	`<getGlobalParmsResults>`	List global build parameters
parms.gbl.update	`<updateGlobalParms>`	`<response>`	Update global build procedures
procs.apl.list	`<getApplBuildProcedures>`	`<getApplBuildProcedures Results>`	List application build procedures
procs.gbl.list	`<getGlobalBuild Procedures>`	`<getGlobalBuild ProceduresResults>`	List global build procedures
reasons.service.list	`<getReasonCodes>`	`<getReasonCodesResults>`	List reason codes for approval actions

Install Site Administration

The `SiteAdminServices` Web service provides read-only access to the install site administration features of ZMF. It is described in `ZMFSiteAdminServices.wsdl`.

XML Services Service.Scope.Message	Web Services API Service Tag	Reply Method Tag	ZMF Function
calendar.service.list	`<getSiteInstallCalendar>`	`<getSiteInstallCalendar Results>`	List site install calendar
site.apl.list	`<getApplSites>`	`<getApplicationSites Results>`	List install sites for application
site.gbl.list	`<getGlobalSites>`	`<getGlobalSitesResults>`	List global install sites
promlib.site.list	`<getPromlibSites>`	`<getPromlibSites Results>`	List install sites for promotion libraries

Index

Symbols

<logoff> tag 25
<logon> tag 23
<response> message 20

A

Adobe Acrobat 8
affected applications
    add to package 30
    delete from package 30
    update for package 30
Apache Axis 6, 11
Apache Tomcat
    downloads 13
    examples 14, 15
APIs for ZMF
    Web Services 17, 18
    XML Services 17, 18
application services
    check component build procedures 32
    list approvers 39
    list build parameters 40
    list build procedures 41
    list component build procedures 32
    list component descriptions 34
    list install sites 41
    list language parsers 40
    list library types 40
    update build procedures 41
approval services
    described 39
    list application approvers 39
    list reason codes 41
array results 18

B

build services
    list application build parameters 40
    list application build procedures 41
    list global build parameters 41
    list global build procedures 41
    update build procedures for application 41
    update global build procedures 41

C

change library administration 40
code samples 6, 22
compatibility 13
component security services 34
component services
    browse 32
    build 32
    check build procedures 32
    check in 32
    check out 33
    check user authorization 34
    compare 33
    copy 33
    delete from package 31
    delete scratch/rename utility record 31
    list application-level component descriptions
            34
    list authorized users 34
    list baseline history 34
    list build procedures 32
    list change descriptions 34
    list change descriptions for package 31
    list complete history 34
    list for package 31
    list history for active components 34
    list history for components in concurrent
            development 34
    list promotion history 34
    list promotion overlays 32
    list scratch/rename utility records 31
    list short history 34
    list staged versions 33
    list user worklist 34
    lock 33
    recompile 33
    refreeze forms 31
    refreeze nonsource 32
    refreeze scratch/rename utility records 32
    refreeze source 32
    relink 33
    rename 33
    retrieve staged version 33
    scratch 33
    unfreeze forms 32
    unfreeze nonsource 32
    unfreeze scratch/rename utility records 32
    unfreeze source 32
    unlock 33

R

reply messages 20
request messages 20
request/reponse message cycle 18

S

schedule services 37
    delete schedule 37
    hold schedule 37
    list schedule 37
    release schedule 37
    update schedule 37
security considerations 14
security services
    check user authorization for component 34
    list authorized users for component 34
SERNET 11
Service Oriented Architecture (SOA) 5, 6
site administration 41
SOAP (Simple Object Access Protocol) 6, 18
SOAP engine 6, 11

T

tag names 19
TCP/IP 18
thick client 18
thin client 18

W

WAR file 12, 15
WD4ZMF Eclipse plugin 13, 14
Web application archive (WAR) file 12
Web application server 11, 13
Web archive file (WAR) 15
Web Services API
    client types 18
    components 11
    described 17
    message flow 17
    methods 18
    overview 5
    protocols 18
    request/response message cycle 18
    services 18
    ZMF functions supported 19
Web Services Basic Profile 6
Web Services client development tools 13
Web Services Definition Language (WSDL) 18
WSDL (Web Service Description Language) 6, 12

X

XML Services API 11, 17, 18

Z

ZMF connector servlet 12, 18
ZMF for Eclipse plugin 13, 14