Identity Manager

Driver for Entity Data Model Implementation Guide

October 2019

Legal Notice

Contents

About this Book and the Library

This guide explains how to install and configure the Identity Manager Driver for Entity Data Model.

Intended Audience

This book provides information for individuals responsible for understanding administration concepts for roles and resource management across the enterprise, and implementing a secure, distributed administration model.

Other Information in the Library

The library provides the following information resources:

Entity Data Model Administrator Guide

Provides conceptual information and step-by-step guidance for administrative tasks in the Entity Data Model product. Specifically, it provides instructions for the following Entity Data Model users:

- All administrator authorizations
- Business Role managers
- Review owners
- Separation of Duties Policy owners
- Application owners

Identity Manager Driver Administration Guide

Provides information about administration tasks that are common to all Identity Manager drivers.

Identity Manager Driver Guides

Provide implementation information specific to an Identity Manager driver.

Identity Manager Setup Guide

Provides an overview of Identity Manager and its components. This book also provides detailed planning and installation information for Identity Manager.

Designer Administration Guide

Provides information about designing, testing, documenting, and deploying Identity Manager solutions in a highly productive environment.

User Application: Administration Guide

Describes how to administer the Identity Manager User Application.

User Application: User Guide

Describes the user interface of the Identity Manager User Application and how you can use the features it offers, including identity self-service, the Work Dashboard, role and resource management, and compliance management.

User Application: Design Guide

Describes how to use the Designer to create User Application components, including how to work with the Provisioning view, the directory abstraction layer editor, the provisioning request definition editor, the provisioning team editor, and the role catalog.

Identity Reporting Guide

Describes the Identity Reporting Module for Identity Manager and how you can use the features it offers, including the Reporting Module user interface and custom report definitions, as well as providing installation instructions.

Analyzer Administration Guide

Describes how to administer Analyzer for Identity Manager.

1 Understanding the Identity Manager Driver for Entity Data Model

The **Identity Manager Driver for Entity Data Model** allows you to provision application-specific permission catalog data from Entity Data Model to Identity Manager. This gives you the ability to review and certify permission assignments using Entity Data Model, as well as to request and provision these permissions using Identity Manager. The driver also can provision users in the Identity Vault for Identity Manager as needed for the customer's use-case.

Understanding the Workflow Process

The following workflow shows how you can streamline the process for maintaining user identities.

Entity Data Model

Entity Data Model collects data from a wide variety of identity and application sources. **Identity sources**, such as SAP User Management and the Identity Vault in Identity Manager, provide the attributes of a user's identity. **Application sources**, such as Salesforce.com, provide account and permission information. Some of the account and permission information might be gathered from systems that are not already connected to identities in Identity Manager.

Entity Data Model helps you join the imported account, permission, and attribute data into a **unified identity**. Then you review and certify whether each unified identity should have the assigned resources. If permission assignments change, Entity Data Model helps you fulfill the changes by creating manual tasks or initiating provisioning workflows in Identity Manager.

Entity Data Model driver

Using an account in Identity Manager, the transfers a snapshot of the permissions and permission assignments from the Entity Data Model database to Identity Manager. This process creates assignment actions for Identity Manager to set the actual state of the affected permissions without the need for user intervention.

You can also configure the driver to create new user accounts in Identity Manager based on identities published from Entity Data Model. After adding the accounts, the driver reports the DN and tree name of the newly created users to Identity Manager.

NetIQ recommends that you create a dedicated system account in the identity applications for the driver. A system account provides the following advantages:

- Allows you to track any actions that the driver takes in Identity Manager
- Allows the driver to set resource assignments in Identity Manager
- Reduces the number of approval workflows required to assign and revoke resources to identities in Identity Manager

Identity Manager

When receiving the data from the Entity Data Model driver, Identity Manager populates the Identity Vault with the user identities and adds account and permission information to the identity applications catalog. Because Entity Data Model collects data from more sources than might be connected to Identity Manager, the catalog now has identities, permissions, and accounts that represent a larger picture of your identity and access environment.

In the catalog, Identity Manager administrators can create roles and permissions associated with the application sources that Entity Data Model collected. Then users can manage their unified identity and request access to other resources in the catalog even if those applications are not directly connected to Identity Manager. To process user requests, administrators can configure workflows. You can also use workflows to fulfill the change requests generated by a review in Entity Data Model.

For more information about using Entity Data Model, see the NetIQ Identity Governance documentation site. For more information about Identity Manager, see the NetIQ Identity Manager documentation site.

Understanding Synchronization and Reflection

Entity Data Model can collect data from identity and application sources that are not connected to Identity Manager. The Entity Data Model driver allows you to **synchronize** changes to identities and applications with user and resource objects in Identity Manager. You can also **reflect** collected user identities and application data as resources in the Identity Vault. The driver provides Global Configuration Values (GCVs) that allow you to delete or disable user objects or delete these resource objects in the Identity Vault. Alternatively, you can remove the association between the user object and the identity in Entity Data Model. For more information, see "Understanding Synchronization and Reflection" in the NetIQ Identity Governance Administrator Guide.

Planning to Install and Configure the Driver

This section provides useful information for planning the installation and configuration process.

Installation Requirements

The Entity Data Model driver requires the following applications and files, at a minimum. When you installed Identity Manager, you might also have chosen to install the files for the Entity Data Model driver.

- Access Review 1.1 or Entity Data Model 2.5
- Identity Manager 4.5 Service Pack 1 at a minimum, particularly the following components:
  - Identity applications
  - Designer
  - Remote Loader
  - Role and Resource Service driver

    `NOVLRSERVB` - Role and Resource Service Driver Base, package version 4.5.0.20140925170245, at a minimum
  - User Application driver

    `NOVLUABASE` - User Application Base, version 4.5.1.20150602213315, at a minimum

`NOVLPROVNOTF` - Provisioning Notification Templates, version 2.0.1.20150528174045, at a minimum

- Drive Set packages

  `NOVLACOMSET` - Driver Set package for Common Settings Advanced Edition

  `NOVLCOMSET` - Driver Set package for Common Settings

- Database JDBC file
  - Third-party JDBC driver for connecting to the Entity Data Model database
- Entity Data Model driver file
  - `arshim.jar` - Entity Data Model driver shim
- Entity Data Model driver packages
  - `NOVLARBASE` - Entity Data Model Base
  - `NOVLARDCFG` - Entity Data Model Default Configuration
  - `NOVLARMSINFO` - Entity Data Model Managed System Information
  - `NOVLARWDSYN` - Entity Data Model Password Sync

Information Needed for Installation and Configuration

Ensure that you have the information that you need to install and configure the Entity Data Model driver. For more information about the process, see “Checklist for Installing and Configuring the Driver” on page 11.

Entity Data Model settings

- Host and port of the Entity Data Model server
- URL for the Entity Data Model application
- (Conditional) For https connectivity, security certificate for the Entity Data Model application
- Account and password for a global or data administrator in Entity Data Model
- Account and password for the administrator of the Entity Data Model databases
- Name of the Operations table in the Entity Data Model database, by default `igops`
- OSP client name and password for Entity Data Model in the Roles Based Provisioning Module configuration utility

Identity Manager settings

- Host and port for the Remote Loader running on the Entity Data Model server
- DN for the User Application driver
- URL for the User Application where an administrator creates user accounts

  By default, the URL contains `IDMProv`.
- (Conditional) For https connectivity, security certificate for the User Application
- Account and password for an administrator of the User Application

2 Installing and Configuring the Entity Data Model Driver

The installation and configuration process for the Identity Manager driver for Entity Data Model, formerly known as Access Review, requires access to the Entity Data Model server, Identity Manager Remote Loader, and Designer for Identity Manager. This guide makes the following assumptions:

- Entity Data Model is not installed on the same server as the Identity Manager engine or the identity applications.
- The Entity Data Model driver is installed with the Identity Manager Remote Loader on the same server as Entity Data Model.

Ensure that you have activated Identity Manager. You do not need to separately activate the Entity Data Model driver.

Checklist for Installing and Configuring the Driver

Before beginning the installation and configuration process, NetIQ recommends that you review the following steps:

	Checklist Items
□	1. Review the considerations for installing and configuring the Entity Data Model driver. For more information, see "Information Needed for Installation and Configuration" on page 9.
□	2. Ensure that your environment meets the requirements for installing and configuring the Entity Data Model driver. For more information, see "Planning to Install and Configure the Driver" on page 8.
□	3. Install the Remote Loader and the driver files on the Entity Data Model server. For more information, see "Installing the Remote Loader and Driver Files" on page 12.
□	4. Ensure that the Entity Data Model driver can perform provisioning tasks in the identity applications. For more information, see "Creating an Identity Manager Provisioning Service Account for the Driver" on page 13.
□	5. Ensure that you have the appropriate packages installed and imported for the Entity Data Model driver, User Application driver, and notifications object in Designer. For more information, see "Updating the Base Package for the Entity Data Model Driver" on page 14.
□	6. Configure the basic settings for the Entity Data Model driver. For more information, see "Configuring the Entity Data Model Driver" on page 14.
□	7. Apply the system account that you created in the identity application for the driver. For more information, see "Adding the Driver Account to the Entity Data Model Driver" on page 17.
□	8. Deploy the updated Entity Data Model driver, User Application driver, and notifications object. For more information, see "Deploying the Entity Data Model Driver and Supporting Objects" on page 17.

	Checklist Items
❑	9. Ensure that Entity Data Model can integrate collected permissions and permission assignment tasks with the role and resource catalog in Identity Manager. For more information, see “Configuring Entity Data Model” on page 17.

Installing the Remote Loader and Driver Files

The files for the Entity Data Model driver need to be on the same server where you install the Remote Loader.

- “Installing the Remote Loader” on page 12
- “Adding the Entity Data Model Driver File to the Identity Vault” on page 12
- “Adding the Entity Data Model Driver Files to the Remote Loader Server” on page 12

Installing the Remote Loader

The Remote Loader loads drivers and communicates with the Identity Manager engine on behalf of drivers installed on remote servers. To ensure appropriate communication between the Entity Data Model driver and Identity Manager, NetIQ recommends that you install the Remote Loader on the Entity Data Model server.

For more information about installation, see “Installing and Managing the Remote Loader” in the *NetIQ Identity Manager Setup Guide*.

Adding the Entity Data Model Driver File to the Identity Vault

By default, the driver files are installed on the Identity Manager server at the same time as the Identity Manager engine. The installation program extends the Identity Vault’s schema and installs the driver shim and the driver configuration file. It does not create the driver in the Identity Vault or upgrade an existing driver’s configuration.

Adding the Entity Data Model Driver Files to the Remote Loader Server

This section provides information for adding the files for the Entity Data Model driver to the Remote Loader server.

1 Log in to the server where you installed the Remote Loader.

NetIQ recommends that you install the Remote Loader on the Entity Data Model server.

2 Copy the `arshim.jar` file from the Identity Vault server to the `lib` directory for the Remote Loader, located by default in the `opt/novell/eDirectory/lib/dirxml/classes` directory.

3 In the `lib` directory, install the third-party JDBC driver that supports the Entity Data Model database.

4 In the `/etc/opt/novell/dirxml/rdxml` Entity Data Model driver directory, create a text file that defines the classpath for the Entity Data Model driver. For example:

```
-description "AR Driver"
-commandport 8000
-connection "port=8090"
-trace 3
-tracefile "/opt/netiq/ar.log"
-tracefilemax 100M
-class "com.novell.nds.dirxml.driver.arshim.AccessReviewDriverShim"
```

For more information about classpaths, see "Installing and Managing the Remote Loader" in the *NetIQ Identity Manager Setup Guide*.

5 Note the port number associated with the Remote Loader instance. You need this value when configuring the driver in Designer.

Creating an Identity Manager Provisioning Service Account for the Driver

The Entity Data Model driver needs a user account in Identity Manager to grant and revoke permissions. The account must have **Resource Administrator** permissions in the identity applications.

1 Log in to Identity Manager Home as an administrator.

2 To create the new system account, complete the following steps:

  2a Select **Create Users and Groups**.

  2b Create a new User object for a system account. For example, in the `OU=sa,O=data` container, create an object called `driverProvServiceAcct`.

  2c Specify values for the required fields for the new user, and then select **Continue**.

  2d Specify a password for the new user object.

3 To assign resource administrator permissions to the account, complete the following steps:

  3a Select **Administration > RBPM Provisioning and Security**.

  3b Select **Administrator Assignments > Assign**.

  3c Specify a description for this assignment request. For example, `Resource Provisioning Account`.

  3d For **Domain**, specify **Resource**.

  3e For **User(s)**, specify the name that you assigned to the new User object.

  3f Select **All Permissions**.

  3g Select **Assign**.

4 Log out of Identity Manager Home.

Preparing the Entity Data Model Driver

This section helps you create, configure, and deploy the Entity Data Model driver. You perform these tasks in your project in Designer.

- "Updating the Base Package for the Entity Data Model Driver" on page 14
- "Configuring the Entity Data Model Driver" on page 14
- "Adding the Driver Account to the Entity Data Model Driver" on page 17
- "Deploying the Entity Data Model Driver and Supporting Objects" on page 17

Updating the Base Package for the Entity Data Model Driver

NetIQ regularly provides updates to the Identity Manager drivers. You must have the latest content for the Entity Data Model driver, User Application driver, and notifications object. For more information about the packages, see "Installation Requirements" on page 8.

1. Open Designer.
2. Select **Help > Check for Package Updates**.
3. Select the updated packages that you want to update, including packages for the User Application driver and notification templates.
4. Click **Yes**.
5. When the update completes, restart Designer.

Configuring the Entity Data Model Driver

This section helps you configure the Entity Data Model driver and establish its basic settings.

The driver interacts with Entity Data Model through database views. It uses the Entity Data Model administrator account as well as an account in the Identity Manager identity applications. When configuring the driver, you need information about Entity Data Model and Identity Manager settings. For more information about required settings, see "Information Needed for Installation and Configuration" on page 9.

NOTE: The Entity Data Model driver requires the driver set packages for common settings: `NOVLACOMSET` and `NOVLCOMSET`. Ensure that you import these packages before configuring the driver. For more information about the packages, see "Installation Requirements" on page 8.

1. In the **Modeler** view of Designer, select **Developer**.
2. (Conditional) If you have more than one driver set in the Identity Vault, select the driver set in the **Modeler** view to which you want to add the driver.
3. In the **Palette** view, expand **Service**.
4. Drag **Entity Data Model** to the **Modeler** view.

   This action opens the Driver Configuration Wizard.
5. For **Select Driver Base Configuration,** select **Entity Data Model Base**, then click **Next**.
6. For **Optional Features**, select the following items:
   - `Default Configuration`
   - `Managed System Information`
   - `Password Synchronization`
7. Click **Next**.
8. For **Driver Name**, specify a value. For example, `Entity Data Model Driver`.
9. Click **Next**.
10. (Conditional) Select *Yes* or *No* to determine if the driver will use the Remote Loader. If you select *No*, skip to Step 11. If you select *Yes*, use the following information to complete the Remote Loader configuration, then click **Next**:
    - **Host Name:** Specify the hostname or IP address of the server where the driver's Remote Loader service is running.

- **Port:** Specify the port number where the Remote Loader is installed and running. The default port number is 8090.
- **KMO:** Specify the Key Name of the Key Material Object (KMO) that contains the keys and certificate the Remote Loader uses for an SSL connection. This parameter is only used when you use SSL for connections between the Remote Loader and the Identity Manager engine.
- **Other Parameters:** Specify any other parameters required to connect to the Remote Loader. Any parameters specified must use a key-value pair format, as follows: `paraName1=paraValue1 paraName2=paraValue2`.
- **Remote Password:** Specify the Remote Loader's password as defined on the Remote Loader. The Identity Manager server (or Remote Loader) requires this password to authenticate to the Remote Loader.
- **Driver Password:** Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Identity Manager server.

11 Specify the following details to connect to the Entity Data Model database, then click **Next**:

Authentication ID: Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application.

Connection Information: Specify the IP address or name of the server the application shim should communicate with.

Password: Specify a password for the driver to communicate with the application.

Driver Options: Select **Show** to display the driver options and specify the following parameters:

- **Entity Data Model Database Connection URL:** Specify the JDBC connection URL. For example, `jdbc:postgresql://(host):(port)/arops`, where `arops` is the default operation table.
- **JDBC Driver Class Name:** Specify the JDBC driver class name. For example, `org.postgresql.Driver`.

Publisher Options: Select **Show** to display the publisher options and specify the following parameters:

- **Entity Data Model Resources Base Container:** Specify the name for the base container for the Entity Data Model resources. For example, `Identity_Governance_Resources`.
- **User Application Driver DN:** Specify the DN for User Application driver. For example, `CN=User Application Driver,CN=driverset1,O=system`.
- **User Application Provisioning URL:** Specify the User Application provisioning URL. For example, `http://<uahost>:<port>/IDMProv`.
- **User Application User Name:** Specify the user name for the User Application. For example, `Admin`.
- **User Application User Password:** Specify the password for the user name of the User Application. For example, `password`.
- **Provisioning Service Account Password:** Specify the password for the Provisioning Service Account. For example, `pswd`.

Allow IDM Account Creation and Migration?: Click **Adds and Migrate Allowed** to allow Identity Manager to create new users based on the identities published from the Entity Data Model repository. Specify the following parameters and click **Next**.

- **Entity Data Model Application URL:** Specify the URL of the server where Entity Data Model application is hosted. For example, `http://arhost:8080`.

- **Entity Data Model Data Administrator User Name:** Specify the name for the Entity Data Model database administrator. For example, `igadmin`.
- **Entity Data Model Data Administrator User Password:** Specify the password for the Entity Data Model database administrator. For example, `igpassword`.
- **OSP Client Name:** Specify the user name for the User Application. For example, `iac`.
- **OSP Client Password:** Specify the password for the user name of the User Application. For example, `iacpswd`.

12 (Conditional) On the **Entity Data Model Default Configuration Information** page, specify the container name where the new users from Entity Data Model will be created in the **Publisher user Object Placement** field. For example, `data\users\igusers`.

13 (Conditional) On the **Entity Data Model Managed System Information** page, fill in the following fields to define the ownership of Entity Data Model, then click **Next**:

General Information

- **Name:** Specify a descriptive name for the managed system.
- **Description:** Specify a brief description of the managed system.
- **Location:** Specify the physical location of the managed system.
- **Vendor:** Specify the vendor of the managed system.
- **Version:** Specify the version of the managed system.

System Ownership

- **Business Owner:** Select a user object in the Identity Vault that is the business owner of Entity Data Model. This can only be a user object, not a role, group, or container.
- **Application Owner:** Select a user object in the Identity Vault that is the application owner of Entity Data Model. This can only be a user object, not a role, group, or container.

System Classification

- **Classification:** Select the classification of Entity Data Model. This information is displayed in the reports. The options are as follows:
  - Mission-Critical
  - Vital
  - Not-Critical
  - Other

    If you select **Other**, you must specify a custom classification for Entity Data Model.
- **Environment:** Select the type of environment Entity Data Model provides. The options are as

  follows:
  - Development
  - Test
  - Staging
  - Production
  - Other

    If you select **Other**, you must specify a custom environment for Entity Data Model.

14 Click **Finish**.

Adding the Driver Account to the Entity Data Model Driver

This section helps you apply the system account that you created for the driver in the identity applications to the driver. For more information about the account, see "Creating an Identity Manager Provisioning Service Account for the Driver" on page 13.

NOTE: Identity Manager shares Global Configuration Values (GCVs) with the entire driver set, the Role and Resource driver, and the Entity Data Model driver. NetIQ recommends that you periodically review the GCVs to ensure that it does not get reset by installations of other drivers or changes to the Entity Data Model driver.

---

1. In the **Outline** view of Designer, right-click the Entity Data Model driver.
2. Select **Properties**.
3. In the navigation pane, select **Driver Configuration** and select **Publisher Options** tab.
4. Specify the DN and password of the service account created for `User Application Provisioning Service Account DN`.

   The **Properties** window displays the name of the service account based on the descriptive name that you created when you added the account to the GCVs for the driver set. For example, `User Application Provisioning Service Account DN`. For more information, see "Creating an Identity Manager Provisioning Service Account for the Driver" on page 13.
5. Click **OK**.

Deploying the Entity Data Model Driver and Supporting Objects

After you create, configure, or modify the driver, you must deploy the Entity Data Model driver, User Application driver, and notifications object.

1. In the **Modeler** or **Outline** view of Designer, right-click **Driver Set** or the driver set where you installed the Entity Data Model driver.
2. Select **Live > Deploy**.
3. Select **Deploy**, then select **OK**.
4. Right-click the Entity Data Model driver, then repeat the two deployment steps.
5. Deploy the User Application driver.
6. Deploy the Default Notification Collection object.
7. (Conditional) If Identity Manager requests Security Equivalences values, set equivalence to the `admin.sa.system` user.

Configuring Entity Data Model

Entity Data Model uses the Entity Data Model driver to integrate collected permissions and permission assignment tasks with the role and resource catalog in Identity Manager. To do so, you must modify the Entity Data Model configuration settings.

- "Integrating the Driver with Entity Data Model" on page 18
- "Integrating Entity Data Model Data with Identity Manager" on page 18

Integrating the Driver with Entity Data Model

You must configure Entity Data Model to support integration with the Entity Data Model driver. NetIQ provides the Identity Governance Configuration utility, which allows you to modify settings for Entity Data Model. For more information about using the utility, see "Running the Identity Governance Configuration Ultility" in the NetIQ Identity Governance Administrator Guide.

1. Log in to the server that hosts Entity Data Model.
2. Navigate to the installation directory for Entity Data Model. For example, `opt/netiq/idm/apps/idgov`.
3. To run the utility, enter the following command:

   ```
   ./bin/configutil.sh -password db_password
   ```

4. Select **Miscellaneous Settings**.
5. Select **Enable integration using Identity Manager Driver for Entity Data Model**, then click **Save**.
6. To enable the new configuration, restart the application server that hosts Entity Data Model.

Integrating Entity Data Model Data with Identity Manager

The Entity Data Model driver helps you integrate data that Entity Data Model collects from application sources with role and resource data in Identity Manager. You might want to do this if your Entity Data Model environment collects permissions from applications that are not also connected systems in Identity Manager. After you set up the integration, you can export the permissions and their assignments from the non-connected applications to Identity Manager.

For more information, see "Integrating Collected Data with identity Manager" in the NetIQ Identity Governance Administrator Guide.

3 Upgrading an Existing Driver

The driver shim files are updated when you update the Remote Loader on the server.

Upgrade Procedure

The driver upgrade process involves upgrading the installed driver packages and updating the driver files. The driver patch file contains the software to update the driver files. Currently, no new versions are available for the driver.

4 Configuring Secure Communication

You can configure a secure connection for communication among the Identity Manager Driver for Identity Governance, formerly known as Access Review, Entity Data Model, and Identity Manager.

- "Configuring TLS/SSL Communication with Identity Manager" on page 21
- "Configuring TLS/SSL Communication with the Entity Data Model Database" on page 22

Configuring TLS/SSL Communication with Identity Manager

To ensure that the Entity Data Model driver communicates securely with the Entity Data Model server and the User Application, you can configure a TLS/SSL connection. The driver supports the following types of certificates for secure communication:

- Self-signed public key certificate for the server
- Trusted root certificate of the certificate authority (CA) used to sign the server's public key certificate

Using a Self-Signed Public Key Certificate

To use a self-signed public key certificate, you need the iac-certtool utility. You can download the utility from the Entity Data Model customer portal.

1. Log in to the Entity Data Model server as an administrator.
2. Run the iac-certtool utility.
3. Specify the URL for the Entity Data Model application or the User Application.
4. Select **Get Certificate**.
5. If the content of the certificate is correct, select **Yes**.
6. Copy the certificate content to a text file.
7. In Designer, run the configuration wizard for the Entity Data Model driver.
8. In the Publisher configuration section, paste the certificate content in the certificate input field.
9. Complete the configuration, and then deploy the updated driver.

Using a Trusted Root Certificate from a Certificate Authority

If your organization uses a public key certificate signed by a certificate authority, such as Verisign or Entrust, you must obtain the appropriate trusted root certificate that corresponds to the certificate authority. You can obtain the trusted root certificate from your organization or the certificate authority your organization used.

1. Acquire the trusted root certificate.
2. In Designer, run the configuration wizard for the Entity Data Model driver.

3. In the Publisher configuration section, import the trusted root certificate.
4. Complete the configuration, and then deploy the updated driver.

Configuring TLS/SSL Communication with the Entity Data Model Database

To ensure that the Entity Data Model driver communicates securely with the Entity Data Model database, you can configure a TLS/SSL connection. You must enable SSL for both the database and the driver.

- "Preparing the Database Platform for SSL Communication" on page 22
- "Enabling the Entity Data Model Databases for SSL Communication" on page 24
- "Enabling the Entity Data Model Driver for SSL Communication" on page 25

Preparing the Database Platform for SSL Communication

This section provides information for creating an SSL server certificate that the PostgreSQL and Oracle database platforms can use for secure communication with the Entity Data Model driver.

- "Preparing PostgreSQL for SSL Communication" on page 22
- "Preparing Oracle for SSL Communication" on page 23

Preparing PostgreSQL for SSL Communication

1. On the server where you deployed Entity Data Model, stop Tomcat.
2. Log in to the PostgreSQL server for Entity Data Model.
3. Stop Postgres.
4. To generate a passphrase-protected certificate, enter the following command:

   ```
   openssl req -new -text -out cert.req
   ```

5. To remove the passphrase so the server can start the postmaster automatically, enter the following command:

   ```
   openssl rsa -in privkey.pem -out cert.pem
   ```

6. To convert the certificate into a self-signed certificate, enter the following command:

   ```
   openssl req -x509 -in cert.req -text -key cert.pem -out cert.cert
   ```

7. Copy the following files to the `data` directory of the PostgreSQL installation:
   - `cp cert.pem $PGDATA/server.key`
   - `cp cert.cert $PGDATA/server.crt`

     where `$PGDATA` = `/opt/netiq/idm/apps/postgresql/data/`
8. To change the permission of the files, navigate to the /opt/netiq/idm/apps/postgresql/data/ directory and enter the following commands:

   ```
   chown postgres:postgres server.key
   chown postgres:postgres server.crt
   chmod 600 server.key
   ```

9 In a text editor, change the SSL setting in the `$PGDATA/postgresql.conf` file to `on`. For example:

```
ssl=on
ssl_cert_file = '/opt/netiq/idm/apps/postgresql/data/server.crt' # (change
requires restart)
ssl_key_file = '/opt/netiq/idm/apps/postgresql/data/server.key'  # (change
requires restart)
```

10 Save and close the file.

11 Start Postgres.

12 (Optional) To verify that SSL communication is enabled for Postgres, complete the following steps:

12a Enter `$ ./opt/netiq/idm/apps/postgres/bin/psql -U postgres -h localhost`.

12b Verify that the output is similar to the following content:

```
psql (9.0.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.
```

13 Add the `server.crt` that you created in Step 7 on page 22 to the `cacert`. For example, enter the following command:

```
keytool -import -trustcacerts -alias ar -file server.crt -keystore /opt/netiq/
idm/apps/jre/lib/security/cacerts
```

14 Start Tomcat.

15 Ensure that you update the Entity Data Model databases to recognize the secured connection.

For more information, see "Enabling the Entity Data Model Databases for SSL Communication" on page 24.

Preparing Oracle for SSL Communication

To enable SSL in Oracle, you must have a certificate for the Oracle Server signed by a certificate authority (CA).

1 Download and unpack the SSL helper scripts named `ssl.ca-0.1.tar.gz`.

2 Create a certification request using Oracle Wallet Manager (`/opt/oracle/product/11gR1/db/owm`) using the following commands:

```
su –oracle
owm
```

3 Select **Wallet > New**.

4 Enter your password, then select **Yes** to create folders for the wallet.

5 Fill in the requested information, then select **OK**.

6 Highlight the certification request, then select **Operations > Export Certificate Request**.

7 Save the file with the extension `.csr` in the folder where you extracted `ssl.ca-0.1.tar.gz` then save the wallet.

8 Create a self-signed `root` certificate by running the `new-root-ca.sh` script in the `ssl.ca-0.1` folder that you extracted in the previous step to create a file called `ca.crt`.

9 To run the script that creates the self-signed server certificate, enter the following command:

```
./sign-server-cert.sh CerReq
```

10 Import the ca.crt into the Oracle wallet as a trusted certificate and import the *certificate-request-filename.crt* as a user certificate.

11 Enable auto-login and save the wallet so that it is now ready for use.

12 To configure Oracle advanced security and listener configuration on the database server, run the following commands:

```
su – oracle
netmgr
```

13 Select **Profile > Select Network Security > SSL**.

14 Ensure that the sqlnet.ora and listener.ora files mention the WALLET.

15 (Conditional) If the SSL_CLIENT_AUTHENTICATION parameter is not set, the default setting is TRUE and clients are required to present a certificate during the SSL handshake. If you do not need client authentication, disable it with the following parameter added to the end of the $TNS_ADMIN/listener.ora and $TNS_ADMIN/sqlnet.ora files: SSL_CLIENT_AUTHENTICATION=FALSE

16 Restart the listener:

```
lsnrctl stop
lsnrctl start
```

17 Ensure that you update the Entity Data Model databases to recognize the secured connection.

For more information, see "Enabling the Entity Data Model Databases for SSL Communication" on page 24.

Enabling the Entity Data Model Databases for SSL Communication

To use TLS/SSL connections, the three Entity Data Model databases need the server certificate information. This section applies to both Oracle and PostgreSQL platforms.

1 Enable SSL functionality in the database platform.

For more information, see "Preparing PostgreSQL for SSL Communication" on page 22 or "Preparing Oracle for SSL Communication" on page 23.

2 Log in to the server where you deployed Entity Data Model.

3 Stop Tomcat:

```
/etc/init.d/idmapps_tomcat_init stop
```

4 Add the SSL server certificate that you created for the database platform to the cacert. For example:

PostgreSQL

```
keytool -import -trustcacerts -alias ar -file server.crt -keystore /opt/
netiq/idm/apps/jre/lib/security/cacerts
```

Oracle

```
keytool -import -trustcacerts -alias aroracle -file ca.crt -keystore /opt/
netiq/idm/apps/jre/lib/security/cacerts
```

5 In a text editor, open the server.xml file.

6 For the three Entity Data Model databases listed in the file, specify the URL for the SSL server certificate. For example:

PostgreSQL

```
url="jdbc:postgresql://hostname:5432/database_username?ssl=true"
```

Oracle

```
url="jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS = (PROTOCOL = TCPS)(HOST =
hostname)(PORT = 2484))(CONNECT_DATA =(SERVER = DEDICATED) (SERVICE_NAME =
name))(SECURITY=(SSL_SERVER_CERT_DN='CN=OracleDB,OU=IN,O=IN,L=IN,ST=IN,C=I
N')))"
```

By default, the databases have the usernames `arops`, `ardcs`, and `arwf`.

7 Start Tomcat:

```
/etc/ini.d/idmapps_tomcat_init start
```

8 Ensure that you update the

to recognize the secured connection.

For more information, see “Enabling the Entity Data Model Driver for SSL Communication” on page 25.

Enabling the Entity Data Model Driver for SSL Communication

The Entity Data Model driver can communicate securely with the Entity Data Model databases. Ensure that you also enable SSL communication in the databases. For more information, see “Preparing PostgreSQL for SSL Communication” on page 22 or “Preparing Oracle for SSL Communication” on page 23.

1 Log in to the server where you installed the Entity Data Model driver and the Remote Loader.

2 Stop the Remote Loader. For example, enter the following command:

```
rdxml -config /home/ARShim.conf -u
```

3 In a text editor, open the Remote Loader `conf` file for the driver, by default `ARshim.conf`.

4 Add the content of the SSL server certificate to the file. For example:

PostgreSQL

```
-description ARDriver
-commandport 8000
-connection "port=8090 rootfile=path/server.crt"
-trace 5
-tracefile "/opt/netiq/ar.log"
-tracefilemax 100M
-class "com.novell.nds.dirxml.driver.arshim.AccessReviewDriverShim"
```

Oracle

```
-description ARDriver
-commandport 8000
-connection "port=8090 rootfile=path/ca.crt"
-trace 5
-tracefile /tmp/remoteloader.log
-class com.novell.nds.dirxml.driver.arshim.AccessReviewDriverShim
```

5 Save and close the file.

6 Add the server certificate to the Remote Loader Java certs. For example:

PostgreSQL

```
keytool -import -trustcacerts -alias ar -file server.crt -keystore /opt/
novell/eDirectory/lib64/nds-modules/jre/lib/security/cacerts
```

Oracle

```
keytool -import -trustcacerts -alias aroracle -file ca.crt -keystore /opt/
novell/eDirectory/lib64/nds-modules/jre/lib/security/cacerts
```

7 Start the Remote Loader. For example, enter the following command:

```
rdxml -config /home/ARShim.conf
```

8 In the AR Driver configuration, verify that the setting for **Entity Data Model Database Connection URL** resembles one of the following values:

PostgreSQL

```
url="jdbc:postgresql://hostname:5432/database_username?ssl=true"
```

Oracle

```
jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS = (PROTOCOL = TCPS)(HOST =
hostname)(PORT = 2484))(CONNECT_DATA =(SERVER = DEDICATED) (SERVICE_NAME =
name))(SECURITY=(SSL_SERVER_CERT_DN='CN=OracleDB,OU=IN,O=IN,L=IN,ST=IN,C=I
N')))
```

By default, the databases have the usernames `arops`, `ardcs`, and `arwf`.

9 Restart the Entity Data Model driver.