SERENA™

SERENA®

MAINFRAME CONNECTOR

Mainframe Connector Guide

Serena Proprietary and Confidential Information

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

License and copyright information for 3rd party software included in this release can be found on the product's news page at http://support.serena.com/ProductNews/default.aspx and may also be found as part of the software download available at http://www.support.serena.com.

Trademarks

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

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1850 Gateway Drive, 4th Floor, San Mateo, CA 94404.

Part number: Product version: 5.1.5

Publication date: 2015-02-11

Table of Contents

Chapter 1: Overview

Mainframe Connector is a flexible set of services and programs that can be used from Windows or UNIX/Linux platforms to deploy to the mainframe through Serena ChangeMan ZMF. Built-in integrations are provided with Release Manager and Deployment Automation, and a number of features are provided for custom integration with SBM.

Mainframe Connector can be used to retrieve a list of change packages for selection from ChangeMan ZMF that are suitable for deployment, and then perform a number of potential actions using the default ChangeMan ZMF functionality. Mainframe Connector uses web services and passes predefined credentials and selection information for ChangeMan ZMF applications, change packages of various types, and so on.

The integration of Release Manager version 5.1 and 5.1.1 with ChangeMan ZMF using the Mainframe Connector is described in this guide, focusing on the mainframe configuration needed and the configuration of supporting processes in Deployment Automation. Other versions of Release Manager may have different variations of the integration.

The mainframe configuration instructions are valid whether you use Release Manager with Deployment Automation or connect to ChangeMan ZMF through your Release Manager or a custom SBM integration. The details on the configuration needed for Release Manager may vary, depending on whether the integration goes directly from the SBM process apps or leverages the provided Deployment Automation plug-ins.

If you are unsure which integration you have, please consult Release Manager documentation for your version of Release Manager. If you are creating a custom integration, you can use this guide to understand the integration and make adjustments as needed for your specific implementation.

Documentation References

- For an overview of the integration, details on configuring the mainframe portion of Mainframe Connector, and information on how to configure the plug-in, see this guide.
- For details on initiating deployment from Release Manager, see the Release Manager documentation.
- For more information on using the Deployment Automation ChangeMan ZMF plug-in, including details on the plug-in steps, see the *Serena Deployment Automation Plugins Guide*.

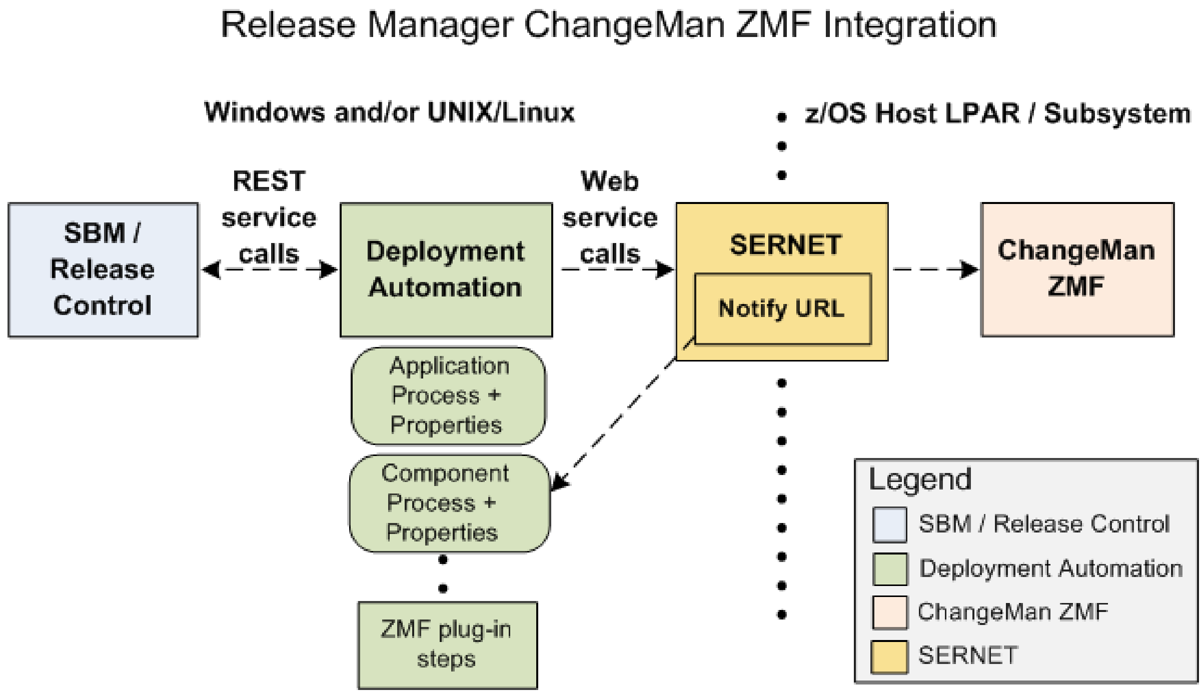
  - **Note:** The `.mtd` files included in the ChangeMan ZMF plug-in bundle file are used for custom solutions only.

The following topics describe the default integration of Release Manager version 5.1 and 5.1.1 with ChangeMan ZMF using the Mainframe Connector.

ChangeMan ZMF Integration Communication Overview

You must configure ChangeMan ZMF communication on the z/OS mainframe and on the Deployment Automation server before you use the ChangeMan ZMF plug-in. The rest of the information needed for Deployment Automation to communicate with ChangeMan ZMF is embedded in the Deployment Automation application and component processes, which use the ChangeMan ZMF plug-in.

The flow of communication between Release Manager, Deployment Automation, and ChangeMan ZMF is shown in the following figure.



ChangeMan ZMF Integration Runtime Communication

The communication between Release Manager, Deployment Automation, and ChangeMan ZMF proceeds as follows:

1. A Release Manager deployment task initiates a request to Deployment Automation for an existing ChangeMan ZMF process through RESTful services.
2. Deployment Automation invokes the ChangeMan ZMF process, which uses a trusted user ID to logon on behalf of a designated username, typically the Deployment Automation user name, to initiate the requested operations in ChangeMan ZMF.
3. When the ChangeMan ZMF operations complete, an event is sent from the SERNET NTFYURL and the Deployment Automation listener detects it. When the operations are complete, Deployment Automation retrieves ChangeMan ZMF information through the listener and puts it in the Deployment Automation execution log.

4. The Deployment Automation process is updated with the completion status and the component process step is flagged as successful or failed.

5. Deployment Automation sends the requested information to Release Manager through REST service calls.

Chapter 2: Configuring ChangeMan ZMF Communication on the Mainframe

The following topics describe how to configure the ChangeMan ZMF communication on the z/OS mainframe that is required to use the ChangeMan ZMF plug-in for Deployment Automation.

- Configuring the Notification URL [page 9]
- Configuring a ChangeMan ZMF Proxy User ID [page 9]
- Configuring ChangeMan ZMF Approvers [page 10]
- Configuring Objects in ChangeMan ZMF [page 11]

Configuring the Notification URL

The NTFYURL parameter is a keyword option used with the SERNET started task. This parameter is required for notifying Deployment Automation when an event is emitted from ChangeMan ZMF.

There are three different ways of passing the NTFYURL parameter to SERNET. Documentation on passing parameters to SERNET is in the *Passing Parameters to SERNET* in the *Serena ChangeMan ZMF Installation Guide*.

Whichever method you choose to pass this parameter, it must be specified as follows:

```
NTFYURL=’hostname:port/serena_ra/services/ZMFALFEventRouter’
```

where `hostname` is the server name where Deployment Automation Web services are installed and `port` is the port number for that server.

This parameter is case-sensitive; the non-variable text must be entered exactly as shown. Be sure to include the quotes around the variable string.

Example

```
NTFYURL='alm_host:8080/serena_ra/services/ZMFALFEventRouter'
```

CAUTION:

If your site is a DP site, you must specify the same hostname and port for the DP site and the P site. If not, the P site will continue to wake up looking for work and will fill up the JESMSGLG (JES message log).

Configuring a ChangeMan ZMF Proxy User ID

A proxy user ID, or trusted user ID, is required for each ChangeMan ZMF host server, or LPAR.

The purpose of the proxy ID is to allow users to automatically access ChangeMan ZMF through the ChangeMan ZMF plug-in. The trusted user ID connects to the host server on behalf of a designated user, such as the Deployment Automation user, whose permissions are then used for the ChangeMan ZMF access.

Configure the proxy ID as follows:

- The proxy ID can be any SAF-defined user ID.
- You must add an OMVS segment for the ZMF proxy ID since this functionality accesses HFS library types.
- No specific attributes are required.
- It is not necessary that this user ID be allowed to access TSO.
- This user ID must be given READ or higher access to the "trusted resource".

The trusted resource is a SAF resource, by default SERENA.SERNET.AUTHUSR in the FACILITY class. The resource and class are user-modifiable by changing the names in the SERLCSEC CSECT, which is delivered as source code with ChangeMan ZMF. This CSECT is used for customizing a variety of security-related functions.

NOTE It is not necessary to alter SERLCSEC to support Deployment Automation in the default fashion, as the latest version is already coded for the above resource name and class. Be sure to use the latest version of this CSECT. If you have previously modified it, you will need to re-apply your customizations.

IMPORTANT! The Deployment Automation *trusted resource* is not related to the RACF user ID TRUSTED attribute.

Configuring ChangeMan ZMF Approvers

You must configure approvers in ChangeMan ZMF as needed to initiate the installation and baselining of change packages.

For example, if you configure just one approver, the following happens:

- Once the approval is given by the approver, the change package will go into APR status and the installation will be initiated.
- The related ChangeMan ZMF approval deployment task will go into the `In Progress` state and will wait for the ChangeMan ZMF change package to be installed and baselined.
- After the ChangeMan ZMF change package goes into BAS status, the related ChangeMan ZMF approval deployment task will go into the `Complete` state.
- If your approval tasks are initiated from Deployment Automation, see the *Serena Release Manager Getting Started Guide* for more information on ChangeMan ZMF approvals.

NOTE In ChangeMan ZMF, specify approvers on the Planned Approvals panel in option **A.A.5**. Approvers must have approval authority.

Configuring Objects in ChangeMan ZMF

Deployment Automation uses several objects that your ChangeMan ZMF administrators configure as part of the ongoing use of ChangeMan ZMF in addition to some that must be configured specifically to support Deployment Automation.

Objects that you will ordinarily already have set up as part of your normal administration and use of ChangeMan ZMF are as follows:

- Applications (Projects)
- Sites (Environments)
- Change Packages
- Approver lists
- Promotion levels
- Audit return code rules

ChangeMan ZMF administrators should change ChangeMan ZMF objects in ChangeMan ZMF according to the ChangeMan ZMF documentation.

Chapter 3: Configuring ChangeMan ZMF Processes in Deployment Automation

The following topics describe how to configure the processes and properties for optimal use of the ChangeMan ZMF plug-in for use with Deployment Automation.

Configuring the ChangeMan ZMF Plug-in

The ChangeMan ZMF plug-in must be configured in the application server before it can be loaded into Deployment Automation.

Important: This plug-in is supported only in a Tomcat application server.

Configure the plug-in as follows:

1. Download the ChangeMan ZMF plug-in bundle file from the Release Manager download location on the Serena website at http://www.serena.com/support. For example, `ZMF_bundle_vvv`, where vvv is the version.

   Note: The bundle appears in the download list only if you have purchased the Mainframe Connector option.

2. Extract the plug-in bundle file. It contains the `zmf-plugin.zip` file and all of the files needed to configure the plug-in.

3. Stop the Deployment Automation application server and deploy the following files to the Deployment Automation deployment location, the actual Deployment Automation application server location where `serena_ra.war` is deployed:

   `almzmf.war`

   `almzmfalf.war`

   `almzmfws.war`

4. Copy `zmf-core-CURRENT.jar` to the following application server path:

   `..\webapps\serena_ra\WEB-INF\lib`

5. Add the following lines to the application server `web.xml` file.

   Note: If you are initiating the ChangeMan ZMF processes directly through Deployment Automation and not through Release Manager, these lines should not be added; you should comment them out if they are already there.

```
<servlet>
      <servlet-name>ZMFALFEventRouter</servlet-name>
      <servlet-class>com.serena.servlet.ZMFALFEventRouter</servlet-class>
      <init-param>
          <param-name>redirectURL</param-name>
          <param-value>/</param-value>
      </init-param>
      <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
      <servlet-name>ZMFALFEventRouter</servlet-name>
      <url-pattern>/servlet/ZMFALFEventRouter</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
      <servlet-name>ZMFALFEventRouter</servlet-name>
      <url-pattern>/services/*</url-pattern>
  </servlet-mapping>
```

6. Restart the application server.

7. After the application server is started, load the `zmf-plugin.zip` file in Deployment Automation through **Administration > Automation > Automation Plugins**.

8. Configure the processes for the plug-in that are required for the Release Manager integration.

Importing the Sample ChangeMan ZMF Environment

For the quickest and most reliable implementation, you should import the sample environment and application and modify the properties to suit your needs. Before you can import the sample application processes, you must first import the environment that is associated with the application.

One sample environment, UAT, is provided to use with all of the sample applications. If you have already imported the environment to use with one of the other sample applications, you should not import it again.

To import the sample UAT environment:

1. Navigate to the directory location where you downloaded the plug-in bundle.

2. Extract the following JSON file if it is not already extracted:

   `Sample UAT.json`

3. If you want to change the name of the environment that will be imported, open the environment JSON file and change the name and description to whatever you want to call your environment.

4. To import the environment:

   a. In Deployment Automation, navigate to **Management > Environments**.

   b. Click **Import Environment** and browse to the path of the `Sample UAT.json` file.

   c. Click **Submit**.

The environment should now be listed in your environments page, as UAT if you did not change the name, or under the name you specified when you changed it.

Importing the Sample ChangeMan ZMF Application

There are several Deployment Automation processes necessary to create the operations needed for this plug-in integration. To make it easier for you to configure your processes, an exported sample application is included in the plug-in bundle.

The sample application includes all of the application and component information needed to get you started. You can import the exported file and modify the details to match your implementation. Otherwise, you must configure all of your processes and properties manually as described in the subsequent topics.

To import the sample application:

1. Navigate to the directory location where you downloaded the plug-in bundle.

2. Extract the following JSON file if it is not already extracted:

   `<product> Sample Application.json`

3. If you changed the name and description of the sample environment that you imported, open the application JSON file and change the corresponding environment name and description to the the ones you used in your environment JSON file.

4. To import the application processes:

   a. In Deployment Automation, navigate to **Management > Applications**.

   b. Click **Import Application** and browse to the path of the JSON file.

   c. Click **Submit**.

The application should now appear in the application list.

Configuring the ChangeMan ZMF Application

An application process is used to run the component processes you need. Most of the properties that are needed for the component processes should be set at the application level, because many properties are used by more than one component process.

When the application process is initiated through Release Manager deployment tasks, a small set of values are passed as part of the request. Any properties with no values passed will use the values set in the application. If no values are set, the default values will be used.

Tip: For the quickest implementation, import the sample environment and application and modify the properties to suit your needs.

To configure the application:

1. Create an application that will contain your properties and component processes or select an existing one. For example, ZMF Application.

2. If you imported the sample application, edit the application and change the application name and description to match your implementation's values.

3. Add properties to your application that are common to all component processes, or modify the existing imported values to match your system information. For example, the connection information property values are as follows:

   ZMF_HOST_NAME, value <your host name>

   ZMF_HOST_PORTID, value <your port #>

   ZMF_HOST_PROXYID, value <your proxy ID>

   ZMF_HOST_PROXY_PASSWORD, value <your proxy ID password>

   ZMF_JOB_CARDS, value <your job card>

   Note: Values must be entered in uppercase. If the proxy ID is not uppercase, the connection will fail.

4. Add the following processes to your application if they have not already been imported. The process names and descriptions for the application processes needed for the Release Manager integration are as follows.

   - Approve Package - Initial approval of a package
   - Approve Package for PROD - This is the last approval, which causes the package to go to APR status and initiates the installation process; this process waits for the event in ChangeMan ZMF that confirms that the package actually installed in production.
   - Audit Package - Submit an audit package job
   - Backout Package - Back out a package from production
   - Deploy Package - Promote a package
   - Freeze Package - Freeze a package
   - Get Applications - Get applications
   - Get Approvers - Get package approvers
   - Get Installation Sites - Get package installation sites
   - Get Packages - Get packages in DEV or FRZ status

- Get Packages1 - Get packages in DEV status
- Get Packages2 - Get packages in FRZ status
- Get Packages3 - Get packages in FRZ, BAK, or BAS status
- Get Packages4 - Get packages in INS, BAK, or BAS status
- Get Production Sites - Get production sites
- Get Promotion Areas - Get promotion areas
- Get Promotion Sites - Get promotion sites
- Get Sites - Get sites
- Reject Package - Reject a package
- Revert Package - Revert a package back to development
- Undeploy Package - Demote a package

5. Add the properties to the application processes that the component processes will inherit, or change them in the imported application. The properties for which values will be passed to and from Release Manager have an RM$ at the beginning of the value, as shown in the following properties for Approve Package.

    - name ZMF_PACKAGE_NAME, label Package Name: value `RM${applicationProcess:Get Packages2;properties:{[name:ZMF_PACKAGE_NAME_FILTER,value: ZMF_PACKAGE_NAME]};displaycols:package, packageStatusDesc, packageTitle}`

    - name ZMF_PACKAGE_APPROVER, label Package Approver: value `RM${applicationProcess:Get Approvers;properties:{[name:ZMF_PACKAGE_NAME, value:ZMF_PACKAGE_NAME]};displaycols:approverEntity, approverDesc}`

Configuring ChangeMan ZMF Component Processes

Component processes are used to combine the ChangeMan ZMF plug-in steps into the processes needed to execute the set of ChangeMan ZMF operations you need.

Note: The ChangeMan ZMF plug-in must be loaded and available before you design a component process.

Tip: For the quickest implementation, import the sample environment and application and modify the properties to suit your needs.

To configure the ChangeMan ZMF component processes:

1. Create a component that will contain your component processes or select an existing one. For example, ZMF Components.

2. Add the following processes to your component if they have not already been imported.

- Approve Package
- Audit Package
- Backout Package
- Deploy Package
- Freeze Package
- Get Applications
- Get Approvers
- Get Installation Sites
- Get Packages
- Get Promotion Areas
- Get Sites
- Reject Package
- Revert Package
- Undeploy Package

3. Specify values or variables for each component process step property that will not be set by application properties.

4. Ensure that any properties that will be passed from the application processes are set to `Set a value here` so that those property values will be replaced with the application properties passed to them.

Chapter 4: Connecting Directly from an SBM Solution

If you are not using the built-in integration provided with Release Manager, but instead are integrating directly from your own custom SBM process apps, you can use the `.mtd` files provided in the Mainframe Connector bundle as examples.

By default, the Release Manager solution connects to ChangeMan ZMF through the Deployment Automation integration. This connectivity is provided through the Mainframe Connector for Release Manager, a separate option for Release Manager.

The Mainframe Connector ChangeMan ZMF plug-in bundle includes orchestration event definitions that can optionally be used for custom Release Manager or SBM solutions. These allow connectivity to ChangeMan ZMF directly from the SBM solution without routing through Deployment Automation.

The orchestration event definition files are:

`ZMFPackageAlfEvents.mtd`

and

`ZMFReleaseAlfEvents.mtd`

For information on importing and using event definition files, see the *SBM Composer Guide*.

Important:

If you use Release Manager with the provided Deployment Automation integration, you do not need to use these extra `.mtd` files for the connectivity to ChangeMan ZMF.