Oracle® DIVA Enterprise Connect

Installation, Configuration, and Administration Guide

Release 1.0.1

E85585-05

June 2018

ORACLE®

Oracle DIVA Enterprise Connect Installation, Configuration, and Administration Guide, Release 1.0.1

E85585-05

Contents

Preface

This document describes installation, configuration, deployment, and administration of the Oracle DIVA Enterprise Connect module required for DIVArchive to communicate with an application implementing the DIVA Web Services API.

See the *Oracle DIVA Enterprise Connect Web Services API Programmer's Guide* in the *Oracle DIVArchive Enterprise Connect documentation* library for a high-level introduction to the SOAP and REST services, and details for all of the calls.

Audience

This document is intended for DIVA Enterprise Connect Installers and System Administrators.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the Oracle DIVArchive documentation set for this release located at https://docs.oracle.com/en/storage/#csm.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

This chapter describes an overview of the Oracle DIVA Enterprise Connect and includes the following information:

Oracle DIVA Enterprise Connect Overview

Oracle DIVA Enterprise Connect is a standards-based Web Service API implemented on the Oracle WebLogic Suite. DIVA Enterprise Connect interacts with Oracle DIVArchive and Oracle DIVAnet systems, acting as a web service binding for the DIVArchive API. DIVA Enterprise Connect provides a client computer with a language and operating system independent method of submitting requests to archive, restore, copy, or delete content. It allows clients to gather information about archived objects and media, and manage archive devices (for example, tape libraries and disk arrays).

The DIVA Enterprise Connect platform consists of Oracle WebLogic, installation and administration scripts, data files, and configuration. The installation scripts assist in installing and configuring the DIVA Enterprise Connect platform.

Oracle DIVArchive and Oracle DIVAnet

Oracle DIVArchive is a Content Management Solution specifically engineered for archiving, tracking, and restoring large amounts of rich media and broadcast digital assets. It integrates with a multitude of industry standard software, archive media, transfer protocols, hardware platforms, and the Cloud. Oracle DIVArchive enables organizations to manage the lifecycle of their digital assets through automated policies that move data to the appropriate tier of storage based on access patterns, age, and storage utilization.

Oracle DIVAnet provides a unified view of archived digital assets across multiple, distributed DIVArchive systems, and the Cloud. It facilitates moving content back and forth among DIVArchive sites, and to and from customer Source/Destination servers and disks. DIVAnet performs tasks for disaster recovery, content distribution, access control, performance, and content availability.

By connecting to multiple DIVArchive sites, DIVAnet creates a virtual archive system that spans geographical locations.

Understanding Oracle DIVA Enterprise Connect

Oracle DIVA Enterprise Connect provides web services that allow client applications to submit requests to a remote Oracle DIVArchive or Oracle DIVAnet system. Web services are software modules that you can invoke remotely using a standard messaging format; in this case, XML. The client applications can use standard internet protocols (for example, HTTP and URLs).

The Web Services API enables a remote client to archive, restore, copy, and delete digital assets. The API also provides services to list archive objects, list archive device status (tape libraries and disks), and obtain the status of requests issued to DIVArchive. SOAP and REST bindings for the API are provided with DIVA Enterprise Connect.

When a caller sends a request to DIVA Enterprise Connect, it forwards the request to either DIVArchive or DIVAnet. The response returns as an XML document back to the caller, in either REST or SOAP format.

The Web Services use basic authentication by default. Oracle recommends using basic authentication with an SSL/TLS transport (the default is port 9444 in Web Services), and also recommends assigning credentials using the API Key Generator. For more information, see *"Authentication" on page 5-3*.

See the *Oracle DIVA Enterprise Connect Web Services API Programmer's Guide* in the *Oracle DIVA Enterprise Connect documentation* library for detailed API information.

Oracle DIVA Enterprise Connect Release Compatibility

This release of the DIVA Enterprise Connect is compatible with DIVArchive 7.4 and later (Oracle recommends DIVArchive 7.5 and later), and DIVAnet 2.1 and later. DIVAnet 2.2 and later is required for the HTTP-based transport to the *DIVAnet ManagerAdapter*.

The current DIVA Enterprise Connect software release is 1.0.1. DIVA Enterprise Connect 1.0+ supports the 2.2 version of the DIVA Web Services API. Earlier API versions are not supported on release 1.0. This is true for both SOAP and REST. Enterprise Connect 1.0.1 introduces a Windows version, in addition to the Linux version.

The Web Services 2.2 API contains the same basic information elements as the 2.1 version, but applications written against version 2.1 may not be compatible with version 2.2. There are some minor changes from 2.1 (for example, different return code behavior, and slightly different naming for namespaces in XML responses). Other modifications include changes in service authentication, HTTP return codes, HTTP header fields that are now strictly required (for example, `Content-Type`), a new 2.2 WSDL document, and new security restrictions. The SOAP 1.2 bindings are no longer supported.

Some deviations from 2.1 will require more effort to remedy. For example, 2.2 no longer supports passing parameters in the URL, and the URL format has changed slightly from the 2.1 release.

2

Installing DIVA Enterprise Connect

> **Note:** You must execute the steps in each section in this chapter sequentially as listed.

Initial Installation

You can use the DIVA Enterprise Connect installation package to install WebLogic 12*c*, Oracle Java 8 JDK, and the DIVA Enterprise Connect services. Starting with 1.0.1, there are two editions of Oracle DIVA Enterprise Connect -- a Windows version, as well as a Linux version that runs on Oracle Linux 7. There are two distinct installation packages for each operating system.

The installation requires that you create two main directories as follows:

- Staging Directory (`%STAGING_DIR%` for Windows, `$STAGING_DIR` for Linux) — A temporary directory the install process uses.
- Home Directory (`%DIVAS_HOME%` for Windows, `$DIVAS_HOME` for Linux) — The top-level directory where the WebLogic and Web Services are installed.

> **Note:** The `STAGING_DIR` and `DIVAS_HOME` variables are not actual environment variables. They only appear in this document to represent the values chosen by the user.

Running the installer package installs Oracle Weblogic, configures the DivaServices domain/servers, and deploys the DivaServices ear file. You must then create an API key, configure the connection to DIVArchive, and then start the WebLogic server. The following sections describe this process.

To install multiple instances of Enterprise Connect on the same machine (for instance, if two DIVArchive systems must be provisioned separately), insure that each instance is configured in its own WebLogic Domain, with unique, non-conflicting admin and web service ports.

Oracle Linux Installation

Task 1 Unzip the Release Tarball

1. Open a command terminal.
2. Create a temporary staging directory (referenced as `$STAGING_DIR` in this procedure).
3. Copy the DIVArchive Web Services installation package in the staging directory.
4. Change to the staging directory and execute the following command:

   ```
   tar -xvzf diva-ec-platform.1.0.0-dws.{release}.bin.tar.gz
   ```

   Where: `{release}` is the specific DIVArchive Web Services release number.

Task 2 Change Installation Properties (optional)

By default, the WebLogic and Java 8 packages are installed to `/home/diva/divas`. The default administrator user name is `weblogic`. If you do not need to customize these, or other install variables, skip ahead to run the main installation script.

1. To customize the `$DIVAS_HOME` directory, and other install-time parameters, open a command terminal and edit the `$STAGING_DIR/install.properties` file.
2. The home directory must be distinct from the staging directory you previously created. You can optionally change the ports used for the WebLogic administration tool, and the DIVArchive Web Services. The following is a sample `install.properties` file:

   ```
   # Installation Directory
   DIVAS_HOME=/home/diva/divas

   # WebLogic Administrator username. Not case sensitive
   # The user will prompted for the password when running the installation.
   WL_USER= weblogic
   # DIVAS Application Ports
   WL_HTTP=7001
   WL_HTTPS=7002
   DIVAS_HTTP=9443
   DIVAS_HTTPS=9444
   ```

Task 3 Run the Main Installation Script

> **Caution:** The following script removes all contents of the current WebLogic installation, and reinstalls all modules. Do not run this script if you have already installed WebLogic and Java unless you want to remove all WebLogic configuration parameters.

1. You must select a (non-administrative) Linux user to use for this procedure. *Do not use the `root` account for installation or to run the services*!
2. Execute the `$STAGING_DIR/installDIVAS.sh` script in the staging directory.
3. Enter in a new WebLogic Admin User password. The password must be at least 8 characters, and contain at least one upper case letter and one number.

4. Decide whether to configure SSL. If you choose yes, enter in the Keystore passwords.
5. When prompted to execute the `root.sh` script as a root user, open another shell window to run the `root.sh` script located in `$DIVAS_HOME/bin` directory. *Make sure you run this script with root permissions on the system*.

Task 4 Generate the API User

1. Change to the `$DIVAS_HOME/bin` directory.
2. Enter the WebLogic user name and password you previously created as arguments to the key generator as follows:

   ```
   ./runAPIKeyGenerator.sh --wluser {WL_admin_user_name} --wlpass {WL_admin_password}
   ```

3. Write down the `clientId` and API key appearing in the script output. API clients must supply these values when making web service requests.

Task 5 Configure the Connection to DIVArchive and DIVAnet

1. Locate the `divas-config.properties` file located in the `$DIVAS_HOME/DIVAS_Domain/config/dwsConfig/AppFileOverrides` directory.
2. Edit the following parameters to reflect the DIVArchive and DIVAnet IP address and port number:

   ```
   divas.service.connection.hostAddress= {DIVA IP address}
   divas.service.connection.hostPort= {DIVA port number}
   ```

Task 6 Restart Enterprise Connect

1. Using the same user account that was used to run the installation script, change to the `/bin` subdirectory under the `$DIVAS_HOME` directory, and execute the following commands:

   ```
   ./DIVAS stop divaSvcs
   ./DIVAS start divaSvcs
   ```

2. This set of commands stops and starts the DivaServices server within WebLogic without affecting the WebLogic AdminServer.

Windows Installation

Oracle DIVA Enterprise Connect runs on a Windows 64-bit OS. The executable installs and configures Oracle WebLogic 12c, the Java 8 JDK, and the Enterprise Connect web services.

After installation, you can start and stop Enterprise Connect by setting the AdminServer and DivaServices Windows Services to the running state, or using the DIVAS command-line utility (see "Starting and Stopping WebLogic Services with the DIVAS Script" on page 4-2).

Task 1 Launch the .exe File and Select Installation Options

1. Launch the Oracle DIVA Enterprise Connect `NEEDFILENAME.exe` file.
2. Select the installation options:
   - Install WebLogic
   - Configure Enterprise Connect

Task 2 Install WebLogic and Java JDK

1. Enter the location of the WebLogic installer jar (by default, this is the same directory as the Enterprise Connect installer). The filename for this jar file follows the pattern `fmw_infrastructure-12.*.jar`.
2. Enter the target installation directory for WebLogic.
3. Enter a temporary staging directory to use for the installation process.

Task 3 Configure Enterprise Connect

1. Enter the home directory where WebLogic was installed.
2. Enter the following:
   - New Weblogic admin username and password
   - New keystore passwords
   - Weblogic domain name
   - Ports for the admin and diva web services
   - DIVArchive Manager address and port

   Note: The passwords must contain at least 8 characters, at least one letter, and at least one number or special character.
3. The installer creates the Weblogic domains and servers, configures and deploys the services, and installs two Windows services: a WebLogic AdminServer service, and a DivaService service.

Task 4 Configure the Connection to DIVArchive and DIVAnet

1. Locate the `divas-config.properties` file located in the `%WL_HOME%/%DOMAIN_NAME%/config/dwsConfig/AppFileOverrides` directory.
2. Edit the following parameters to reflect the DIVArchive and DIVAnet IP address and port number:

```
divas.service.connection.hostAddress= {DIVA IP address}
divas.service.connection.hostPort= {DIVA port number}
```

Generating an API User

DIVArchive Web Services clients must supply a user name and API key to access the services. See "Generating API Users and API Keys" on page 4-1.

3

Configuring DIVA Enterprise Connect

This chapter describes how to configure DIVA Enterprise Connect after installation, and includes the following information:

Configuration Overview

Chapter 2 described the steps for a minimal DIVA Enterprise Connect configuration. However, additional configuration may be necessary based on your requirements. Configuring DIVA Enterprise Connect may include changes to the WebLogic configuration, the user, group, or role configuration in the WebLogic embedded LDAP server, or changes to the configuration file.

During installation, you may have customized the `install.properties` file with values for the HTTP and HTTPS ports, the `DIVAS_HOME` directory (for the API), and the WebLogic Admin Console ports. The `install.properties` file is located in the parent staging directory, and is only used at installation time. If changes to these values are required after installation, you must make those changes using the WebLogic Admin Console.

Editing the Configuration File

The `divas-config.properties` file contains parameters for the DIVArchive connection and the Web Services sessions. The file is located in the following directory:

Windows:

```
%DIVAS_HOME%/%DOMAIN_NAME%/config/dwsConfig/AppFileOverrides/
```

Linux

```
$DIVAS_HOME/DIVAS_Domain/config/dwsConfig/AppFileOverrides/
```

After modifying the configuration file, you must restart the DIVAServices service. On Linux, use the DIVAS command. On Windows, you can use the DIVAS command, or restart the DivaServices Windows service.

Basic Parameters of the divas-config.properties File

`divas.service.connection.hostProtocol`
This parameter is the protocol used to connect to the DIVArchive Manager and DIVAnet (**SOCKET**, **HTTP**, **HTTPS**). The use of HTTP or HTTPS requires that the *DIVAnet ManagerAdapter* is installed and configured properly. Using the HTTP option, the DIVArchive Web Services connect to the *DIVAnet ManagerAdapter*, and not directly to the DIVArchive Manager.

`divas.service.connection.hostAddress`
This parameter identifies the host name or IP address of DIVArchive or DIVAnet.

`divas.service.connection.hostPort`
This parameter identifies the port number of DIVArchive or DIVAnet.

`divas.service.connection.userName`
This parameter identifies the default user name and group name to pass to DIVArchive or DIVAnet if unassigned in WebLogic.

`divas.service.connection.applicationName`
This parameter identifies the default application name passed on the DIVArchive or DIVAnet connection.

Advanced Parameters of the divas-config.properties File

`divas.service.connection.responseTimeout`
This parameter identifies the time (in seconds) to wait for a response from DIVArchive or DIVAnet before timing out.

`divas.service.connection.retryInterval`
This parameter identifies the time (in seconds) to wait between retry attempts when a response is not received.

`divas.service.connection.retryCount`
This parameter identifies the number of retries when a response is not received from DIVArchive or DIVAnet.

`divas.service.connection.siteName`
This parameter identifies the DIVAnet site name of this server. This value is passed to DIVAnet.

The following Web Services identify session behavior parameters and configured in the `divas-config.properties` file:

`divas.service.client.maxSessions`
This parameter identifies the maximum number of simultaneous active sessions (per user).

divas.service.session.maxAge
This parameter identifies the maximum duration (in minutes) before a session is terminated.

divas.service.session.maxIdle
This parameter identifies the maximum duration (in minutes) before an idle session is terminated.

divas.service.session.allowDifferentIP
This parameter is a boolean value. This value is `true` if a session ID can be used in requests originating from different IP addresses.

Connecting to DIVAnet

DIVA Enterprise Connect can connect to DIVAnet in several ways. The services can connect directly to the *DIVAnet ManagerAdapter* through HTTP or HTTPS. There are security benefits when you use this mode. The services can also connect to the *DIVAnet ClientAdapter* either using SOCKETS or through the *ManagerAdapter* using HTTP or HTTPS. Connecting to DIVAnet has disaster recovery, content availability, and security benefits.

Connecting through the DIVAnet ManagerAdapter

DIVA Enterprise Connect can interface with DIVArchive by connecting directly to the *DIVAnet ManagerAdapter*. Connecting in this manner enables a secure outbound HTTPS connection to be established to DIVArchive. You set the `divas.service.connection.hostProtocol` attribute in the DIVA Web Services configuration file to either **HTTP** or **HTTPS**.

See Appendix B for instructions on configuring SSL certificates to support the HTTPS connection between DIVA Enterprise Connect and DIVAnet.

WebLogic Group Names for Access Control

Connecting to a *DIVAnet ManagerAdapter* can provide additional access control benefits. By default, DIVA Enterprise Connect sends a single default user name to DIVAnet. This user name is configured in the `divas-config.properties` file. However, you can configure DIVA Enterprise Connect to send a name to the ManagerAdapter on a per request basis for access control. API users can be assigned to an *Access Group* within WebLogic. When a user is assigned an access group, the Access Group name is sent to the ManagerAdapter on every web service call.

For example, a DIVArchive Web Services Client API user name Fred might be a member of a group named `divanetAdmin`. This group is configured and assigned in WebLogic. The `divanet` prefix indicates to DIVA Enterprise Connect that `divanetAdmin` is an Access Control Group, and the group name should be sent to the ManagerAdapter on every client request. You can use access rules in the *DIVAnet ManagerAdapter* to enforce access control. You use the `username` attribute in the ManagerAdapter access rules to assign access rights for the group.

Configuring Access Group Names

You can configure WebLogic users to have an Access Group name for DIVAnet using the WebLogic Admin Console. Use the following procedure to assign an Access Group name to a user:

1. Log in to the WebLogic Admin Console.

A typical URL for the admin console is `http://127.0.0.1:7001/console`.

2. Now you will add the group by navigating to **Domain Structure**, **Security Realms**, **myrealm**, **Users and Groups**, and then **Groups**.
3. Confirm that you are on the **Groups** tab, and click **New**.
4. Create a group name containing the string `divanet`, and then click **OK**.
5. Next you must assign the group to a user by navigating to **Domain Structure**, **Security Realms**, **myrealm**, **Users and Groups**, and then **Users**.
6. Click the user name you want to change, and then navigate to the **Groups** tab.
7. Click the group you want to assign (in the box on the left) and click **>** to assign the group.

   The group is now displayed in the box on the right. If it is not there, then you have not properly assigned it.
8. Click **Save** to complete the process.
9. Oracle suggests (although it is not required) adding a rule enforcing that only users with an Access Group can use the services. To create this rule, navigate to **Domain Structure**, **Deployments**, **DIVAS.1.0.0**, **Security**, and then **Policies**.

   The **DIVAS.1.0.0** menu item is a link to the deployed DIVArchive Web Services application. It appears in the table on the *Deployments* screen.
10. Click **Add Conditions**, then select **Group** from the **Predicate List** menu list, and then click **Next**.
11. Enter the first name of the group you previously added, and then click **Add**.
12. Add any remaining group names following this procedure, and then click **Finish**.

Connecting to the DIVAnet ClientAdapter

Alternatively, you can connect to the *DIVAnet ClientAdapter* running in MultiDiva mode. This will allow you to archive and restore content from multiple DIVArchive sites, and list the sites where assets are stored. You can connect to the *DIVAnet ClientAdapter* by setting the value of the `divas.service.connection.hostProtocol` attribute to **SOCKET**, and then configuring the address and port of the *DIVAnet ClientAdapter*. Finally, you can connect to the ClientAdapter through the *DIVAnet ManagerAdapter* as if you were connecting to DIVArchive.

4

Administering the Platform

This chapter describes administration of the DIVA Enterprise Connect platform and includes the following information:

Administration Overview

After you complete the basic configuration, you can use a combination of the WebLogic Administration Console and command-line utilities to administer the DIVA Enterprise Connect. The services must be operational to use the basic administrative functions described in the following sections.

Generating API Users and API Keys

DIVArchive Web Services clients must supply a user name and API key to access the services. Oracle recommends adding the user to the WebLogic platform using the *API Key Generator* command-line utility. You execute the API Key Generator script which generates a random `clientID` and a random API key, and adds them to the WebLogic user directory.

Linux

1. Change to the `$DIVAS_HOME/bin` directory.
2. Enter the WebLogic user name and password you previously created as arguments to the key generator as follows:

   ```
   ./runAPIKeyGenerator.sh --wluser {WL_admin_user_name} --wlpass {WL_admin_password}
   ```

3. Write down the `clientId` and API key appearing in the script output. API clients must supply these values when making web service requests.

Windows

1. From the command prompt, change to the `%DIVAS_HOME%/%DOMAIN_NAME%_ec` directory.
2. Enter the WebLogic user name and password you previously created:

   ```
   runAPIKeyGenerator.cmd --wluser {WL_admin_user_name} --wlpass {WL_admin_password}
   ```

3. Write down the `clientId` and API key that appear in the script output. API clients must supply these values when making web service requests.

Deleting or Modifying Users using the WebLogic Admin Console

You can delete, or modify, users through the WebLogic Admin Console. Use the following procedure to delete a user:

1. Log in to the WebLogic Admin Console.

   A typical log in URL is `http://127.0.0.1:7001/console`.

2. Navigate to **Domain Structure**, **Security Realms**, **myrealm**, **Users and Groups**, and then **Users**.
3. Select the check box next to the name of the user that you want to delete, and click **Delete**.

Use the following procedure to modify a user:

1. Log in to the WebLogic Admin Console.

   A typical log in URL is `http://127.0.0.1:7001/console`.

2. Navigate to **Domain Structure**, **Security Realms**, **myrealm**, **Users and Groups**, and then **Users**.
3. Click the name of the user that you want to modify.
4. Complete the required modifications (for example, adding a new description), and click **Save**.

Starting and Stopping WebLogic Services with the DIVAS Script

The DIVArchive Web Services platform includes the `DIVAS` script located in the `%DIVAS_HOME%/%DOMAIN_NAME%_ec` directory on Windows or the `$DIVAS_HOME/bin` directory on Linux. You use the `start`, `stop`, and `status` options to control WebLogic, and the DIVArchive Web Services. You use the `divaWebLogic`, `divaSvcs`, and `all` options to dictate the scope of the command. You must provide one option from each category to run the script as follows:

```
DIVAS {start|stop|status|help} {divaWebLogic|divaSvcs|all}
```

The following list describes the command line options:

`start`
You use this option to start WebLogic (`divaWebLogic`), the DIVArchive Web Services (`divaSvc`), or both (`all`).

`stop`
You use this option to stop WebLogic (`divaWebLogic`), the DIVArchive Web Services (`divaSvc`), or both (`all`).

status
You use this option to view status information for WebLogic (`divaWebLogic`), the DIVArchive Web Services (`divaSvc`), or both (`all`).

help
You use this option to view help for the command.

divaWebLogic
You use this option to start, stop, or view status information for only the WebLogic server.

divaSvcs
You use this option to start, stop, or view status information for only the DIVArchive Web Services.

all
You use this option to start, stop, or view status information for DIVArchive Web Services and the WebLogic server platform.

Deploying a New `.ear` File

You can use the included script to redeploy the DIVArchive Web Services `.ear` file, or deploy a new one.

1. Open a terminal console, and temporarily place the new `.ear` file in the directory `%DIVAS_HOME%` on Windows or `$DIVAS_HOME` on Linux.
2. Change to `%DIVAS_HOME%/%DOMAIN_NAME%_ec` on Windows or `$DIVAS_HOME/bin` on Linux and execute the following command:

   Linux:
   ```
   ./deploy_ear.sh -svr DivaServices -app DIVAS.1.0.0 -ear
   ../diva-ws-ear-1.0.0-{buildNumber}.ear -wluser {wlUser} -wlpass
   {wlPassword}
   ```

   Windows:
   ```
   deploy_ear.cmd {earFilename} {wluser} {wlpassword}
   ```

   Where:

   - `{buildNumber}`— the specific DIVArchive Web Services build number
   - `{wlUser}` — the WebLogic user name
   - `{wlPassword}` — the WebLogic password associated with the user

This command deploys the `.ear` file in the `dwsConfig` directory to WebLogic - undeploying the service with the name `DIVAS.1.0.0`. In the current release, you must use `DIVAS.1.0.0` as the application name for this operation.

You can also deploy an `.ear` file using the WebLogic Admin Console.

For more information refer to https://docs.oracle.com/middleware/12212/wls/WLACH/taskhelp/applications/ApplicationOverview.html.

Viewing Logs

The application logs are located in `%DIVAS_HOME%/%DOMAIN_NAME%_ec/logs` on Windows, and `$DIVAS_HOME/logs` on Linux. The installation log files are in `%DIVAS_HOME%/%DOMAIN_NAME%_ec/logs` on Windows or `$DIVAS_HOME/logs/install` on Linux.

To change the log level, open the WebLogic Admin Console and navigate to **Environment**, **Servers**, **DivaServices**, **Logging**, **General**, **and then Advanced**.

Set the ***Minimum severity to log***, and the ***Message Destination Severity level*** to the appropriate logging level. You can optionally set additional severities for ***Standard out*** and the ***Domain log***, and then click **Save**.

The HTTP access logs are also in the same directory as the application logs. The access logs track each HTTP request issued to the DIVA Enterprise Connect API. The HTTP access logs are stored in files with the file name format `access*.log`.

Uninstalling DIVA Enterprise Connect and Weblogic

Linux

Run the `uninstall.sh` script in `$DIVA_HOME/bin` as the root user. This script removes the service definition for DIVA Enterprise Connect. You can then delete the application files.

Windows

First stop the service by running the `uninstall.cmd` script located in `%DIVA_HOME%/%DOMAIN_NAME%_ec`. Then, run the Weblogic `deinstall.cmd` utility located in the `%DIVAS_HOME%/oui/bin` directory. The Windows Weblogic installer also creates a shortcut to a UI version of this utility. After running the utility, you can delete the application files.

5

Testing the Installation

This chapter describes testing your installation, and incudes the following information:

Testing Overview

You can test DIVA Enterprise Connect using standard SOAP or REST test clients that can consume WSDL (Web Services Description Language) documents, WADL (Web Application Description Language) documents, or both. The DIVArchive Web Services also contains a set of scripts that make testing the DIVArchive Web Services easier.

Using Automated Testing Tools

Many open source and commercial tools exist for testing Web Services. Many of them can parse a WSDL or WADL document to enable Web Services testing. You must supply the following URL of the WSDL or WADL to use these tools:

```
http://127.0.0.1:9443/diva/service/soap/2.2/DIVArchiveWS_SOAP?WSDL

http://127.0.0.1:9443/diva/service/rest/2.2/DIVArchiveWS_REST/application.wadl
```

Substitute the actual network address of the DIVArchive Web Services platform in the URL. You may need to provide a user name and password to access the WSDL, or WADL, document from.

Using the `wsTest` Scripts

The DIVA Enterprise Connect release includes scripts that can be used for testing. If you have access to the DIVA Enterprise Connect release, these scripts can make testing DIVArchive Web Services easier.

On Linux, the wsTest scripts use the standard cURL utility to invoke the DIVArchive Web Services. The scripts automatically create a session for each call by default. This simplifies the testing process for the Web Service calls. The wsTest scripts are located in the $DIVAS_HOME/scripts/wsTest on Linux, and %DIVAS_HOME%/%DOMAIN_NAME%_ec on Windows

You use the following syntax to execute the scripts:

Linux

```
./wsTestRest.sh {baseUrl} {version} {templateCommand}
./wsTestSoap.sh {baseUrl} {version} {templateCommand}
```

Windows

```
wsTestSoap.cmd {baseUrl} {version} {templateCommand}
wsTestRest.cmd {baseUrl} {version} {templateCommand}
```

The following is a list of the parameters used with the scripts:

baseUrl
This parameter identifies the base URL of the DIVArchive Web Service (for example, http://127.0.0.1:9443/diva/service).

version
This parameter identifies the protocol version level of the DIVArchive Web Service API (the current protocol release is the string v2.2). *This is not the software release level of the DIVA Enterprise Connect*.

templateCommand
This parameter identifies the Web Services command to execute. The parameters for the command are looked up in the template file, which contains a section for each call. The templates are located in the soapTemplate.sh script for SOAP, and in the restTemplate.sh script for REST (**NOTE**: the template filenames have the .sh extension on the Windows platform as well). Each template depicts a sample call with all parameters populated with test data. You must edit these scripts to supply the desired parameters for each command.

--nosess
This is an optional parameter. Normally, the scripts will attempt to obtain a Web Services session ID before executing each command. You can use this argument to supply your own session ID in the template.

Example Call

The following is an example execution of the wsTest scripts. The getRequestInfo() call is being tested in the example. In this example, a new session ID will be fetched if needed. The parameters for this call are retrieved from the restTemplate.sh file.

Linux

```
./wsTestRest.sh http://127.0.0.1:9443/diva/service v2.2 getRequestInfo
```

Windows

```
wsTestRest.cmd http://127.0.0.1:9443/diva/service 2.2 getRequestInfo
```

Creating Sessions Automatically

Caution: You must only use the automatic session creation feature for testing, and not in production.

The scripts are configured to automatically create the required session ID to run each service. The scripts generate the new session ID, and then pass it in the chosen web service request. The `wsTest` scripts save the new session ID in a file, so it can be reused on the next call. Sessions (by default) last for 30 minutes, and then they expire.

If you do not want the `wsTest` scripts to automatically create the session ID, you must include the `--nosess` parameter in the command line when executing the test script. You can assign a fixed value for the *SESS_ID* variable in the template file. If you call `registerClient()` yourself, and populate your own session code, the code will eventually expire, and you will have to repeat the process.

Editing the Templates

If you view the `restTemplate.sh` or `soapTemplate.sh` shell scripts, you will see a shell variable representing each Web Service call, and XML parameters. You can edit the XML to customize the arguments to the Web Service. Some parameter values in the template are derived from a shell variable. For example, the following is a section in the `restTemplate.sh` file that defines the parameters for the `getObjectInfo()` request:

```
getObjectInfo='
<p:getObjectInfo xmlns:p="http://interaction.api.ws.diva.fpdigital.com/xsd">
   <p:sessionCode>'$SESS_ID'</p:sessionCode>
   <p:objectName>'$OBJ_NAME'</p:objectName>
   <p:objectCategory>'$OBJ_CATEGORY'</p:objectCategory>
</p:getObjectInfo>'
```

The `$OBJ_NAME` variable is defined at the top of the template script. The variables let you easily customize parameters for many commands. You can choose to populate the variable at the top of the template, or remove the variable and hard code the `objectName` directly in the `getObjectInfo()` command.

Authentication

Because basic authentication for Web Services is turned on by default, you must populate the *$CLIENT_ID* and *$AUTH_KEY* variables at the top of `restTemplate.sh`, or `soapTemplate.sh` if you are using SOAP. You must use values that were created by executing the DIVA Enterprise Connect Key Generator, or preset values assigned to you by Oracle Support.

Calling the cURL Tool Directly

The cURL command line tool enables transferring data to, or from, a server using one of the supported protocols. The cURL utility supports HTTP transfers and basic authentication. On Linux, the `wsTest` scripts use cURL in the implementation, but you can call cURL directly as needed. The cURL tool is a standard utility in Oracle Linux 7 and later.

A

DIVArchive Options and Licensing

The following table identifies DIVArchive options and licensing metrics.

Part Number	Description	Licensing Metric
L101163	Oracle DIVArchive Nearline Capacity	Per TB
L101164	Oracle DIVArchive Archive Capacity	Per Slot
L101165	Oracle DIVArchive Actor	Per Server
L101166	Oracle DIVArchive Manager	Per Server
L101167	Oracle DIVArchive Partial File Restore	Per Wrapper
L101168	Oracle DIVArchive Avid Connectivity	Per Server
L101169	Oracle DIVArchive Application Filtering	Per Server
L101170	Oracle DIVArchive Storage Plan Manager (2 storage plans are included with a DIVArchive Manager License)	Per Server
L101171	Oracle DIVAnet	Per Server
L101172	Oracle DIVAdirector	Per User
L101918	Oracle DIVArchive Export / Import	Per Server
L101919	Oracle DIVArchive Additional Archive Robotic System	Per Tape Library
L101920	Oracle DIVArchive Automatic Data Migration	Per Server

B

Configuring Outbound SSL Connections to DIVAnet

The following procedures describe how to configure a secure HTTPS connection from Oracle DIVA Enterprise Connect to the *DIVAnet ManagerAdapter*. The *DIVAnet ManagerAdapter* should be located on the DIVArchive Manager platform.

Before starting this procedure, ensure that during the DIVA Enterprise Connect Linux installation, you answered `yes` when prompted with *Do you wish to configure SSL?*. Ensure that you have properly configured HTTPS in the `divas-config.properties` file (see Chapter 3 for more information on setting the properties file).

The following steps require running utilities found in the Java JDK. The `%JDK_HOME%` is not an actual variable, but instead refers to `%DIVAS_HOME%/JDK` on Windows, and `$DIVAS_HOME/JDK/default` on Linux.

Adding the DIVA Enterprise Connect Certificate to DIVAnet

Use the following procedure to add the DIVA Enterprise Connect certificate to DIVAnet:

1. **Copy Certificate**: Take the DIVA Enterprise Connect certificate, located (by default) in `%DIVAS_HOME%/%DOMAIN_NAME%/config/cert/divasCert.cer`, and copy it to the computer where the *DIVAnet ManagerAdapter* is running. Change directory to `%DIVANET_HOME%` and place the certificate at that location.
2. **Import**: Run the following import command on the DIVAnet platform:

   ```
   "%JKD_HOME%\keytool" -importcert -file divasCert.cer -alias divaselfsigncert
   -keystore Java\lib\security\cacerts2
   ```

   At the password prompt, enter the DIVAnet keystore password (by default, `changeit`).

   Enter `Y` when prompted *Trust this certificate?*.
3. **Verify**: Use the following command to verify the import:

   ```
   "%JDK_HOME%\keytool" -list -keystore Java\lib\security\cacerts2 -alias
   divaselfsigncert -v
   ```

4. **Delete the File**: `%DIVANET_HOME%\divasCert.cer`.

Adding the DIVAnet Certificate to DIVA Enterprise Connect

Use the following procedure to add the DIVAnet certificate to DIVA Enterprise Connect:

1. **Export**: On the DIVAnet computer, change directory to `%DIVANET_HOME%`, and execute the following command to export the DIVAnet certificate:

   ```
   "%JDK_HOME%\keytool" -exportcert -rfc -alias diva1219 -file Oracle_DIVAnet.cer
   -keystore Program\divanet\lib\diva129.jks
   ```

   At the password prompt, enter the DIVAnet keystore password (by default, `changeit`).

2. **Copy Certificate**: Copy the Oracle_`DIVAnet.cer` that you just created to the computer where DIVA Enterprise Connect is running. Place the certificate in `%DIVAS_HOME%`.

3. **Import**: Change directory to `%DIVAS_HOME%`, and run the following import command on the DIVA Enterprise Connect platform:

   ```
   %JDK_HOME%/bin/keytool -importcert -file Oracle_DIVAnet.cer -alias "oracle
   divanet" -keystore JDK/default/jre/lib/security/cacerts
   ```

4. **Verify**: Use the following command to verify the import:

   ```
   %JDK_HOME%/bin/keytool -list -keystore JDK/default/jre/lib/security/cacerts
   -alias "oracle divanet" -v
   ```

   At the password prompt, enter the DIVA Enterprise Connect keystore password (by default, `changeit`).

   Enter `Y` when prompted *Trust this certificate?*.

5. **Delete the File**: `%DIVAS_HOME%/Oracle_DIVAnet.cer`.

Configuring SSL in WebLogic

Use the following procedure to configure SSL in WebLogic:

1. Log into the Weblogic Admin Console, and click **Lock and Edit** located in the upper-left corner of the screen.
2. Navigate to **Environment**, **Servers**, **DivaServices**, **SSL**, and then **Advanced**.
3. Set ***Hostname Verification*** to **None**, and set ***Two Way Cert Behavior*** to **Client Certs Not Requested**, and then click **Save**.
4. Click **Activate Changes** at the upper-left corner of the screen.
5. Execute the following commands to restart the services:

   ```
   DIVAS stop divaSvcs
   DIVAS start divaSvcs
   ```

   On Windows, you can also restart the services from the Microsoft Services Control Manager.

   The *DIVAnet ManagerAdapter* must be running in HTTPS mode (HTTPS is the default). See to the *Oracle DIVAnet Installation, Configuration, and Operations Guide* in the *Oracle DIVAnet documentation* library for more information.