Informatica

Informatica® Multidomain MDM
10.5 HotFix 2

Upgrading from Version 10.1, 10.2, 10.3, 10.4, or 10.5

Informatica Multidomain MDM Upgrading from Version 10.1, 10.2, 10.3, 10.4, or 10.5
10.5 HotFix 2
October 2022

Publication Date: 2023-11-08

Table of Contents

Preface

Follow the instructions in the Informatica® *Multidomain MDM Upgrade Guide* to upgrade your Multidomain MDM implementation to the most recent version. When you upgrade, ensure that you use the *Multidomain MDM Upgrade Guide* that applies to the currently installed version.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Upgrade Overview

This chapter includes the following topics:

Introduction

Thank you for upgrading Informatica Multidomain MDM. You can upgrade directly from the installed version to this version of Multidomain MDM.

Important: You must upgrade all the MDM components to the same version of Multidomain MDM.

A Multidomain MDM environment can include a development environment, test environment, and production environment. You must upgrade each of these environments. As a best practice, upgrade your development environment first. Identify and resolve any upgrade issues. After you successfully upgrade the development environment, you can upgrade the test and production environments with a high degree of confidence.

Before You Begin

Before you begin, ensure that you have the necessary guides and that you review the Product Availability Matrix:

- Ensure that you have the correct upgrade guide for your installed version of Multidomain MDM. On the title page, check the major version number in the title *Upgrading from Version*. Make sure it matches the major version number in the installed product. To find the version number in the installed product, in the MDM Hub Console click **Help** > **About**. Click **Installation Details**. The version number appears in the release name column. For the purposes of the upgrade, it does not matter whether hotfixes have been applied to the installed product.
- Download the *Multidomain MDM Installation Guide* that applies to your database and application server. To ensure a smooth upgrade, you must perform some of the tasks that are described in the installation guide.
- Review the Product Availability Matrix to learn about changes to the system requirements for this version of Multidomain MDM. You can find the matrix on Informatica Network:

  https://network.informatica.com/community/informatica-network/product-availability-matrices.

Select an Upgrade Process

You have the following options for upgrading Multidomain MDM:

Clean Upgrade

In a clean upgrade, you prepare a new environment with machines that meet the current MDM system requirements. You clone the MDM Hub Master Database and Operational Reference Stores from the existing environment, copy the cloned databases to the new environment, and run the Hub Store upgrade. You install the Hub Server and Process Servers in the new environment by following the instructions in the *Installation Guide*. For information about planning an infrastructure, see the *Multidomain MDM Infrastructure Planning Guide*.

In-place Upgrade

For an in-place upgrade, you update the machines that run Multidomain MDM to be compliant with the current MDM system requirements. Then you upgrade the MDM components.

Guidelines for Selecting an Upgrade Process

The type of upgrade determines which type of upgrade process you can choose.

The following table lists the types of upgrades and tells you which upgrade process you can use:

Type of Upgrade for Multidomain MDM	Clean Upgrade	In-place Upgrade
Upgrade to any version of MDM where the application server has to be updated to a major version to meet the MDM system requirements	Yes	No
Upgrade to any version of MDM where the application server does not need to be updated to a major version to meet the MDM system requirements	Yes	Yes

Upgrade Process for a Clean Upgrade

The process for a clean upgrade consists of the following phases:

Phase	Task	Location of Instructions
1	Database tasks	Upgrade Guide
2	Application server tasks	Upgrade Guide
3	Pre-upgrade tasks	Upgrade Guide
4	Back up and clone the Hub Store, and then copy the Hub Store to the new machines	Ask your DBA to perform this task
5	Hub Store upgrade	Upgrade Guide
6	Hub Server installation and post-installation tasks	Installation Guide
7	Process Server installation and post-installation tasks	Installation Guide
8	Post-upgrade tasks	Upgrade Guide

Phase	Task	Location of Instructions
9	ActiveVOS post-installation tasks for the application server	Installation Guide
10	ActiveVOS post-upgrade tasks for the business entity adapter	Upgrade Guide
11	ActiveVOS post-upgrade tasks for the subject area adapter	Upgrade Guide
12	Resource Kit installation tasks and, if applicable, the Resource Kit post-installation tasks	Installation Guide

Upgrade Process for an In-place Upgrade

The process for an in-place upgrade consists of the following phases:

Phase	Task	Location of Instructions
1	Database tasks	Upgrade Guide
2	Application server tasks	Upgrade Guide
3	Pre-upgrade tasks	Upgrade Guide
4	Hub Store upgrade	Upgrade Guide
5	Hub Server upgrade	Upgrade Guide
6	Process Server upgrade	Upgrade Guide
7	Resource Kit upgrade	Upgrade Guide
8	Post-upgrade tasks	Upgrade Guide
9	ActiveVOS post-installation tasks for the application server	Installation Guide
10	ActiveVOS Post-Installation tasks for the business entity adapter	Upgrade Guide
11	ActiveVOS Post-Installation tasks for the subject area adapter	Upgrade Guide

* If there are multiple releases between your installed version and the current version, the best practice is to either do a clean upgrade or review all the pre-installation tasks to ensure that you do not miss any changes that have occurred for the database or application server that you use.

Document the Upgrade

You must capture the details of the Informatica MDM Hub environment before, during, and after you upgrade to verify and, if required, troubleshoot the upgrade. You can provide copies of this information to Informatica Global Customer Support if you require assistance with troubleshooting the upgrade.

The following table describes the sources of upgrade information:

Upgrade Information	Source of Information
Operational Reference Store (ORS) metadata	Validation results from the Repository Manager tool in the Hub Console Generate the validation results for the ORS metadata before and after the upgrade.
Configuration settings	Environment report from the Enterprise Manager tool in the Hub Console The Environment report documents the Hub Server, Process Server, Master Database, and ORS database information. Save the Environment report before and after the upgrade.
Upgrade events for the Hub Server and Process Server upgrades	Informatica MDM Hub installer log files
Hub Server and Process Server information	Hub Server and Process Server log files
Hub Console information	Hub Console log files

Related Topics:

- "Saving the MDM Hub Environment Report" on page 122

CHAPTER 2

Pre-Upgrade Tasks

This chapter includes the following topics:

Pre-Upgrade Tasks

Whether you perform a clean upgrade or an in-place upgrade, perform the pre-upgrade tasks to ensure your environment is properly configured.

Prepare for the Upgrade

Perform the following tasks to prepare for the upgrade:

Upgrade Task	Details
Read the Release Notes	The Release Notes contain information about updates to the installation and upgrade process. **Important:** Some versions of application servers and databases have known limitations when running Multidomain MDM. Ensure that you perform all suggested workarounds.
Read the Release Guide	The Release Guide contains information about new features and changed behavior.
Get the latest license file	Request the latest license file when you request the upgrade software for Multidomain MDM.

Upgrade Task	Details
Review the upgrade restrictions	Review the following conditions before you begin the upgrade: - All the components of the Multidomain MDM implementation must be the same version. If you have multiple versions of Multidomain MDM, each version must be installed in a separate environment. Different versions of Multidomain MDM cannot coexist in the same environment. - You must not make any major changes to your environment while you upgrade Multidomain MDM. For example, do not upgrade the Oracle, IBM Db2, or Microsoft SQL Server database during the upgrade process. - You must upgrade your Operational Reference Store (ORS) using the upgrade scripts provided. The Repository Manager is not intended to be used as an upgrade tool because some of the artifacts might not be carried over or might be carried over incorrectly from one version to another. For more information, see the *Multidomain MDM Release Notes*.
Create an upgrade documentation folder	Create a folder named `upgradedoc` to store copies of all your upgrade artifacts such as metadata validation results, environment reports, and log files. If you encounter problems during the upgrade, you need to send a copy of this directory to Informatica Global Customer Support for analysis.
Review the MDM Hub Environment Report	Use the Enterprise Manager tool in the Hub Console to review the current MDM Hub configuration for Hub Servers, Process Servers, the MDM Hub Master Database, and Operational Reference Store databases. Also note the version history of the components. Save a copy of the environment report in the upgrade documentation folder.

Prepare the Environment

Perform the tasks for the upgrade process you chose.

Clean Upgrade

Perform the following tasks for a clean upgrade:

Task	Description
Verify minimum system requirements	Verify that the machines meet the hardware and software requirements for the MDM Hub installation. The hardware requirements are dependent on the data, processing volumes, and business rules. To install the MDM Hub, the machines must meet the following minimum requirements: - Disk space. 4.9 GB - RAM for the development environment. 4 GB To verify the run-time physical memory requirements of the MDM Hub components, use the following formula: `Total run-time memory requirement for MDM Hub components = JDK JVM max heap size of the application server + operating system native heap size`
Install Java Development Kit (JDK)	Install a supported version of the JDK on the machine on which you want to install the MDM Hub. The JDK is not bundled with the MDM Hub installers. In a JBoss environment, you must install the version of Azul Zulu JDK that Informatica provides. Contact Informatica Shipping to download Azul Zulu JDK. **Note:** Use the same Java version on the application server machines and on the machines on which you want to launch the Hub Console.

Task	Description
Install Visual C++ Redistributable for Visual Studio 2019 on Windows only	On Windows systems, Multidomain MDM requires Visual C++ Redistributable for Visual Studio 2019 to support the name search feature and the matching feature.
Set environment variables	Set the environment variables for the MDM Hub installation. To use the correct JDK, set the following environment variables to point to the JDK directory: - JAVA_HOME. Required - PATH. Required To set the correct locale behavior for Oracle, set the NLS_LANG environment variable for client application environments such as the Oracle loader, and the MDM Hub components. Specify the NLS_LANG parameter in the following format: `NLS_LANG = <language>_<territory>.<character set>` **Note:** To store and search for records that contain Chinese, Japanese, Korean, or accented characters, set the character set to UTF-8. For more information about NLS_LANG settings, see the Oracle documentation.
Set the operating system locale	Set the same operating system locale for the Hub Server, the MDM Hub Master Database, Operational Reference Store, and the Hub Console.
Set up the X Window System on UNIX	If you want to run the installer in graphical mode on UNIX, set up an X Window System. An X Window System is a graphics display server. For more information about setting up an X Window System, see the documentation for your operating system.
Disable access to the root welcome page for your application server	To improve security, disable access to the root welcome page for your application server. For instructions, see the documentation for your application server.
Disable insecure TLS cipher suites	To improve security, in the Java runtime environment that is used with Multidomain MDM, disable insecure TLS cipher suites. 1. Open the following file: `../jdk<version>/jre/lib//security/java.security` 2. Find the property `jdk.tls.disabledAlgorithms` and update the value to include the following list of insecure cipher suites: `jdk.tls.disabledAlgorithms = SSLv3, RC4, MD5withRSA, DH keySize < 1024, EC keySize < 224, DES40_CBC, RC4_40, 3DES_EDE_CBC, EDH-RSA-DES-CBC3-SHA, ECDHE-RSA-DES-CBC3-SHA, DES-CBC3-SHA` For more information about the property, see the documentation for your JDK.

Perform the following additional tasks for a clean upgrade:

Task	Details
Validate the metadata	Ensure the Operational Reference Stores (ORS) do not have any validation errors. If you upgrade the Hub Store when an ORS contains metadata that is not valid, the upgrade might generate unexpected results. Use the Repository Manager in the Hub Console to validate metadata. Resolve any validation issues and then validate the metadata again to confirm that you have resolved the validation issues. Save a copy of the final validation results in the `upgradedoc` upgrade documentation folder. Use the Repository Manager tool in the MDM Hub Console to validate and then save the validation results.
Update persistent ID implementation	If you use persistent IDs, contact Informatica Global Customer Support. You must update the persistent ID implementation to be compatible with the upgraded version of Multidomain MDM.

In-place Upgrade

Perform the following tasks for an in-place upgrade:

Task	Details
Update the MDM environment to meet system requirements	You might need to update the operating system, application server, JDK, and database server. For system requirements, see the Product Availability Matrix for this version of Multidomain MDM (MDM) on Informatica Network: https://network.informatica.com/community/informatica-network/product-availability-matrices/overview . In a JBoss environment, you must install the version of Azul Zulu JDK that Informatica provides. Contact Informatica Shipping to download Azul Zulu JDK. Update the machines that run MDM to be compliant with the current MDM system requirements.
Install Visual C++ Redistributable for Visual Studio 2019 on Windows only	On Windows systems, Multidomain MDM requires Visual C++ Redistributable for Visual Studio 2019 to support the name search feature and the matching feature.
Validate the metadata	Ensure the Operational Reference Stores (ORS) do not have any validation errors. If you upgrade the Hub Store when an ORS contains metadata that is not valid, the upgrade might generate unexpected results. Use the Repository Manager in the Hub Console to validate metadata. Resolve any validation issues and then validate the metadata again to confirm that you have resolved the validation issues. Save a copy of the final validation results in the `upgradedoc` upgrade documentation folder. Use the Repository Manager tool in the MDM Hub Console to validate and then save the validation results.
Update persistent ID implementation	If you use persistent IDs, contact Informatica Global Customer Support. You must update the persistent ID implementation to be compatible with the upgraded version of Multidomain MDM.

Task	Details
Back up the implementation	Back up your current implementation to retain your customizations and to allow you to restore the environment. If you encounter issues during the upgrade, you can restore your environment from the backup. **Back Up the Schema** Perform a full back up of the schema. You cannot roll back schema changes that the upgrade process makes. If you encounter upgrade issues, you can restore the schema from the backup. To back up your schema, see the documentation for the database. **Back up the Hub Server and Process Server installed directories** Before you install the Hub Server and Process Servers, back up the Hub Server and Process Server installation folders in your environment. **Back Up Your Data Director Applications** For information about exporting Data Director applications with subject areas, see the *Multidomain MDM Data Director Implementation Guide*. **Register or Back Up Customizations** Customizations that you register in the Hub Console such as custom queries, custom functions, and user exits are maintaining during the upgrade process. Back up the latest source code of customizations that you do not register in the Hub Console. Unregistered customizations might be unavailable after you upgrade. **Back Up Customized Cleanse Engine Configuration Files** Back up any cleanse engine configuration files that you have customized.
Configure the JAVA_HOME environment variable	JBoss or WebLogic. If you apply the upgrade to the existing installation directory, perform the following steps: 1. In a JBoss environment, manually remove the JAVA_HOME environment variable setting from the following file: - Windows: `<MDM Hub installation directory>\setSiperianEnv.bat` - UNIX: `<MDM Hub installation directory>/setSiperianEnv.sh` 2. In a JBoss or WebLogic environment, set the JAVA_HOME environment variable to the path of a supported JDK version.
Disable access to the root welcome page for your application server	To improve security, disable access to the root welcome page for your application server. For instructions, see the documentation for your application server.

Process Existing Jobs

Perform the following tasks to process existing jobs:

Upgrade Task	Details
Run load job on staging tables that contain records	In Microsoft SQL Server environments, run the load batch job on staging tables that contain records. If you upgrade the Operational Reference Store when the staging tables contain records, the upgrade can fail because the log file size can exceed the available hard drive space.
Complete stage jobs and delete stage table contents	Before you upgrade the Operational Reference Store, complete any stage jobs that are in progress, and then delete the contents of the stage table. If you do not delete the contents of the stage table, the Operational Reference Store upgrade takes longer than expected to complete in Microsoft SQL Server environments.

Update the MDM Hub Configuration

Perform the following tasks to update the MDM Hub configuration.

Upgrade Task	Details
Grant select right on SYS.V_$PARAMETER for the ORS user	To grant the select right, run the following SQL statement: `grant select on SYS.V_$PARAMETER to <Operational Reference Store user>;`
Remove special characters from names in the subject area configuration	You cannot use special characters in names in the Provisioning tool. If you plan to generate a business entity schema from a subject area configuration, you must first remove any special characters from the names in the subject area configuration.
Ensure column names do not contain reserved words	Additional keywords have been marked as reserved in recent versions of Multidomain MDM. Ensure you do not use reserved keywords in your implementation. For a complete list of reserved words, refer to the "Requirements for Defining Schema Objects" section of the *Multidomain MDM Configuration Guide*. To request a script that changes the name of a column that contains data, contact Informatica Global Customer Support.
Ensure mapped columns have compatible data types	If you use an Oracle database for your Operational Reference Store, check the mappings from landing table to staging table to ensure that the data types of the mapped columns are compatible. In previous versions, you can create a mapping from a landing table to a staging table where the data types of the mapped columns are not compatible. In this version, the data types of the mapped columns must be compatible. When you load data, if there is a data type mismatch, the load fails with an error in the log files. For more information about mapping columns, see the *Multidomain MDM Configuration Guide*.

Upgrade Task	Details
Back up your user preferences from the C_REPOS_DS_PREF table and the C_REPOS_DS_PREF_DETAIL table	If your Data Director environment includes user preferences for shown or hidden columns, the settings are lost when you upgrade because the cryptographic hash algorithm has changed in this version. After you upgrade, recreate your user preferences. For more information about Data Director global properties, see the *Multidomain MDM Data Director Implementation Guide*.
Ensure users have a valid email address	Ensure that existing users have valid email addresses associated with their user names in the Hub Console. To reset their passwords to access the MDM Hub, users must have a valid email address. For more information about editing user information in the Hub Console, see the *Multidomain MDM Security Guide*. **Note:** You cannot change the email address for the `admin` user in the Hub Console. To change the email address for the admin user, update the admin user entry directly in the C_REPOS_USER table under CMX_SYSTEM schema.
Register indexes	If you upgrade schemas with custom indexes, you must register the custom indexes. Use the RegisterCustomIndex SIF API to register the custom indexes. For more information about the RegisterCustomIndex SIF API, see the *Multidomain MDM Services Integration Framework Guide*.

Register the Operational Reference Stores with the Original Schema Owner

If any of the Operational Reference Store (ORS) databases are registered with a proxy user, register the ORS database with the original schema owner. You must register the ORS with the original schema owner to ensure that you have sufficient privileges to perform the upgrade operations. You can reregister the ORS with the proxy user after you upgrade. The original default schema owner is CMX_ORS.

1. Start the Hub Console.

   The **Change database** dialog box appears.
2. Select the MDM Hub Master database, and click **Connect**.
3. Select the **Databases** tool from the **Configuration** workbench.
4. Select **Write Lock** > **Acquire Lock**.
5. Click the **Register database** button.

   The **Informatica MDM Hub Connection Wizard** appears and prompts you to select the database type.
6. Select the type of database, and click **Next**.

7. Configure connection properties for the database.
   a. Select an Oracle connection method, and then click **Next**.

      You can select the following Oracle connection methods:

      Service

      Connect to Oracle by using the service name.

      SID

      Connect to Oracle by using the Oracle System ID.

      For more information about SERVICE and SID names, see the Oracle documentation.

      The **Connection Properties** page appears.

   b. Specify the connection properties for the connection type that you select, and click **Next**.

      You can configure the following connection properties:

      Database Display Name

      Name for the Operational Reference Store that must appear in the Hub Console.

      Machine Identifier

      Prefix given to keys to uniquely identify records from the Hub Store instance.

      Database host name

      IP address or name of the server that hosts the Oracle database.

      SID

      Oracle System Identifier that refers to the instance of the Oracle database running on the server. The **SID** field appears if you selected the **SID** connection type.

      Service

      Name of the Oracle SERVICE used to connect to the Oracle database. The **Service** field appears if the you selected the **Service** connection type.

      Port

      The TCP port of the Oracle listener running on the Oracle database server. The default is `1521`.

      Oracle TNS Name

      Name by which the database is known on your network as defined in the `TNSNAMES.ORA` file of the application server.

      For example: `mydatabase.mycompany.com`.

      You set the Oracle TNS name when you install the Oracle database. For more information about the Oracle TNS name, see the Oracle documentation.

      Schema Name

      Name of the Operational Reference Store.

      User name

      Specify the original user name for the ORS. Default is CMX_ORS.

      Password

      Password associated with the original user for the ORS.

Note: The default password `ChangeMe` is displayed as *******, when `cmx.server.database.authentication.method=windowsauthentication` in the `cmxserver.properties` file. The application will use the Windows authentication to connect to the data source.

The **Summary** page appears.

c. Review the summary, and specify additional connection properties.

The following table describes the additional connection properties that you can configure:

Property	Description

Connection URL	Connect URL. The Connection Wizard generates the connect URL by default. The following list shows the format of the connect URL for the Oracle connection types:
	Service connection type
	`jdbc:oracle:thin:@//database_host:port/service_name`
	SID connection type
	`jdbc:oracle:thin:@//database_host:port:sid`
	For a service connection type, you have the option to customize and later test a different connection URL.

Create datasource after registration	Select to create the datasource on the application server after registration. **Note:** If you do not select the option, you must manually configure the data source.

d. For a service connection type, if you want to change the default URL, click the **Edit** button, specify the URL, and then click **OK**.

8. Click **Finish**.

The **Registering Database** dialog box appears.

9. Click **OK**.

The MDM Hub registers the ORS.

Prepare the BPM Upgrade

To upgrade the business process management system, start by choosing a suitable workflow adapter. If you use standalone ActiveVOS, you must migrate to embedded ActiveVOS and edit the `build.properties` file. If ActiveVOS is installed in your environment, follow the upgrade tasks to edit the `build.properties` file. Informatica supports ActiveVOS 9.2.4.6 only.

Choose a Workflow Adapter

Review the following upgrade options for workflow adapters, and choose the upgrade option that suits your needs.

The following table describes current workflow adapters and their upgrade options:

Current Workflow Adapter	Upgrade Options
SIPERIAN	Deprecated. **Option 1** Keep SIPERIAN as the primary workflow adapter. **Option 2** Upgrade to the BE ActiveVOS workflow adapter based on business entities. You must have a business entity configuration to use the BE ActiveVOS workflow adapter. For more information, see *Migrating IDD Applications to the Business Entity Data Model*.
Informatica ActiveVOS	This workflow adapter is based on subject areas and operates using SIF APIs. **Option 1** Keep Informatica ActiveVOS as the primary workflow adapter. **Option 2** Upgrade to the BE ActiveVOS workflow adapter based on business entities. You must have a business entity configuration to use the BE ActiveVOS workflow adapter. For more information, see the *Multidomain MDM Data Director Migration Guide*.
BE ActiveVOS	This workflow adapter is based on business entities and operates using business entity services. You must use the version of ActiveVOS that is defined in the Product Availability Matrix. If an older version is detected in your environment, the upgrade process installs the required version of ActiveVOS.

Migrate from Standalone ActiveVOS to Embedded ActiveVOS (In-place Upgrade)

If you use standalone ActiveVOS, you must migrate to embedded ActiveVOS. Standalone ActiveVOS is not supported.

The following table describes the pre-upgrade tasks required to migrate from standalone ActiveVOS to embedded ActiveVOS:

Upgrade Task	Description
Remove the ActiveVOS data sources	If the standalone ActiveVOS Server is in the same JBoss instance, WebSphere profile, or WebLogic domain as Multidomain MDM, remove the data source for ActiveVOS from the application server.
Edit the URN mapping in a WebSphere environment	1. Launch the ActiveVOS Console. In a browser, type the following URL, substituting the correct host name and port number: - Secure connections. `https://<host>:<port>/activevos` - Non-secure connections. `http://<host>:<port>/activevos` 2. In the ActiveVOS Console, on the **Home** page, click **Administration** > **Configure Server** > **URN Mappings**. 3. Ensure that URN `java:comp/env/jdbc/ActiveVOS` maps to URL `java:comp/env/jdbc/ActiveVOS`.

Important: When you run the Hub Server installer as part of the upgrade process, ensure that you install the embedded ActiveVOS.

Edit the Build Properties File (In-place Upgrade)

If ActiveVOS is installed in your MDM environment, you must perform upgrade tasks to edit the `build.properties` file.

The following table describes the upgrade tasks required to edit the `build.properties` file:

Upgrade Task	Details

Configure the database connection properties for IBM Db2	In IBM Db2 environments, before you upgrade, add the ActiveVOS database connection properties to the `build.properties` file in `<MDM Hub installation directory>/hub/server/bin`.
	The following sample shows the ActiveVOS database connection properties with sample entries:
	`activevos.db.type=db2`
	`activevos.db.server=localhost`
	`activevos.db.port=50000`
	`activevos.db.user=AVOS`
	`activevos.db.dbname=INFA102`
	`activevos.db.schemaname=AVOS`
	`activevos.db.jdbc.url=jdbc:db2://localhost:50000/INFA102`
	`activevos.b4p.url=http://localhost:9080/active-bpel/services/AeB4PTaskClient-taskOperations`

Configure the ActiveVOS installation directory	If the upgrade process detects that the installed version of ActiveVOS does not meet the system requirements, the upgrade process installs the required version of ActiveVOS in a new directory. To install ActiveVOS in the same directory as a previous installation, delete or comment out the following entry in the build.properties file: `activevoslinstall.dir=<ActiveVOS installation directory>`
Configure the ActiveVOS settings	Applicable for JBoss environments only. If the ActiveVOS settings property `sip.appserver.web.url` is not set to the default port number `8080`, you must manually update the property and the port number to the following: `-Dsip.appserver.web.url= http://localhost:8079` You can set the port number to `8079` or another unused port number. Update the properties in the `build.properties` file in `<MDM Hub installation directory>/hub/server/bin`.

CHAPTER 3

Database Tasks

This chapter includes the following topics:

Database Tasks Overview

Configure your database to work with Multidomain MDM. Follow the instructions for your database.

When you perform a clean upgrade, perform all the steps for your database as you would for a new installation of Multidomain MDM.

When you perform an in-place upgrade where your database version is still supported, verify that the database is configured to work with this version of Multidomain MDM.

Set Up Oracle Database

Before you create an MDM Hub Master Database and Operational Reference Store, set up the database environment.

To set up the database environment, perform the following tasks:

1. Install and configure Oracle.
2. Set up a database instance.
3. Set up database privileges and connections.
4. Create the ActiveVOS schema.

Step 1. Install and Configure Oracle

You can install and configure Oracle Database according to the instructions in the Oracle documentation.

The following table describes the Oracle installation and configuration tasks that you must perform on each machine on which you want an Oracle instance :

Tasks	Description
Install Oracle	Install the supported version of Oracle Database. **Note:** If you want to use the Oracle multitenant feature, set up a pluggable database (PDB) for the MDM Hub installation.
Install clients and utilities	Install the Oracle client and utilities software to communicate with the MDM Hub, and to run the MDM Hub processes. On each machine where you want to run the Hub Server or Process Server, install the following software: - Oracle client - SQL*Loader
Disable recycle bins	Disable the recycle bins, USER_RECYCLEBIN and DBA_RECYCLEBIN. Recycle bins can interfere with the MDM Hub processes. Disable the recycle bins at the system or session level. Also, purge existing objects in the recycle bin.
Set initialization parameters	Configure the Oracle initialization parameters in the `init.ora` file. For information about Oracle initialization parameters, see the *MDM Multidomain Edition Performance Tuning Guide*.

For information about installing and configuring Oracle, see the Oracle documentation.

Step 2. Create a Database and Tablespaces

After you install and configure Oracle Database, create and configure databases and tablespaces.

The following table describes the tasks that you need to perform to configure databases:

Tasks	Description
Create a database	Create a database for each database instance.
Create tablespaces	Create tablespaces for the MDM Hub data. Adjust the default tablespace sizes and the number of data files based on the volume of data that you want to load into the MDM Hub. Create the following tablespaces: - CMX_DATA. Contains the metadata and user data of the MDM Hub. - CMX_INDX. Contains indexes that the MDM Hub creates and uses. - CMX_TEMP. Contains temporary tables for the MDM Hub. - BPM_DATA. Contains the ActiveVOS data for business process management. **Note:** If you want to create multiple MDM Hub Master Databases, create unique tablespaces for each MDM Hub Master Database.

Creating Tablespaces on Premises

If you use Oracle on premises, create tablespaces as permanent tablespaces that you manage locally or configure manually.

Note: Create bigfile tablespaces to simplify database management for large data loads. However, you can create smallfile tablespaces if required.

1. Log in to Oracle as a database administrative user such as SYSTEM.
2. Create the tablespaces.

   The following table contains sample SQL statements to create tablespaces:

Tablespace Name	Sample SQL Statement
CMX_DATA	**Note:** Do not change the default tablespace name to prevent Repository Manager validation errors. `CREATE BIGFILE TABLESPACE CMX_DATA NOLOGGING DATAFILE '<Oracle install directory>/CMX_DATA1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;`
CMX_INDX	`CREATE BIGFILE TABLESPACE CMX_INDX NOLOGGING DATAFILE '<Oracle install directory>/CMX_INDX1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;`
CMX_TEMP	`CREATE BIGFILE TABLESPACE CMX_TEMP NOLOGGING DATAFILE '<Oracle install directory>/CMX_TEMP1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;`
BPM_DATA	`CREATE BIGFILE TABLESPACE BPM_DATA NOLOGGING DATAFILE '<Oracle install directory>/BPM_DATA1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;`

Creating Tablespaces in Amazon Relational Database Service

If you use Amazon Relational Database Service (RDS) for Oracle, create tablespaces as permanent tablespaces in Amazon RDS.

Note: Create bigfile tablespaces to simplify database management for large data loads. However, you can create smallfile tablespaces if required.

1. Log in to Amazon RDS for Oracle as a database administrative user.
2. Create the tablespaces.

The following table contains sample SQL statements to create tablespaces:

Default Tablespace Name	Sample SQL Statement

CMX_DATA	`CREATE BIGFILE TABLESPACE CMX_DATA`
	`DATAFILE SIZE 2048M`
	`AUTOEXTEND ON NEXT 2048M;`
	Do not change the default tablespace name to prevent Repository Manager validation errors.

CMX_INDX	`CREATE BIGFILE TABLESPACE CMX_INDX`
	`DATAFILE SIZE 2048M`
	`AUTOEXTEND ON NEXT 2048M;`

CMX_TEMP	`CREATE BIGFILE TABLESPACE CMX_TEMP`
	`DATAFILE SIZE 2048M`
	`AUTOEXTEND ON NEXT 2048M;`

Configuring the Database Environment for Custom Tablespace Names

If you use a tablespace name other than the default tablespace name for `CMX_INDX` or `CMX_TEMP`, disable the `DEFERRED_SEGMENT_CREATION` initialization parameter. You disable the parameter to prevent Repository Manager validation errors.

- To disable `DEFERRED_SEGMENT_CREATION`, run the following SQL statement, and restart the database:

```
ALTER SYSTEM SET DEFERRED_SEGMENT_CREATION=FALSE SCOPE=BOTH;
```

Step 3. Set Up Database Privileges and Connections

Set up database privileges and connections.

The following table describes the tasks that you need to perform to set up database privileges and connections:

Tasks	Description

Grant privileges to the database administrative user	If you want to use a database administrative user to create the MDM Hub Master Database and the Operational Reference Store, grant privileges to the database administrative user. The user must have the privileges to grant options for distributed transactions and `DBMS_LOCK`.
	To grant privileges to the database administrative user, connect to the database as a user with privileges to grant options, and run the following SQL statements:
	`GRANT SELECT ON sys.pending_trans$ TO <DBA user> with grant option;`
	`GRANT SELECT ON sys.dba_pending_transactions TO <DBA user> with grant option;`
	`GRANT SELECT ON sys.dba_2pc_pending TO <DBA user> with grant option;`
	`GRANT EXECUTE ON sys.dbms_xa TO <DBA user> with grant option;`
	`GRANT EXECUTE ON sys.dbms_lock TO <DBA user> with grant option;`

Add the Oracle TNS name	For connections to the Oracle database, add TNS name entries to the `tnsnames.ora` file on the Hub Server and Process Server machines.
	To add TNS name entries, use the following syntax:
	`<TNS NAME> =`
	`(DESCRIPTION =`
	`(ADDRESS_LIST =`
	`(ADDRESS = (PROTOCOL = TCP)(Host = <Oracle server host name>)(Port = <Oracle server port>))`
	`)`
	`(CONNECT_DATA =`
	`(SERVICE_NAME = <Oracle SID>)`
	`)`
	`)`
	Note: The TNS names on the Hub Server and Process Server machines must be the same.

Test the database connection	Test the connection to the database from each machine on which you want to run the Hub Server or the Process Server. In SQL*Plus, use the following SQL statement syntax: `sqlplus <user name>/<password>@<TNS Name>`

Step 4. Create the ActiveVOS Schema

To install ActiveVOS, you need to create the ActiveVOS schema. To create the schema, run the `create_bpm` script.

If you want to create multiple MDM Hub Master Databases, create an ActiveVOS schema for each MDM Hub Master Database.

Note: If you want to use the Oracle multitenant feature, create the ActiveVOS schema in a pluggable database (PDB).

1. Open a command prompt and change to the following directory:

   `<MDM Hub distribution directory>/database/bin`

2. Run the following command:

   On UNIX. `./sip_ant.sh create_bpm`

On Windows. `sip_ant.bat create_bpm`

3. Answer the prompts that appear.

   The prompt displays default text in brackets. Press **Enter** to use the default value and go to the next prompt.

Property	Description
Database Type	The type of database. For an Oracle database, specify `Oracle`. The database type must be the same as the database type selected for the MDM Hub Master Database and the Operational Reference Stores.
Oracle Connection Type	Connection type. Use one of the following values: - SERVICE. Uses the service name to connect to Oracle. - SID. Uses the Oracle System ID to connect to Oracle.
ActiveVOS Database Host Name	Name of the machine that hosts the database.
ActiveVOS Database Port	Port number that the database listener uses.
Database Service Name	Name of the Oracle service. This property is required when the selected Oracle connection type is SERVICE.
Oracle Net Connect Identifier (TNS Name)	Oracle TNS name.
Database SID	Name of the Oracle System ID. This property is required when the selected Oracle connection type is SID.
DBA User Name	User name of the database administrative user.
DBA Password	Password of the administrative user.
ActiveVOS User Name	User name of the ActiveVOS Server administrative user.
ActiveVOS User Password	Password of the administrative user.
ActiveVOS User Tablespace	The name of the tablespace that contains the records that are involved in MDM workflows.
ActiveVOS User Temp Tablespace	The name of the temporary tablespace.

4. After you create the schema, review the `sip_ant.log` file in the following directory:

   `<MDM Hub distribution directory>/database/bin`

   The `sip_ant.log` file logs any errors that might occur when you run the `sip_ant` script to create the ActiveVOS schema.

Set Up Microsoft SQL Server

Before you create an MDM Hub Master Database and Operational Reference Store, set up the database environment.

To set up the database environment, perform the following tasks:

1. Install and configure Microsoft SQL Server.
2. Create a data file store.
3. Install ODBC driver.
4. Create the ActiveVOS schema.

Step 1. Install and Configure Microsoft SQL Server

You can install and configure Microsoft SQL Server according to the instructions in the Microsoft SQL Server documentation.

The following table describes the Microsoft SQL Server installation and configuration tasks:

Tasks	Description
Install Microsoft SQL Server	Install the supported version of Microsoft SQL Server. **Note:** Ensure that you specify mixed mode as the database engine authentication security mode.
Configure JDBC driver	If you want to install Multidomain MDM in an environment that uses Microsoft SQL Server 2017, perform the following tasks: 1. Download the latest supported version of Microsoft JDBC Driver from the Microsoft website. 2. Copy the driver file to the `Binn` directory of the machine on which Microsoft SQL Server is installed.
Configure distributed transactions	Configure Microsoft SQL Server for distributed transactions because the MDM Hub requires a distributed transactions environment. To configure Microsoft SQL Server for distributed transactions, enable MS DTC for XA transactions, and configure the JDBC distributed transactions component.
Enable the TCP/IP network protocol	Configure Microsoft SQL Server to use the TCP/IP network protocol that the MDM Hub environment requires.

Configuring Microsoft SQL Server for Distributed Transactions

Before you start Microsoft SQL Server, ensure that you configure it for distributed transactions. The MDM Hub requires a distributed transaction environment.

1. To ensure that the Microsoft Distributed Transaction Coordinator (MS DTC) service starts when you start Microsoft SQL Server, mark MS DTC as Automatic in the Service Manager.
2. Enable MS DTC for XA transactions.
3. Enable MS DTC for SNA LU 6.2 transactions.
4. Configure the JDBC distributed transactions component.
    a. Download and extract the supported version of the Microsoft SQL Server JDBC driver from the Microsoft website to a directory on the machine.

b. Copy the `sqljdbc_xa.dll` file to the `Binn` directory of the machine on which Microsoft SQL Server is installed.

c. Run the `xa_install.sql` script on the Microsoft SQL Server instance as an administrator.

The `xa_install.sql` script is in the `xa` directory of the JDBC driver.

d. Verify that the script creates the SqlJDBCXAUser role in the Microsoft SQL Server master database.

For more information about how to install and configure Microsoft SQL Server for distributed transactions, see the Microsoft SQL Server documentation.

Enabling the TCP/IP Network Protocol

Use SQL Server Configuration Manager to enable the TCP/IP network protocol, which the MDM Hub environment requires.

1. Start the SQL Server Configuration Manager.
2. In the console pane, expand **SQL Server Network Configuration**, and then click **Protocols for MSSQLSERVER**.
3. In the details pane, right-click **TCP/IP**, and then click **Enable**.
4. Restart the SQL Server service.

   The TCP/IP network protocol is enabled.

Step 2. Create a Microsoft SQL Server Data File Store

Ensure that you create a Microsoft SQL Server data file store if one does not exist. When you create the MDM Hub Master Database and the Operational Reference Stores, you need to provide the path to the Microsoft SQL Server data file store.

Step 3. Install ODBC Driver

If you install the application server on UNIX or Linux, install an ODBC driver for Microsoft SQL Server so that data sources can be created.

▶ Download and install the ODBC driver for the operating system.

- On Linux. Download the Microsoft SQL Server ODBC driver from the Microsoft website.
- On UNIX. Download the UNIX ODBC driver from the unixODBC Project home page.

Step 4. Create the ActiveVOS Database

To install ActiveVOS, you need to create the ActiveVOS database. To create the database, run the `create_bpm` script.

If you want to create multiple MDM Hub Master Databases, create an ActiveVOS database for each MDM Hub Master Database.

1. Open a command prompt and change to the following directory:

   `<MDM Hub distribution directory>/database/bin`

2. Answer the prompts that appear.

The prompt displays default text in brackets. Press **Enter** to use the default value and go to the next prompt.

Property	Description
Database Type	Type of database to use. For Microsoft SQL Server, specify `MSSQL`. The database type must be the same as the database type selected for the MDM Hub Master Database and the Operational Reference Stores.
ActiveVOS User Name	User name of the ActiveVOS Server administrative user. The user name is equivalent to the name of the Microsoft SQL Server database.
ActiveVOS User Password	Password of the administrative user.
ActiveVOS Collation Name	Name of the ActiveVOS database collation. For example, `Latin1_General_CI_AS`.
ActiveVOS Database Host Name	Name of the machine that hosts the database.
ActiveVOS Database Path	Path to the database location.
DBA User Name	User name of the SA user.
DBA Password	Password of the SA user account.

3. After you create the database, review the `sip_ant.log` file in the following directory:

   `<MDM Hub distribution directory>/database/bin`

   The `sip_ant.log` file logs any errors that might occur when you run the `sip_ant` script to create the ActiveVOS database.

Set Up IBM Db2 Database

Before you create an MDM Hub Master Database and Operational Reference Store, set up the database environment.

To set up the database environment, perform the following tasks:

1. Install and configure IBM Db2.
2. Create a database and tablespaces.
3. Bind packages on the database server.
4. Create the ActiveVOS schema.

Step 1. Install and Configure IBM Db2

You can install and configure IBM Db2 according to the instructions in the IBM Db2 documentation.

The following table describes the IBM Db2 installation and configuration tasks that you must perform on each machine on which you want an IBM Db2 instance:

Tasks	Description
Install IBM Db2	Install the supported version of IBM Db2.
Install clients and utilities	Install the IBM Db2 client and utilities software to communicate with the MDM Hub and run the MDM Hub processes. On each machine where you want to run the Hub Server or Process Server, install the following software: - Db2 client - Db2 Java utilities for the Db2 client Ensure that you catalog the IBM Db2 database from each Db2 client.
Configure IBM Db2 drivers	To configure IBM Db2 drivers, copy the `db2jcc.jar` and `db2jcc_license_cu.jar` driver files from the source to the target directory: Source: `<IBM Db2 installation directory>/java` Target: `<MDM Hub distribution directory>/database/lib`
Create the MDM Hub schema users	Create users to access the following MDM Hub schemas: - MDM Hub Master Databases - Operational Reference Stores

Step 2. Create a Database and Tablespaces

After you install and configure IBM Db2, create and configure databases and tablespaces. You must create a database for each database instance.

Note: If you want to create multiple MDM Hub Master Databases, create unique tablespaces for each MDM Hub Master Database.

The following table describes the tablespaces that you require for the MDM Hub schemas:

Tablespace Name	Description
CMX_DATA	Default tablespace for the Operational Reference Store schema. Contains the metadata and user data of the MDM Hub.
CMX_INDX	Tablespace to contain indexes that the MDM Hub creates and uses.
CMX_TEMP	Tablespace to contain temporary tables that the MDM Hub creates and uses.
CMX_REPOS	Tablespace to contain the Operational Reference Store objects.
CMX_USER_TEMP	Temporary tablespace to contain operational temporary tables.
CMX_SYS_TEMP	Temporary tablespace for SQL operations.

Use one of the following procedures to create a database and tablespaces:

- Manually create the database and tablespaces

- Use a script to create the database and tablespaces

Manually Create a Database and Tablespaces

You can manually create a database and tablespaces. Ensure that you create the database with the compatibility vector turned on and with the `UTF-8 TERRITORY US` locale.

Set the Db2 Environment and Db2 Registry Variables

If you create the database manually, set the Db2 environment and Db2 registry variables that the MDM Hub requires.

Use the following commands to set the Db2 environment and Db2 registry variables:

```
db2set DB2CODEPAGE=1208
db2set DB2_COMPATIBILITY_VECTOR=
db2set DB2_DEFERRED_PREPARE_SEMANTICS=YES
db2set DB2_RESTORE_GRANT_ADMIN_AUTHORITIES=ON
db2set DB2_HASH_JOIN=YES
db2set DB2_ANTIJOIN=YES
db2set DB2_INLIST_TO_NLJN=NO
db2set DB2_SELECTIVITY=ALL
db2set DB2_SKIPINSERTED=YES
db2set DB2_SKIPDELETED=YES
db2set DB2_EXTENDED_OPTIMIZATION=ON,ENHANCED_MULTIPLE_DISTINCT,IXOR,SNHD
db2set DB2NTNOCACHE=ON
db2set DB2_REDUCED_OPTIMIZATION=REDUCE_LOCKING
```

Set the Database Manager Configuration for the Database Instance

You need to optimize the database manager configuration for the database instance.

Use the following commands to optimize the database manager configuration:

```
db2 update dbm cfg using MON_HEAP_SZ AUTOMATIC
db2 update dbm cfg using JAVA_HEAP_SZ 2048
db2 update dbm cfg using AGENT_STACK_SZ 256
db2 update dbm cfg using SHEAPTHRES 0
db2 update dbm cfg using INTRA_PARALLEL YES
```

Note: The values specified in the commands are minimum requirements for the MDM Hub.

Set Database Configuration Parameters

Set the configuration parameters for the database.

Use the following commands to set the database configuration parameters:

```
db2 update db cfg using LOCKLIST AUTOMATIC
db2 update db cfg using MAXLOCKS AUTOMATIC
db2 update db cfg using PCKCACHESZ 128000
db2 update db cfg using DBHEAP AUTOMATIC
db2 update db cfg using CATALOGCACHE_SZ 25000
db2 update db cfg using LOGBUFSZ 4096
db2 update db cfg using UTIL_HEAP_SZ 50000
db2 update db cfg using BUFFPAGE 250
db2 update db cfg using STMTHEAP AUTOMATIC
db2 update db cfg using APPLHEAPSZ AUTOMATIC
db2 update db cfg using APPL_MEMORY AUTOMATIC
db2 update db cfg using STAT_HEAP_SZ AUTOMATIC
db2 update db cfg using LOGFILSIZ 128000
db2 update db cfg using LOGPRIMARY 10
db2 update db cfg using LOGSECOND 200
db2 update db cfg using auto_reval deferred_force
```

```
db2 update db cfg using decflt_rounding round_half_up
db2 update db cfg using SHEAPTHRES_SHR AUTOMATIC
db2 update db cfg using DFT_DEGREE 1
```

Note: The values specified in the commands are minimum requirements for the MDM Hub.

Grant Privileges to SYSIBMADM Modules

You must grant privileges to UTL_DIR, UTL_FILE, and DBMS_SQL SYSIBMADM modules.

Use the following commands to grant privileges to modules:

```
GRANT EXECUTE ON MODULE SYSIBMADM.UTL_DIR TO PUBLIC WITH GRANT OPTION
GRANT EXECUTE ON MODULE SYSIBMADM.UTL_FILE TO PUBLIC WITH GRANT OPTION
GRANT EXECUTE ON MODULE SYSIBMADM.DBMS_SQL TO PUBLIC WITH GRANT OPTION
```

Define Buffer Pools for the Database Manager

Define the REPOS_POOL and CMX_POOL buffer pools.

Use the following commands to define buffer pools:

```
CREATE BUFFERPOOL REPOS_POOL IMMEDIATE SIZE 1500 PAGESIZE 32 K
CREATE BUFFERPOOL CMX_POOL IMMEDIATE SIZE 3000 PAGESIZE 32 K
```

Create Tablespaces

You need to create tablespaces that the MDM Hub schemas require.

Create the tablespaces in the following sequence:

1. CMX_DATA
2. CMX_INDX
3. CMX_REPOS
4. CMX_TEMP
5. CMX_USER_TEMP
6. CMX_SYS_TEMP

Use the following statements to create tablespaces for the MDM Hub schemas:

```
CREATE TABLESPACE CMX_DATA PAGESIZE 32 K
    MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_DATA\cmx_data01.dat' 500
M )
    EXTENTSIZE 16
    AUTORESIZE YES
    OVERHEAD 10.5
    PREFETCHSIZE 16
    BUFFERPOOL CMX_POOL

CREATE TABLESPACE CMX_INDX PAGESIZE 32 K
    MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_INDX\cmx_indx01.dat' 500
M )
    EXTENTSIZE 16
    AUTORESIZE YES
    OVERHEAD 10.5
    PREFETCHSIZE 16
    BUFFERPOOL CMX_POOL

CREATE TABLESPACE CMX_REPOS PAGESIZE 32 K
    MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_REPOS\cmx_repos01.dat' 500
M )
    EXTENTSIZE 16
    AUTORESIZE YES
    OVERHEAD 10.5
```

```
    PREFETCHSIZE 16
    BUFFERPOOL REPOS_POOL

CREATE TABLESPACE CMX_TEMP PAGESIZE 32 K
    MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_TEMP\cmx_temp01.dat' 500
M )
    EXTENTSIZE 16
    AUTORESIZE YES
    OVERHEAD 10.5
    PREFETCHSIZE 16
    BUFFERPOOL CMX_POOL

CREATE USER TEMPORARY TABLESPACE CMX_USER_TEMP PAGESIZE 32 K
    MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\USER_TEMP\cmx_user_temp01.dat'
500 M )
    EXTENTSIZE 16
    AUTORESIZE YES
    OVERHEAD 10.5
    PREFETCHSIZE 16
    BUFFERPOOL CMX_POOL

CREATE SYSTEM TEMPORARY TABLESPACE CMX_SYS_TEMP PAGESIZE 32 K
    MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\SYSTEM_TEMP\cmx_sys_temp01.dat'
500 M )
    EXTENTSIZE 16
    AUTORESIZE YES
    OVERHEAD 10.5
    PREFETCHSIZE 16
    BUFFERPOOL CMX_POOL
```

Optionally, to create tablespaces with the dropped table recovery feature enabled, add the following clause to the `CREATE TABLESPACE` statement:

```
DROPPED TABLE RECOVERY ON
```

Use a Script to Create a Database and Tablespaces

The MDM Hub distribution includes a script to create the database and associated tablespaces. To run the script, you need administrative privileges with write and execute permissions to the Db2 data directory.

On UNIX, before you create the database, update the `db2.storage.path` property in the `database.properties` file with the correct database storage path. The `database.properties` file is in the following directory:

```
<MDM Hub distribution directory>/database/bin/db2
```

1. Open a command prompt, and change to the following directory:

   ```
   <MDM Hub distribution directory>/database/bin
   ```

2. To create the database, run the following command:

   On UNIX. `./sip_ant.sh create_db`

   On Windows. `sip_ant.bat create_db`

3. Answer the prompts described in the following table:

Prompt	Description
Enter the database type (ORACLE, MSSQL, DB2)	Database type. Specify `DB2`.
Enter the database instance name [db2]	Name of the database instance. Default is `db2`.
Enter the database name [SIP97]	Name of the database. Default is `SIP97`.
Enter the database storage path [C:\DB2DATA]	Path to the directory where the database must be stored. Default is `C:\DB2DATA`. **Note:** On UNIX, accept the default value. The database storage path that you specify in the `database.properties` file will be used.
Enter the DBA user name [DB2ADMIN]	User name of the administrative user. Default is `DB2ADMIN`.
Enter the DBA password	Password of the administrative user.

The script creates the database and the following tablespaces:

- CMX_DATA
- CMX_INDX
- CMX_TEMP
- CMX_REPOS
- CMX_USER_TEMP
- CMX_SYS_TEMP

To verify that the database was created successfully, review the `sip_ant.log` file in the `<MDM Hub distribution directory>/database/bin` directory.

Step 3. Bind Packages on the Database Server

To ensure that the IBM Db2 client can connect to the database server to run DB2 commands, bind packages on the database server.

1. Open an IBM Db2 command window, and change to the following directory:

   `<IBM Db2 installation directory>/SQLLIB/bnd`

2. Connect to the database by running the following command:

   ```
   db2 connect to <database name> user <database user> using <database user password>
   ```

   Note: The database user must have the bind permission.

3. Run the following bind command:

   ```
   db2 bind @db2cli.lst blocking all grant public sqlerror continue CLIPKG 10
   ```

   The required packages are bound to the database server.

Step 4. Create the ActiveVOS Schema

To install ActiveVOS, you need to create the ActiveVOS schema. To create the schema, run the `create_bpm` script.

If you want to create multiple MDM Hub Master Databases, create an ActiveVOS schema for each MDM Hub Master Database.

1. Open a command prompt and change to the following directory:

   `<MDM Hub distribution directory>/database/bin`

2. Run the following command:

   On UNIX. `./sip_ant.sh create_bpm`

   On Windows. `sip_ant.bat create_bpm`

3. Answer the prompts that appear.

   The prompt displays default text in brackets. Press **Enter** to use the default value and go to the next prompt.

Property	Description
Database Type	Type of database to use. For IBM Db2, specify `DB2`. The database type must be the same as the database type selected for the MDM Hub Master Database and the Operational Reference Stores.
ActiveVOS Database Host Name	Name of the machine that hosts the database.
ActiveVOS Database TCP/IP Port	Port number that the database listener uses.
ActiveVOS Database Name	Name of the database.
ActiveVOS Database Schema/User Name	User name of the ActiveVOS Server administrative user.
ActiveVOS User Password	Password of the administrative user.
DBA User Name	User name of the database administrative user.
DBA Password	Password of the administrative user.
ActiveVOS Tablespace Name	The name of the tablespace that contains the records that are involved in MDM workflows.

4. After you create the schema, review the `sip_ant.log` file in the following directory:

   `<MDM Hub distribution directory>/database/bin`

   The `sip_ant.log` file logs any errors that might occur when you run the `sip_ant` script to create the ActiveVOS schema.

CHAPTER 4

Application Server Tasks

This chapter includes the following topics:

Application Server Tasks Overview

Configure your application server to work with Multidomain MDM. Follow the instructions for your application server.

When you perform a clean upgrade, perform all the steps for your application server as you would for a new installation of Multidomain MDM.

When you perform an in-place upgrade where your application server is still supported, verify that the application server is configured to work with this version of Multidomain MDM.

Set Up JBoss

You can install the MDM Hub in a JBoss cluster environment or on standalone JBoss instances. Install and configure JBoss according to the instructions in the JBoss documentation. Whether you install the MDM Hub in a JBoss cluster environment or on standalone JBoss instances, install the JBoss standalone configuration, and use the full profile of the configuration.

A JBoss cluster consists of one or more cluster nodes on one or more machines. Install and configure JBoss on all machines on which you want cluster nodes. In a cluster environment, ensure that the directory structure of the JBoss installations is the same on all the cluster nodes.

Note: Install the application server in the same time zone as the database server.

Configure Java Virtual Machines

To configure a Java Virtual Machine (JVM), set Java options by using the JAVA_OPTS environment variable.

You can set the Java options in the following file:

On UNIX. `<JBoss installation directory>/bin/standalone.conf`

On Windows. `<JBoss installation directory>\bin\standalone.conf.bat`

The following table describes the Java options settings:

Java Options	Description
-server	Results in a slower startup but subsequent operations are faster.
-De360.connection.channel -De360.mdm.host -De360.mdm.port	Application server communication protocol, host, and port. To deploy the MDM Hub applications on a JBoss port other than 4447, set the following Java options: - `-De360.connection.channel`. Set to the communication protocol that you want to use. Valid values are HTTP and HTTPS. Default is HTTP. - `-De360.mdm.host`. Set to the IP address of the JBoss host. If the environment uses the HTTPS communication protocol and the security certificate is issued to a Fully Qualified Domain Name (FQDN), set to the FQDN. - `-De360.mdm.port`. Set to the JBoss remote port configured in place of 4447. If you do not configure these parameters, Data Director screens that are based on the Entity 360 Framework might not work as expected.
-Didd.mdm.host -Didd.mdm.port -Didd.protocol	Required for Data Director with subject areas. To deploy Data Director with subject areas, set the following Java options: - `-Didd.mdm.host`. Set to the host name or IP address of the application server host. - `-Didd.mdm.port`. Required property, used internally by the Data Director with subject areas application during server initialization. Specifies the HTTP or HTTPS listener port used by the JVM for the applications. Default is `8080`. - `-Didd.protocol`. Required property that is used for deploying the subject area application during server initialization. Specifies whether the communication protocol to use is HTTP or HTTPS. Default is `HTTP`.
-Dio.undertow.legacy.cookie.ALLOW_HTTP_SEPARATORS_IN_V0	Mandatory property if you are using JBoss Version 7.3. Set to `true` to set the property as a system property and prevent REST API authentication failure of a business entity service request that did not contain required credentials due to truncated cookie values. Default is `false`. You do not need to set this property for JBoss Version 7.1 or 7.2.

Java Options	Description
-Ddb2.jcc.charsetDecoderEncoder	Required to use the MDM Hub Sample Operational Reference Store. Enables the JDBC driver to return the Unicode replacement character (U+FFFD) in place of a sequence of bytes that is not a UTF-8 string. Set to `3`.
-Djava.net.preferIPv4Stack	Specifies whether Java uses Internet Protocol version 4 (IPv4). If the operating system uses Internet Protocol version 6 (IPv6), set to `true`.
-Djavax.net.ssl.trustStore -Djavax.net.ssl.trustorePassword	Required if you want to use an HTTPS port for the Process Server. The `-Djavax.net.ssl.trustStore` Java option specifies the path to the truststore file to use for validating client certificates. The `-Djavax.net.ssl.trustStorePassword` Java option specifies the password to access the truststore file.
-Djava.security.egd	Reduces the startup time of Data Director in Linux environments. Set the value to `file:/dev/./urandom`.
-Djboss.as.management.blocking.timeout	Time in seconds to wait for JBoss to deploy. To ensure that JBoss does not fail to start, you can set the value to `5000`. Adjust the time period based on your environment. Default is `300`. If you do not configure the parameter, you might encounter a JBoss deployment timeout.
-Djgroups.bind_addr	Interface on which JGroup must receive and send messages. Required in multinode or cluster environments. Ensure that each node binds to its own network interface.
-DFrameworksLogConfigurationPath	Path to the `log4j.xml` file.
-Dmdm.node.groupid	Specifies a group ID for Java Virtual Machines in the MDM Hub implementation. Required only if you want logical groupings of Hub Servers and Process Servers.
-DUseESLegacyFqSearch	Specifies whether fielded search returns exact matches from within child nodes for a business entity type. Applicable only when you perform a fielded search on multiple fields. Indicates whether a search must return records that contain search values in the same child node if multiple query fields are at the child level. Set to `true` to return records that might match the child level query field from different child nodes. Default is `false`.
-Dfile.encoding -Dorg.apache.catalina.connector.URI_ENCODING	Required if you want to use Informatica Data Director and use REST APIs to search for records. Set both the Java options to `UTF-8` to ensure that you can find and save records that contain UTF-8 characters.
-Dorg.apache.coyote.http11.Http11Protocol.MAX_HEADER_SIZE	Maximum size of the HTTP headers, in bytes. The search requests might fail if the header size is low. Set to `16384`.

Java Options	Description
-Dtask.pageSize=<maximum number of tasks>	Specifies the maximum number of ActiveVOS tasks that are retrieved for each request. Default is `5000`. Increase the number if your environment has a large number of tasks.
-Dstricttransportsecurity.flag	Specifies whether web browsers must convert all attempts to access Data Director using the HTTP requests to the HTTPS requests instead. Set to `true`.
-Xms	Initial heap size. Set to `2048m`.
-Xmx	Maximum JVM heap size. Set to 6 GB or higher. For example, to set the `-Xmx` to `6144m`, use the following JAVA_OPTIONS environment variable setting: `set "JAVA_OPTIONS=-server ... -Xmx6144m"`
XX:+UseCodeCacheFlushing	Specifies whether the JVM disposes of compiled code when the code cache is full.
-XX:ReservedCodeCacheSize	JIT code cache size. To enhance the performance of the MDM Hub environment, set to `512m`.
-XX:MaxMetaspaceSize	Maximum metaspace size. To prevent the JVM from running out of memory, set to `1G`.

Logical Grouping of Java Virtual Machine Example

By grouping Java Virtual Machines (JVMs), you get a logical group of Hub Servers and Process Servers. When you deploy the Hub Server and Process Server applications in a logical JVM group, communication between the Hub Server and Process Server applications stay within the group. To group JVMs, you assign a group ID to each JVM in the MDM Hub environment.

Note: Process Server grouping is applicable to the cleanse and match process only. The logical groups are not applied to the internal server cache of the MDM Hub.

The following table shows an example of logical JVM groups:

JVM Group	JVM	Hub Server	Process Server
Group1	JVM1	Yes	Yes
Group1	JVM4	-	Yes
Group2	JVM2	Yes	Yes
Group3	JVM3	-	Yes

For JVM1, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group1
```

For JVM2, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group2
```

For JVM3, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group3
```

For JVM4, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group1
```

After you configure the JVMs, and deploy the Hub Servers and Process Servers, the groups have the following characteristics:

- Group1 has two Process Servers, Group2 has one Process Server, and Group3 has one Process Server.
- All cleanse and batch calls stay in their own group with the exception of search. For example, any real-time call on the Hub Server in Group1 affects only the Group1 Process Servers (JVM1 and JVM4).

Configure Server Properties for the Full Profile

Configure the server properties for the full profile of the standalone mode in the `standalone-full.xml` file. The file is in the following directory: `<JBoss installation directory>/standalone/configuration`. To configure the server properties, you can run the commands for the configuration from the JBoss Command Line Interface (CLI).

To use the JBoss CLI, perform the following steps:

1. Navigate to the following directory: `<JBoss installation directory>/bin`.
2. To launch the JBoss CLI, run the following script:
   On UNIX. `jboss-cli.sh`

   On Windows. `jboss-cli.bat`
3. To connect to the server, run the following command:

   ```
   connect
   ```

For more information about configuring server properties, see the JBoss documentation.

Transaction Timeout

The transaction timeout property specifies the time in seconds to wait for the MDM Hub transactions to complete. Set the value based on your environment.

Command:

```
/subsystem=transactions:write-attribute(name=default-timeout,value=<timeout in seconds>)
```

Sample configuration:

```
<subsystem xmlns="urn:jboss:domain:transactions:4.0">
    ...
    <coordinator-environment default-timeout="3600"/>
</subsystem>
```

Maximum Post Size

The maximum post size property configures the maximum size in bytes of files that you upload. Set the value to the size limit of files that you want to attach in the Data Director application.

Set the value to 20000000 or higher. Default is 10000000.

Note: After installing the MDM Hub, set the same value for the Hub Server property `cmx.file.max_file_size_mb` in the `cmxserver.properties file.`

Command:

```
/subsystem=undertow/server=default-server/<listener type>=<listener name>/:write-
attribute(name=max-post-size,value=<maximum file size in bytes>)
```

Sample configuration:

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">
    ...
    <server name="default-server">
        <http-listener name="default" socket-binding="http" redirect-socket="https"
enable-http2="true" max-post-size="20000000"/>
        <https-listener name="https" socket-binding="https" security-
realm="ApplicationRealm" enable-http2="true" max-post-size="20000000"/>
        ...
        </host>
    </server>
    ...
</subsystem>
```

Remoting-Connector

The remoting-connector property configures the remoting-connector port and socket binding.

To log in to the Hub Console from a remote machine, set the port to 4447 and the socket binding to remoting.

Note: By default, remoting-connector security is disabled. If you want to configure remoting-connector security for the MDM Hub, ensure that you configure only the Elytron security framework that is supported.

Command:

```
/socket-binding-group=standard-sockets/socket-binding=remoting:add(port=4447)


/subsystem=remoting/connector=remoting-connector:add(socket-binding=remoting)
```

Sample configuration for remoting-connector port:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="$
{jboss.socket.binding.port-offset:0}">
    ...
    <socket-binding name="remoting" port="4447"/>
    ...
</socket-binding-group>
```

Sample configuration for remoting-connector socket binding:

```
<subsystem xmlns="urn:jboss:domain:remoting:4.0">
    ...
    <connector name="remoting-connector" socket-binding="remoting"/>
    ...
</subsystem>
```

Create the ActiveVOS Console Administrative User

If you want to use ActiveVOS, create the ActiveVOS Console administrative user with the `abAdmin` role in the application server container.If you want to use ActiveVOS, create the ActiveVOS Console administrative user with the `abAdmin` role. If you do not create an administrative user, the Hub Server deployment fails. Use the ActiveVOS Console administrative user name and password when the Hub Server installer prompts you to enter the administrative user credentials for the ActiveVOS Console.

1. Change to the following directory:

   ```
   <JBoss installation directory>/bin
   ```

2. To run the add-user utility, use the following script:

   On UNIX. `add-user.sh`

   On Windows. `add-user.bat`

3. Answer the prompts that appear.

The following table describes the values to specify for each prompt:

Prompt	Value to Specify
What type of user do you wish to add? a) Management User or b) Application User	To select Application User, enter `b`.
Realm (ApplicationRealm)	Realm name. Enter the realm name that you specified in the `login-module` that you added to the `standalone-full.xml` file.
Username	ActiveVOS Console administrator name.
Password	Password that complies with the JBoss password standard.
What roles do you want this user to belong to?	`abAdmin`.
About to add user <user name> for realm <realm name>. Is this correct?	To add the user, enter `yes`.
Is this new user going to be used for one AS process to connect to another AS process?	`yes`.

4. Log in to the WebSphere console, and create the ActiveVOS Console administrative user.

   Note: The ActiveVOS console user is mapped to the `abAdmin` role when you run the `postInstallSetup` or the `patchInstallSetup` script during the post-installation or post-upgrade process.
5. Log in to the WebLogic console.
6. Create the `abAdmin` role.
7. Create the ActiveVOS Console administrative user.
8. Assign the administrative user to the `abAdmin` role

Start JBoss

Before you install the Hub Server and the Process Server, start the JBoss application server. Based on your environment, you either start standalone JBoss instances or JBoss cluster nodes.

Starting Standalone JBoss Instances

If you use standalone JBoss instances, start each instance on which you want to install the MDM Hub components.

1. Navigate to the following directory:

   ```
   <JBoss installation directory>/bin
   ```
2. To start a JBoss instance, run the following command:

   On UNIX. `standalone.sh -c standalone-full.xml -b 0.0.0.0 -Djboss.as.management.blocking.timeout=5000`

   On Windows. `standalone.bat -c standalone-full.xml -b 0.0.0.0 -Djboss.as.management.blocking.timeout=5000`

JBoss starts on the available network interfaces and listens for the current host that is defined in the hosts file in the `/etc/hosts` directory. Adjust the timeout interval based on your environment.

3. If you have multiple JBoss instances on the same machine, to start the second and any subsequent JBoss instances, add the following argument to the startup command:

```
-Djboss.socket.binding.port-offset=<port offset range such as, 0,100,200,...n>
```

Starting JBoss Cluster Nodes

If you use a JBoss cluster environment, start the cluster nodes on which you want to install the MDM Hub components.

1. Navigate to the following directory:

```
<JBoss installation directory>/bin
```

2. To start a JBoss cluster node, run the following command on machines that have cluster nodes:

On UNIX.
```
standalone.sh -c standalone-full.xml -b 0.0.0.0 -Djboss.node.name=<Name of the cluster node> -Djboss.server.base.dir=../<node path> -Djboss.as.management.blocking.timeout=5000 -u <multicast address> -Djgroups.bind_addr=<bind address> -Djboss.socket.binding.port-offset=<port offset value> -Djboss.partition.name=<Partition name>
```

On Windows.
```
standalone.bat -c standalone-full.xml -b 0.0.0.0 -Djboss.node.name=<Name of the cluster node> -Djboss.server.base.dir=../<node path> -Djboss.as.management.blocking.timeout=5000 -u <multicast address> -Djgroups.bind_addr=<bind address> -Djboss.socket.binding.port-offset=<port offset value> -Djboss.partition.name=<Partition name>
```

Set the port offset value if multiple cluster nodes run on the same machine. Use the default partition name or ensure that the partition name is the same for all nodes that belong to a cluster. Adjust the timeout interval based on your environment.

Set Up Oracle WebLogic

You can install the MDM Hub in an Oracle WebLogic cluster environment or on standalone WebLogic instances. Install and configure WebLogic according to the instructions in the WebLogic documentation.

Important: There are some known limitations with how Multidomain MDM interacts with some Oracle WebLogic versions, including version 12.2.1.3. Before you begin, see the "Installation and Upgrade" chapter of the *Multidomain MDM Release Notes*.

Before you install the Hub Server and the Process Server on the WebLogic application server, create WebLogic domains. Use the WebLogic Administration console to create domains for the Hub Server and the Process Server applications. For more information, see the WebLogic documentation.

Note: Install the application server in the same time zone as the database server.

Configure the Java Virtual Machines

To configure a Java Virtual Machine (JVM), set the Java options by using the the JAVA_OPTIONS environment variable.

You can set the Java options in the following file:

On UNIX. `<WebLogic domain>/bin/setDomainEnv.sh`

On Windows. `<WebLogic domain>\bin\setDomainEnv.cmd`

The following table describes the Java options:

Java Options	Description
-server	Results in a slower startup but subsequent operations are faster.
-Djgroups.bind_addr	Interface on which JGroup must receive and send messages. Required in a multinode or clustered environment. Ensure that each node binds to its own network interface.
-Djava.net.preferIPv4Stack	Specifies whether Java uses Internet Protocol version 4 (IPv4). If the operating system uses Internet Protocol version 6 (IPv6), set to `true`.
-Djava.security.egd	Reduces the startup time of Informatica Data Director in Linux environments. Set the value to: Djava.security.egd=file:/dev/./urandom
-Doracle.jdbc.J2EE13Compliant	Sets the oracle.jdbc.J2EE13Compliant system variable. Set to `true`. If you do not set the parameter to true, you might encounter Java Database Connectivity (JDBC) issues
-Djavax.wsdl.factory.WSDLFactory	Required for the WebLogic 12.2.1 or later environments to invoke a service WSDL that might have a default namespace prefix. Set to `com.ibm.wsdl.factory.WSDLFactoryImpl`.
-DANTLR_USE_DIRECT_CLASS_LOADING	Required for launching WebLogic 12.2.1 or later environments on which the MDM Hub is deployed. Set to `true`.
-Dmdm.node.groupid	Specifies a group ID for Java Virtual Machines in the MDM Hub implementation. Required only if you want logical groupings of Hub Servers and Process Servers.
-De360.connection.channel	Communication protocol that the application server must use for communication between the Hub Server and the Data Director screens based on the Entity 360 Framework. Valid values are HTTP and HTTPS. Default is HTTP. Ensure that you set the following supporting Java options: - -De360.mdm.host - -De360.mdm.port
-DUseESLegacyFqSearch	Specifies whether fielded search returns exact matches from within child nodes for a business entity type. Applicable only when you perform a fielded search on multiple fields. Indicates whether a search must return records that contain search values in the same child node if multiple query fields are at the child level. Set to `true` to return records that might match the child level query field from different child nodes. Default is `false`.

Java Options	Description
-Dweblogic.security.SSL.trustedCAKeyStore -Dweblogic.security.SSL.enable.renegotiation -Dweblogic.security.SSL.verbose	Required if HTTPS is enabled for WebLogic. The `-Dweblogic.security.SSL.trustedCAKeyStore` Java option specifies the path to the keystore that contains the trusted certificates. Set to the absolute keystore path. The `-Dweblogic.security.SSL.enable.renegotiation` Java option enables SSL renegotiation. Set to `true`. The `-Dweblogic.security.SSL.verbose` Java option enables additional SSL debugging. Set to `true`.
-De360.mdm.host	Application server host. If the environment uses the HTTPS communication protocol and the security certificate is issued to a Fully Qualified Domain Name (FQDN), set to the FQDN. Set the Java option in one of the following scenarios: - To deploy the MDM Hub applications only on Managed Servers, add the Java option to the script of each Managed Server. Set to the IP address or symbolic name of the Managed Server. - To deploy the MDM Hub applications on a WebLogic port other than 7001, add the Java option to the application server startup script. Set to the IP address or symbolic name of the WebLogic host. If you do not configure this parameter, Data Director screens based on the Entity 360 Framework might not work as expected.
-De360.mdm.port	Application server port. Set the Java option in one of the following scenarios: - To deploy the MDM Hub applications only on Managed Servers, add the Java option to the script of each Managed Server. Set to the port number of the Managed Server. - To deploy the MDM Hub applications on a WebLogic port other than 7001, add the Java option to the application server startup script and set to the port number. If you do not configure this parameter, Data Director screens based on the Entity 360 Framework might not work as expected.
-Didd.mdm.host -Didd.mdm.port -Didd.protocol	Required for Data Director with subject areas. To deploy Data Director with subject areas, set the following Java options: - `-Didd.mdm.host`. Set to the host name or IP address of the application server host. - `-Didd.mdm.port`. Required property, used internally by the Data Director with subject areas application during server initialization. Specifies the HTTP or HTTPS listener port used by the JVM for the applications. Default is `8080`. - `-Didd.protocol`. Required property that is used for deploying the subject area application during server initialization. Specifies whether the communication protocol to use is HTTP or HTTPS. Default is `HTTP`.
-Dfile.encoding -Dweblogic.http.URIDecodeEncoding	Required if you want to use Data Director and use REST APIs to search for records. Set both the Java options to `UTF-8` to ensure that you can find and save records that contain UTF-8 characters.
-DFrameworksLogConfigurationPath	Sets the `log4j.xml` file configuration path.

Java Options	Description
-Dtask.pageSize=<maximum number of tasks>	Specifies the maximum number of ActiveVOS tasks that are retrieved for each request. Default is `5000`. Increase the number if your environment has a large number of tasks.
-Dstricttransportsecurity.flag	Specifies whether web browsers must convert all attempts to access Data Director using the HTTP requests to the HTTPS requests instead. Set to `true`.
-Xms	Initial heap size. Set to `2048m`.

-Xmx	Maximum JVM heap size. Set to 6 GB or higher.
	For example, use the following USER_MEM_ARGS environment variable setting:
	`USER_MEM_ARGS="-Xms2048m -Xmx6000m"`
	`export USER_MEM_ARGS`

XX:+UseCodeCacheFlushing	Specifies whether the JVM disposes of compiled code when the code cache is full.
-XX:ReservedCodeCacheSize	JIT code cache size. To enhance the performance of the MDM Hub environment, set to `512m`.

Logical Grouping of Java Virtual Machine Example

By grouping Java Virtual Machines (JVMs), you get a logical group of Hub Servers and Process Servers. When you deploy the Hub Server and Process Server applications in a logical JVM group, communication between the Hub Server and Process Server applications stay within the group. To group JVMs, you assign a group ID to each JVM in the MDM Hub environment.

Note: Process Server grouping is applicable to the cleanse and match process only. The logical groups are not applied to the internal server cache of the MDM Hub.

The following table shows an example of logical JVM groups:

JVM Group	JVM	Hub Server	Process Server
Group1	JVM1	Yes	Yes
Group1	JVM4	-	Yes
Group2	JVM2	Yes	Yes
Group3	JVM3	-	Yes

For JVM1, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group1
```

For JVM2, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group2
```

For JVM3, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group3
```

For JVM4, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group1
```

After you configure the JVMs, and deploy the Hub Servers and Process Servers, the groups have the following characteristics:

- Group1 has two Process Servers, Group2 has one Process Server, and Group3 has one Process Server.
- All cleanse and batch calls stay in their own group with the exception of search. For example, any real-time call on the Hub Server in Group1 affects only the Group1 Process Servers (JVM1 and JVM4).

Disable WebLogic Server Authentication

The MDM Hub uses HTTP basic authentication for which you must disable WebLogic Server authentication. To disable WebLogic Server authentication, edit the `config.xml` file.

1. Navigate to the following WebLogic directory:

   ```
   <WebLogic installation directory>/user_projects/domains/<user domain>/config
   ```

2. Open the `config.xml` file in a text editor.
3. Add the following element within the <security-configuration> element:

   ```
   <enforce-valid-basic-auth-credentials>
      false
   </enforce-valid-basic-auth-credentials>
   ```

Create the ActiveVOS Console Administrative User

If you want to use ActiveVOS, create the ActiveVOS Console administrative user with the `abAdmin` role in the application server container.If you want to use ActiveVOS, create the ActiveVOS Console administrative user with the `abAdmin` role. If you do not create an administrative user, the Hub Server deployment fails. Use the ActiveVOS Console administrative user name and password when the Hub Server installer prompts you to enter the administrative user credentials for the ActiveVOS Console.

1. Change to the following directory:

   ```
   <JBoss installation directory>/bin
   ```

2. To run the add-user utility, use the following script:

   On UNIX. `add-user.sh`

   On Windows. `add-user.bat`

3. Answer the prompts that appear.

   The following table describes the values to specify for each prompt:

Prompt	Value to Specify
What type of user do you wish to add? a) Management User or b) Application User	To select Application User, enter `b`.
Realm (ApplicationRealm)	Realm name. Enter the realm name that you specified in the `login-module` that you added to the `standalone-full.xml` file.
Username	ActiveVOS Console administrator name.

Prompt	Value to Specify
Password	Password that complies with the JBoss password standard.
What roles do you want this user to belong to?	`abAdmin`.
About to add user <user name> for realm <realm name>. Is this correct?	To add the user, enter `yes`.
Is this new user going to be used for one AS process to connect to another AS process?	`yes`.

4. Log in to the WebSphere console, and create the ActiveVOS Console administrative user.

   Note: The ActiveVOS console user is mapped to the `abAdmin` role when you run the `postInstallSetup` or the `patchInstallSetup` script during the post-installation or post-upgrade process.
5. Log in to the WebLogic console.
6. Create the `abAdmin` role.
7. Create the ActiveVOS Console administrative user.
8. Assign the administrative user to the `abAdmin` role

Additional Oracle WebLogic Configuration

Perform additional WebLogic configuration based on the requirements of the MDM Hub environment.

The following table describes the configurations that you can perform:

Configuration	Description
Configuring WebLogic for standalone Process Server instances	Required to configure WebLogic for standalone Process Server instances in the following scenarios: - You want to install a Process Server instance on a WebLogic instance on which you do not have the Hub Server installed. - You want to install multiple, standalone Process Server instances.
Configuring WebLogic for multiple MDM Hub Master Databases	Required if you want to configure multiple MDM Hub Master Database instances.
Configuring the HTTPS protocol	Required if you want to configure the HTTPS protocol for the MDM Hub communications.

Configuring WebLogic for Standalone Process Server Instances

If you want to install multiple, standalone Process Server instances, configure WebLogic to use the appropriate data source. Also, if you want to install a Process Server instance on a WebLogic instance on which you do not have the Hub Server installed, configure the data source.

Perform the following tasks to configure WebLogic to use the appropriate data source:

1. Install the JDBC driver.
2. Create an MDM Hub Master Database data source.
3. Create an Operational Reference Store data source.

Step 1. Install the JDBC Driver

Before you create data sources for the MDM Hub Master Database and the Operational Reference Store (ORS), install the JDBC driver.

Contact Oracle to get the supported version of the JDBC driver.

Contact Microsoft to get the supported version of the JDBC driver.

Contact IBM to get the supported version of the JDBC driver.

1. Copy the JDBC driver to the following directory:

   `<WebLogic installation directory>/wlsserver/server/lib`

2. Add the path to the JDBC driver to the `CLASSPATH` variable in the following file:

   On UNIX. `<WebLogic domain>/bin/commEnv.sh`

   On Windows. `<WebLogic domain>\bin\commEnv.cmd`

   Note: Place the path to the JDBC driver before the path to other Weblogic Server libraries.

Step 2. Create an MDM Hub Master Database Data Source

After you install the JDBC driver, on the Process Server machine, create a data source for the MDM Hub Master Database.

1. On the WebLogic Administration Console, click the **Lock & Edit** button to acquire a lock.
2. Click **Services** > **JDBC** > **Data Sources**, and then click **New**.

   The **JDBC Data Sources Properties** page appears.

3. Specify the following data source properties:

Property	Description
Name	Name of the JDBC data source. Set the name to `MDM Master Data Source`.
JNDI Name	JNDI path to where the JDBC data source will be bound. Specify `jdbc/siperian-cmx_system-ds`.
Database Type	Database type that you want to connect to. Select **Oracle**.Select **MS SQL Server**.
Database Driver	JDBC driver that you want to use to connect to the database. Select **Oracle driver (Thin XA)**.Select **MS SQL Server Driver (Type 4 XA) Versions: 2005 or later**.

4. Click **Next**, and again click **Next**.

The **Connection Properties** page appears.

5. Enter values for the following connection properties:

Property	Description
Database Name	Name of the database you want to connect to.
Host Name	DNS name or IP address of the server that hosts the database. To deploy the MDM Hub application on a WebLogic port other than 7001, set the host name and the IP address of the WebLogic host. Do not use localhost.
Port	Port on which the database server listens for connection requests.
Database User Name	Database user name that you want to use for each connection in the data source.
Password	Password of the database user account.
Confirm Password	Password of the database user account.

6. Click **Next**.

   The **Test Database Connection** page appears.

7. Click **Test Configuration** to test the driver connections.

   If the test is unsuccessful, you must update the values in the **Connection Properties** page and then retry the connection until successful.

8. Click **Next**, and then select the server on which you want to deploy the data source.
9. Click **Finish**, and then click **Activate Changes**.

Step 3. Create an Operational Reference Store Data Source

On the Process Server machine, create a data source for each Operational Reference Store.

1. On the WebLogic Administration Console, click the **Lock & Edit** button to acquire a lock.
2. Click **Services** > **JDBC** > **Data Sources**, and then click **New**.

   The **JDBC Data Sources Properties** page appears.

3. Specify the following data source properties:

Property	Description
Name	Name of the JDBC data source. Set the name to `MDM ORS Data Source`.
JNDI Name	JNDI path to where the JDBC data source will be bound. Specify `jdbc/siperian-<oracle host name>-<oracle sid>-<Operational reference Store name>-ds`. Specify `jdbc/siperian-<Microsoft SQL Server host name>-<Operational reference Store name>-ds`.

Property	Description
Database Type	Database type that you want to connect to. Select **Oracle**. Select **MS SQL Server**.
Database Driver	JDBC driver that you want to use to connect to the database. Select **Oracle driver (Thin XA)**. Select **MS SQL Server Driver (Type 4 XA) Versions: 2005 or later**.

4. Click **Next**, and again click **Next**.

   The **Connection Properties** page appears.

5. Enter values for the following connection properties:

Property	Description
Database Name	Name of the database you want to connect to.
Host Name	DNS name or IP address of the server that hosts the database. To deploy the MDM Hub application on a WebLogic port other than 7001, set the host name and the IP address of the WebLogic host. Do not use localhost.
Port	Port on which the database server listens for connection requests.
Database User Name	Database user name that you want to use for each connection in the data source.
Password	Password of the database user account.
Confirm Password	Password of the database user account.

6. Click **Next**.

   The **Test Database Connection** page appears.

7. Click **Test Configuration** to test the driver connections.

   If the test is unsuccessful, you must update the values in the **Connection Properties** page and then retry the connection until successful.

8. Click **Next**, and then select the server on which you want to deploy the data source.
9. Click **Finish**, and then click **Activate Changes**.

Configuring WebLogic for Multiple MDM Hub Master Databases

If you want to configure multiple MDM Hub Master Database instances, configure as many WebLogic domains as the number of MDM Hub Master Database instances. Each MDM Hub Master Database instance must have its own MDM Hub instance. Therefore, create as many WebLogic domains to deploy each MDM Hub instance on a separate WebLogic domain.

Configuring the HTTPS Protocol

You can configure the HTTPS protocol for the MDM Hub communications. Use the WebLogic Server Administration Console to configure the HTTPS protocol. Alternatively, you can use the default JDK secure certificates to enable HTTPS.

Note: Before you configure the keystores, either use Java Keytool or OpenSSL to generate self-signed certificates for WebLogic Server.

1. On the WebLogic Server Administration Console, under the **Environment** section, click **Servers**.
2. On the **Summary of Servers** page, click the **AdminServer(admin)** link from the Servers list.
3. Enable the **SSL Listen Port Enabled** option, and enter the SSL listen port number.
4. Click the **SSL** tab, and select **None** from the **Hostname Verification**.
5. Select the name of the server that you want to configure the identity and trust keystores for.
6. Click **Configuration** > **Keystores**.
7. Change the password for the following configuration options:

Configuration Option	Description
Demo Identity Keystore Passphrase	Encrypted password of the demo identity keystore.
Demo Trust Keystore Passphrase	Encrypted password of the demo trust keystore.
Java Standard Trust Keystore Passphrase	Password of the Java Standard Trust keystore.

8. Save the changes.

Set Up IBM WebSphere

You can install the MDM Hub in IBM WebSphere cluster environments or standalone WebSphere instances. Install and configure WebSphere according to the instructions in the WebSphere documentation.

Ensure that there are no white spaces in the WebSphere installation directory path.

Note: Install the application server in the same time zone as the database server.

The following table lists the properties and their values to configure before installation, followed by a brief description of where to set the property:

Custom Property	Value	Description
com.ibm.ws.scripting.echoparams	false	Set this property in the `wsadmin.properties` file, which is located the following directory: `<WebSphere installation directory>\WebSphere\AppServer\profiles\<profile name>\properties` Set this value to false to prevent the ActiveVOS database password from appearing in script text in the `patchinstallSetup.log` file. Default is true.

Configure Java Virtual Machines

To configure a Java Virtual Machine (JVM), set Java options by using the JAVA_OPTIONS environment variable. After you edit or add any Java options, restart the JVM.

If you use a WebSphere clustered environment, set the Java options for the following cluster components:

- Server. Set all the required Java options on each server in the cluster.
- Deployment Manager. Set all the required Java options.
- Node agent. Set only heap size by using the `-Xmx` and `-Xms` Java options.

The following table describes the Java options settings:

Java Options	Description
-server	Results in a slower startup but subsequent operations are faster.
-Djava.net.preferIPv4Stack	Specifies whether Java uses Internet Protocol version 4 (IPv4). If the operating system uses Internet Protocol version 6 (IPv6), set to `true`.
- Djava.security.egd	Reduces the startup time of Data Director in Linux environments. Set the value to `file:/dev/./urandom`.
-DUseESLegacyFqSearch	Specifies whether fielded search returns exact matches from within child nodes for a business entity type. Applicable only when you perform a fielded search on multiple fields. Indicates whether a search must return records that contain search values in the same child node if multiple query fields are at the child level. Set to `true` to return records that might match the child level query field from different child nodes. Default is `false`.
-Ddb2.jcc.charsetDecoderEncoder	Required to use the MDM Hub Sample Operational Reference Store. Enables the JDBC driver to return the Unicode replacement character (U+FFFD) in place of a sequence of bytes that is not a UTF-8 string. Set to `3`.
-Dcom.ibm.crypto.provider.DoRSATypeChecking	Specifies whether Java allows the RSA type encryption of data with private key and decryption with public key. Required for the MDM Hub installer to read the license certificates and for password hashing to work in the MDM Hub. Set to `false`. If you do not set `-Dcom.ibm.crypto.provider.DoRSATypeChecking` to `false`, the Hub Server might not start and you can encounter license errors.
-Djgroups.bind_addr	Interface on which JGroup must receive and send messages. Required in a multinode or clustered environment. Ensure that each node binds to its own network interface.

Java Options	Description
-De360.mdm.host -De360.mdm.port -De360.connection.channel	Application server communication protocol, host, and port. To deploy the MDM Hub applications on a Bootstrap port other than 2809, set the following Java options: - `-De360.connection.channel`. Set to the communication protocol that you want to use. Valid values are HTTP and HTTPS. Default is HTTP. - `-De360.mdm.host`. Set to the IP address of the WebSphere host. If the environment uses the HTTPS communication protocol and the security certificate is issued to a Fully Qualified Domain Name (FQDN), set to the FQDN. - `-De360.mdm.port`. Set to the WebSphere Bootstrap port configured in place of 2809. If you do not configure this parameter, Data Director screens that are based on the Entity 360 Framework might not work as expected.
-Didd.mdm.host -Didd.mdm.port -Didd.protocol	Required for Data Director with subject areas. To deploy Data Director with subject areas, set the following Java options: - `-Didd.mdm.host`. Set to the host name or IP address of the application server host. - `-Didd.mdm.port`. Required property, used internally by the Data Director with subject areas application during server initialization. Specifies the HTTP or HTTPS listener port used by the JVM for the applications. Default is `8080`. - `-Didd.protocol`. Required property that is used for deploying the subject area application during server initialization. Specifies whether the communication protocol to use is HTTP or HTTPS. Default is `HTTP`.
-DFrameworksLogConfigurationPath	Path to the `log4j.xml` file.
-Doracle.jdbc.J2EE13Compliant	Specifies whether the system variable for the Oracle driver is fully compliant with J2EE. Set to `true`. If you do not set the parameter to true, you might encounter Java Database Connectivity (JDBC) issues
-Dmdm.node.groupid	Specifies a group ID for Java Virtual Machines in the MDM Hub implementation. Required only if you want logical groupings of Hub Servers and Process Servers.
-Dfile.encoding -Dclient.encoding.override	Required if you want to use Informatica Data Director and use REST APIs to search for records. Set both the Java options to `UTF-8` to ensure that you can find and save records that contain UTF-8 characters.
-Dstricttransportsecurity.flag	Specifies whether web browsers must convert all attempts to access Data Director using the HTTP requests to the HTTPS requests instead. Set to `true`.
-XX:codecachetotal	JIT code cache size. To enhance the performance of the MDM Hub environment, set to `512m`.

Java Options	Description
-Xmx	Maximum JVM heap size. Set to 6 GB or higher. For example, to set the `-Xmx` to `6144m`, use the following JAVA_OPTIONS environment variable setting: `set "JAVA_OPTIONS=-server ... -Xmx6144m"`
-Xms	Initial heap size. Set to `2048m`.
-Xmso	Required for the Process Server JVMs. Initial stack size for operating system threads. Prevents the application server from shutting down unexpectedly due to low system thread stack size. Set to `4096k`.
-Xss	Initial stack size. Set to `2000k`.
XX:+UseCodeCacheFlushing	Specifies whether the JVM disposes of compiled code when the code cache is full.
-Dtask.pageSize=<maximum number of tasks>	Specifies the maximum number of ActiveVOS tasks that are retrieved for each request. Default is `5000`. Increase the number if your environment has a large number of tasks.

Logical Grouping of Java Virtual Machine Example

By grouping Java Virtual Machines (JVMs), you get a logical group of Hub Servers and Process Servers. When you deploy the Hub Server and Process Server applications in a logical JVM group, communication between the Hub Server and Process Server applications stay within the group. To group JVMs, you assign a group ID to each JVM in the MDM Hub environment.

Note: Process Server grouping is applicable to the cleanse and match process only. The logical groups are not applied to the internal server cache of the MDM Hub.

The following table shows an example of logical JVM groups:

JVM Group	JVM	Hub Server	Process Server
Group1	JVM1	Yes	Yes
Group1	JVM4	-	Yes
Group2	JVM2	Yes	Yes
Group3	JVM3	-	Yes

For JVM1, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group1
```

For JVM2, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group2
```

For JVM3, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group3
```

For JVM4, add the following Java option in the startup script:

```
-Dmdm.node.groupid=Group1
```

After you configure the JVMs, and deploy the Hub Servers and Process Servers, the groups have the following characteristics:

- Group1 has two Process Servers, Group2 has one Process Server, and Group3 has one Process Server.
- All cleanse and batch calls stay in their own group with the exception of search. For example, any real-time call on the Hub Server in Group1 affects only the Group1 Process Servers (JVM1 and JVM4).

Encrypt Passwords in the MDM Hub Environment

To encrypt sensitive data such as passwords that appear in log files in the MDM Hub environment, configure scripting administration in WebSphere.

1. Open the `wsadmin.properties` file in the following directory:

   ```
   <WebSphere installation directory>/profiles/<Application server profile name>/properties
   ```

2. Set the `com.ibm.ws.scripting.echoparams` Java property to `false`.

Create a Secure Profile in a WebSphere Environment

In WebSphere, configure a secure profile to use with Multidomain MDM and Informatica ActiveVOS.

1. From a command line, create a secure profile as shown in the following sample code:

   On Windows

   ```
   <app_server_root>\bin\manageprofiles.bat -create -profileName AppSrv01
   -profilePath <app_server_root>\profiles\AppSrv01
   -templatePath <app_server_root>\profileTemplates\default
   -adminUserName administrator -adminPassword password1 -enableAdminSecurity true
   ```

   On UNIX

   ```
   <app_server_root>/bin/manageprofiles.sh -create -profileName AppSrv01
   -profilePath <app_server_root>/profiles/AppSrv01
   -templatePath <app_server_root>/profileTemplates/default
   -adminUserName administrator -adminPassword password1 -enableAdminSecurity true
   ```

2. In the WebSphere console, change the security Transport type to SSL-Supported.
   a. Expand **Security** and click **Global Security**.
   b. Under Authentication, expand **RMI/IIOP security** and click **CSIv2 inbound communications**
   c. Under CSIv2 Transport Layer, from the Transport list, select **SSL-Supported**.
   d. Click **Apply**, and then click **Save**.
   e. Click **CSIv2 outbound communications**
   f. Under CSIv2 Transport Layer, from the Transport list, select **SSL-Supported**.
   g. Click **Apply**, and then click **Save**.
3. In the WebSphere console, ensure that application security is set.
   a. Expand **Security** and click **Global Security**.
   b. Under Application Security, select **Enable application security**.
   c. Click **Apply**, and then click **Save**.
4. Set up federated repositories.
   a. Expand **Security** and click **Global Security**.

b. Under User account repository, from the Available realm definitions list, select **Federated repositories**.

c. Click **Configure**.

d. Under Repositories in the realm, click **Use built-in repository**.

e. Specify a password for the administrative user.

f. Click **Apply**, and then click **Save**.

5. Restart the WebSphere profile.

Create the ActiveVOS Console Administrative User

If you want to use ActiveVOS, create the ActiveVOS Console administrative user with the `abAdmin` role in the application server container.If you want to use ActiveVOS, create the ActiveVOS Console administrative user with the `abAdmin` role. If you do not create an administrative user, the Hub Server deployment fails. Use the ActiveVOS Console administrative user name and password when the Hub Server installer prompts you to enter the administrative user credentials for the ActiveVOS Console.

1. Change to the following directory:

   `<JBoss installation directory>/bin`

2. To run the add-user utility, use the following script:

   On UNIX. `add-user.sh`

   On Windows. `add-user.bat`

3. Answer the prompts that appear.

   The following table describes the values to specify for each prompt:

Prompt	Value to Specify
What type of user do you wish to add? a) Management User or b) Application User	To select Application User, enter `b`.
Realm (ApplicationRealm)	Realm name. Enter the realm name that you specified in the `login-module` that you added to the `standalone-full.xml` file.
Username	ActiveVOS Console administrator name.
Password	Password that complies with the JBoss password standard.
What roles do you want this user to belong to?	`abAdmin`.
About to add user <user name> for realm <realm name>. Is this correct?	To add the user, enter `yes`.
Is this new user going to be used for one AS process to connect to another AS process?	`yes`.

4. Log in to the WebSphere console, and create the ActiveVOS Console administrative user.

   Note: The ActiveVOS console user is mapped to the `abAdmin` role when you run the `postInstallSetup` or the `patchInstallSetup` script during the post-installation or post-upgrade process.

5. Log in to the WebLogic console.
6. Create the `abAdmin` role.
7. Create the ActiveVOS Console administrative user.
8. Assign the administrative user to the `abAdmin` role

Configure SOAP Request Timeout for MDM Hub Deployments

To ensure that deployment of the MDM Hub components do not time out, set the SOAP request timeout property. After a successful installation, you can reset the property to its default value.

1. Open the `soap.client.props` file in the following directory:

   `<WebSphere installation directory>/profiles/<Application server profile name>/properties`
2. Set the `com.ibm.SOAP.requestTimeout` property to 1800 or higher.

Additional IBM WebSphere Configuration

Perform additional WebSphere configuration based on the requirements of the MDM Hub environment.

The following table describes the configurations that you can perform:

Configuration	Description
Configuring WebSphere for standalone Process Server instances	Required to configure WebSphere for standalone Process Server instances in the following scenarios: - You want to install a Process Server instance on a WebSphere instance on which you do not have the Hub Server installed. - You want to install multiple, standalone Process Server instances.
Configuring WebSphere for multiple MDM Hub Master Databases	Required if you want to configure multiple MDM Hub Master Database instances.
Configuring the HTTPS protocol	Required if you want to configure the HTTPS protocol for the MDM Hub communications.
Configuring WebSphere for Informatica Data Director	Required if you want to use Data Director.

Configuring WebSphere for Standalone Process Server Instances

If you want to install multiple, standalone Process Server instances, configure WebSphere to use the appropriate data source. Also, if you want to install a Process Server instance on a WebSphere instance on which you do not have the Hub Server installed, configure the data source.

Perform the following tasks to configure WebSphere to use the appropriate data source:

1. Install the JDBC driver.
2. Create an MDM Hub Master Database data source.
3. Create an Operational Reference Store data source.

Step 1. Install the JDBC Driver

Before you create data sources for the MDM Hub Master Database and the Operational Reference Store (ORS), install the JDBC driver.

Contact Oracle to get the supported version of the JDBC driver.

Contact Microsoft to get the supported version of the JDBC driver.

Contact IBM to get the supported version of the JDBC driver.

- Copy the JDBC driver to the following directory:

  ```
  <WebSphere installation directory>/lib
  ```

Step 2. Create an MDM Hub Master Database Data Source

After you install the JDBC driver, on the Process Server machine, create a data source for the MDM Hub Master Database.

1. Start the WebSphere Application Server Administrative Console.
2. Specify the location of the driver libraries.
   a. Expand **Environment** in the console navigation tree.
   b. Click the **WebSphere Variables** link.
   c. Update the JDBC variable to point to the following JDBC driver directory:

      ```
      <WebSphere installation directory>/lib
      ```
3. Create the security account that the MDM Hub Master Database data source will use.
   a. Expand **Security** in the console navigation tree.
   b. Click the **Secure administration, applications, and infrastructure** link.
   c. Under **Authentication**, expand **Java Authentication and Authorization Service**, and click **J2C Authentication Data**.
   d. Click **New**, and specify the following properties:

Property	Description
Alias	Name of the MDM Hub Master Database.
User ID	User name to connect to the MDM Hub Master Database.
Password	Password to access the MDM Hub Master Database.

   e. Click **OK**.
4. Create the JDBC Provider.
   a. Expand **Resources** > **JDBC**, and then click **JDBC Providers**.

      The **JDBC Provider** page appears.
   b. Select the scope for applications to use the data source.

c. Click **New**, and specify the following properties:

Property	Description
Database type	Type of database. Select **Oracle.**
Provider type	Type of JDBC provider. Select **Oracle JDBC Driver**.
Implementation type	Data source implementation type. Select **XA data source**.
Name	Name of the JDBC provider. Change the name to `Informatica MDM Oracle JDBC Provider (XA)`.

d. Click **Next**, and then click **Finish**.

5. Create the MDM Hub Master Database data source.
   a. Click the JDBC provider that you created.

      The **Configuration** page appears.
   b. Under **Additional Properties**, click **Data sources**.

      The **Data Sources** page appears.
   c. Click **New**.
   d. Specify the following data source properties:

Property	Description
Name	Data source name. Specify `MDM Master Data Source`.
JNDI Name	JNDI path to where the JDBC data source will be bound. Specify `jdbc/siperian-cmx_system-ds`. **Note:** The JNDI name must be in lower case.
Component-managed Authentication Alias	Authentication alias of the Master Database data source. Select `<host name>/cmx_system`.

   e. Click **Next**, and then click **Finish**.

Step 3. Create an Operational Reference Store Data Source

After you install the JDBC driver, on the Process Server machine, create a data source for each Operational Reference Store.

1. Start the WebSphere Application Server Administrative Console.
2. Specify the location of the driver libraries.
   a. Expand **Environment** in the console navigation tree.
   b. Click the **WebSphere Variables** link.
   c. Update the JDBC variable to point to the following JDBC driver directory:

      `<WebSphere installation directory>/lib`

3. Create the security account that the Operational Reference Store will use.
   a. Expand **Security** in the console navigation tree.
   b. Click the **Secure administration, applications, and infrastructure** link.
   c. Under **Authentication**, expand **Java Authentication and Authorization Service**, and click **J2C Authentication Data**.
   d. Click **New**, and set the following properties:

Property	Description
Alias	Name of the Operational Reference Store.
User ID	User name to connect to the Operational Reference Store.
Password	Password to access the Operational Reference Store.

   e. Click **OK**.

4. Create the JDBC Provider.
   a. Expand **Resources** > **JDBC**, and then click **JDBC Providers**.

      The **JDBC Provider** page appears.
   b. Select the scope for applications to use the data source.
   c. Click **New**, and specify the following properties:

Property	Description
Database type	Type of database. Select **Oracle.**
Provider type	Type of JDBC provider. Select **Oracle JDBC Driver**.
Implementation type	Data source implementation type. Select **XA data source**.
Name	Name of the JDBC provider. Change the name to `Informatica MDM Oracle JDBC Provider (XA)`.

   d. Click **Next**, and then click **Finish**.

5. Create the Operational Reference Store data source.
   a. Click the JDBC provider that you created.

      The **Configuration** page appears.
   b. Under **Additional Properties**, click **Data sources**.

      The **Data Sources** page appears.
   c. Click **New**.

d. Specify the following data source properties:

Property	Description
Name	Data source name. Specify `MDM ORS Data Source`.
JNDI Name	JNDI path to where the JDBC data source will be bound. Specify `jdbc/siperian-<Oracle host name>-<Oracle SID>-<Operational Reference Store name>-dsjdbc/siperian-<IBM Db2 host name>-<IBM Db2 database name>-<Operational Reference Store name>-dsjdbc/siperian-<Microsoft SQL Server host name>-<Operational Reference Store name>-ds`. **Note:** The JNDI name must be in lower case.
Component-managed Authentication Alias	Authentication alias of the Master Database data source. Select `<host name>/<Operational Reference Store name>`.
Driver Class Name	JDBC driver class. Set to `com.microsoft.sqlserver.jdbc.SQLServerXADataSource`.

e. Click **Next**, and then click **Finish**.

Configuring WebSphere for Multiple MDM Hub Master Databases

If you want to configure multiple MDM Hub Master Database instances, configure as many WebSphere profiles as the number of MDM Hub Master Database instances. Each MDM Hub Master Database instance must have its own MDM Hub instance. Therefore, create as many WebSphere profiles to deploy each MDM Hub instance on a separate WebSphere profile.

Configuring the HTTPS Protocol

To use the HTTPS protocol for communication between the MDM Hub components, such as the Hub Server, Process Server, and ActiveVOS, configure the HTTPS protocol in the WebSphere application server.

1. Create an SSL-enabled WebSphere port.
2. Configure WebSphere to allow self-signed certificates.
3. Configure the following custom JVM properties:

Custom JVM Property	Description
javax.net.ssl.keyStore	Location of the keystore.
javax.net.ssl.keyStorePassword	Password of the keystore.
javax.net.ssl.keyStoreType	Type of the keystore.
javax.net.ssl.trustStore	Location of the truststore.
javax.net.ssl.trustStorePassword	Password of the truststore.
javax.net.ssl.trustStoreType	Type of the truststore.

For more information about configuring the HTTPS protocol, see the WebSphere documentation.

Configuring WebSphere for Informatica Data Director

If you want to use Data Director, configure WebSphere and then restart WebSphere for the changes to take effect.

Ensure that you perform the following configurations:

- Set the web container custom property.
  Use the WebSphere Console to set `com.ibm.ws.webcontainer.invokerequestlistenerforfilter` to `true`. For instructions on setting web container custom properties, see the WebSphere documentation.
- To support the management of tasks, increase the value for timeout properties by a factor of 2.
  Perform the following task by using the the WebSphere Console:
  1. navigate to **WebSphere Console Servers** > **Server Types** > **WebSphere application servers** > **<target server name>**.
  2. In the **Container Services** category, click **Transaction service** and increase the values for the timeout properties.

CHAPTER 5

Hub Store Upgrade

This chapter includes the following topics:

Hub Store Upgrade Overview

The Hub Store is the database that contains the MDM Hub Master Database and one or more Operational Reference Store (ORS) databases. Use the scripts provided in the distribution to upgrade the databases.

If your database environment is set to a non-English locale, ensure that the character set is a Unicode character set before you upgrade the Hub Store. After the upgrade completes successfully, you can select your preferred locale. The locale is stored as a user account preference, rather than at the database level.

Note: Ensure that you do not include spaces in the path to the database directory or folder names. If you specify a path that has spaces in the directory or folder names, the upgrade fails.

Clone the Hub Store (Clean Upgrade)

If you are performing a clean upgrade, ask a DBA to back up and clone the Master Database and the Operational Reference Stores. Copy the cloned databases to the new environment. You upgrade the Master Database and Operational Reference Stores in the new environment.

Databases Set to a Non-English Locale

If the Hub Store database environment is set to a non-English locale, ensure that the database environment uses a Unicode character set before you run the upgrade script. You set the character set using a database environment variable.

Note: This task is not required for Microsoft SQL Server environments.

The upgrade script translates metadata to English and associates a translation key with the metadata. After the upgrade is successful, each MDM Hub Console user can select any supported locale for the user interface and databases. The locale selection for each user is stored in the Master Database with all user data.

For example, consider an MDM Hub Store that resides in an Oracle database environment in a Korean locale. Before you upgrade, you ensure that the database environment variable `NLS_LANG` is set to `KOREAN_KOREA.AL32UTF8` (Korean Unicode). After the upgrade, you can set your locale to Korean, while someone else can choose a different supported locale.

The following table lists database environment variables that you can use to set the character set:

Database	Environment Variable Name
Oracle	`NLS_LANG`
IBM Db2	`DB2CODEPAGE`

For more information about database environment variables, see the documentation for your database.

Upgrading the MDM Hub Master Database in Verbose Mode

To upgrade the MDM Hub Master Database, run the upgrade script.

Note: If you did not use the Hub Console to make metadata changes, the database upgrade script might fail. If a script fails, contact Informatica Global Customer Support.

1. Open a command prompt.
2. Navigate to the following directory:
   - In UNIX. <MDM Hub distribution directory>/database/bin
   - In Windows. <MDM Hub distribution directory>\database\bin
3. Run the MDM Hub Master Database upgrade script with the following command:
   - In UNIX. `sip_ant.sh updatemasterdatabase`
   - In Windows. `sip_ant.bat updatemasterdatabase`
4. Answer the prompts.
5. Answer the following prompt if it appears during the Master Database upgrade:

```
This upgrade should be performed by a DBA to grant 'create sequence' privileges for
the master database. The master database does not have 'create sequence' privileges,
you can either grant it now (manually) and then move forward or re-start the
```

```
upgrade, or direct this process to do so for you now, and continue the current
upgrade.
Do you want the process to create this privilege? Yes/No
```

- If enter **No**, the upgrade process checks again to ensure the user granted the privilege, and then returns to the TNS name prompt.
- If you enter **Yes** , you must answer the following prompts before you can continue with the upgrade process:

  Enter DBA username:

  Enter DBA password:

6. In JBoss environments, restart the application server.
7. Save a copy of the CMX_SYSTEM upgrade log files to the upgrade documentation directory. There is a log file for each change script.

   The upgrade process saves the files to the following location:

   - In UNIX.

     ```
     <MDM Hub installation directory>/server/resources/database/db_changes/<database
     name>/Master
     ```

   - In Windows.

     ```
     <MDM Hub installation directory>\server\resources\database\db_changes\<database
     name>\Master
     ```

Upgrading the MDM Hub Master Database in Silent Mode

To upgrade the MDM Hub Master Database in silent mode, run the upgrade script with the appropriate command for the environment.

Note: To display the silent upgrade process in the command line, set `-Dnoprompt` to `true`. To display the only the start time and end time in the command line, set `-Dnoprompt` to `true -silent`.

1. Open a command prompt.
2. Navigate to the following directory:
   - UNIX. <MDM Hub distribution directory>/database/bin
   - Windows. <MDM Hub distribution directory>\database\bin
3. To upgrade the MDM Hub Master Database in an Oracle environment, run the following command:

   UNIX.

   ```
   sip_ant.sh updatemasterdatabase -Dmaster.tnsname=<TNS name> -
   Dmaster.connectiontype=<SID or SERVICE> -Dmaster.server=<host name> -
   Dmaster.port=<port> -Dmaster.sid=<SID name> -Dmaster.username=<MDM Hub Master
   Database username> -Dmaster.password=<MDM Hub Master Database password>-
   Dcmx.username=<administrator username> -Dcmx.password=<administrator password> -
   Dcmx.server.masterdatabase.type=ORACLE -Dnoprompt=true
   ```

   Windows.

   ```
   sip_ant.bat updatemasterdatabase -Dmaster.tnsname=<TNS name> -
   Dmaster.connectiontype=<SID or SERVICE> -Dmaster.server=<host name> -
   Dmaster.port=<port> -Dmaster.sid=<SID name> -Dmaster.username=<MDM Hub Master
   Database username> -Dmaster.password=<MDM Hub Master Database password>-
   ```

```
Dcmx.username=<administrator username> -Dcmx.password=<administrator password> -
Dcmx.server.masterdatabase.type=ORACLE -Dnoprompt=true
```

4. To upgrade the MDM Hub Master Database in an IBM Db2 environment, run the following command:

   UNIX.

   ```
   sip_ant.sh updatemasterdatabase -Dcmx.server.masterdatabase.type=DB2 -
   Dmaster.hostname=<host name>
   -Dmaster.port=<port> -Dmaster.username=<MDM Hub Master Database username> -
   Dmaster.password=<MDM Hub Master Database password> -Ddba.username=<DBA username>
   -Ddba.password=<DBA password> -Dnoprompt=true
   ```

   Windows.

   ```
   sip_ant.bat updatemasterdatabase -Dcmx.server.masterdatabase.type=DB2 -
   Dmaster.hostname=<host name>
   -Dmaster.port=<port> -Dmaster.username=<MDM Hub Master Database username> -
   Dmaster.password=<MDM Hub Master Database password> -Ddba.username=<DBA username>
   -Ddba.password=<DBA password> -Dnoprompt=true
   ```

5. To upgrade the MDM Hub Master Database in a Microsoft SQL Server environment, run the following command:

   UNIX.

   ```
   sip_ant.sh updatemasterdatabase -Dmaster.hostname=<hostname> -Dmaster.port=<port> -
   Dmaster.username=<MDM Hub Master Database username>
   -Dmaster.password=<MDM Hub Master Database password>
   -Dcmx.username=<administrator username> -Dcmx.password=<administrator password> -
   Dcmx.server.masterdatabase.type=MSSQL
   -Dmaster.database=<MDM Hub Master Database name> -Dnoprompt=true
   ```

   Windows.

   ```
   sip_ant.bat updatemasterdatabase -Dmaster.hostname=<hostname> -Dmaster.port=<port> -
   Dmaster.username=<MDM Hub Master Database username>
   -Dmaster.password=<MDM Hub Master Database password>
   -Dcmx.username=<administrator username> -Dcmx.password=<administrator password> -
   Dcmx.server.masterdatabase.type=MSSQL
   -Dmaster.database=<MDM Hub Master Database name> -Dnoprompt=true
   ```

Upgrading Operational Reference Store Databases in Verbose Mode

To upgrade each Operational Reference Store (ORS) database, run an upgrade script. Upgrade the MDM Hub Master Database before you upgrade the ORS databases.

Note: If you did not use the Hub Console to make metadata changes, the database upgrade script might fail. If a script fails, contact Informatica Global Customer Support.

1. Stop the application server.
2. Open a command prompt.
3. Navigate to the following directory:
   - On UNIX. <MDM Hub distribution directory>/database/bin
   - On Windows. <MDM Hub distribution directory>\database\bin
4. Run the Operational Reference Store upgrade script with the following command:
   - UNIX. `./sip_ant.sh updateorsdatabase`
   - Windows. `sip_ant.bat updateorsdatabase`

5. Answer the prompts.

   For Oracle environments, provide the following information:

Prompts	Description
Enter database type (ORACLE, MSSQL, DB2)	Database type. Specify `Oracle`.
Enter Oracle Connection Type (service, sid). [service]	Connection type. Use one of the following values: **SERVICE** Uses the service name to connect to Oracle. **SID** Uses the Oracle System ID to connect to Oracle. Default is `SERVICE`.
Enter the Operational Reference Store database host name [localhost]	Name of the host that is running the database.
Enter the Operational Reference Store database port number. [1521]	Port number that the database listener uses. Default is `1521`.
Enter the Operational Reference Store database service name [orcl]	Name of the Oracle service. This prompt is displayed if the selected Oracle connection type is SERVICE.
Enter Oracle Net connect identifier (TNS Name) [orcl]	Oracle TNS name. Default is `orcl`.
ORS DB Connect URL: "jdbc:oracle:thin:@//<host_name>:<port>/<service_name>". Do you want to change the connect URL (y/n) [n]	Connect URL for the Oracle connection type SERVICE. You can type `y` to change the default connect URL. To use the default connect URL, type `n`.
Enter database SID [orcl]	Name of the Oracle System ID. This prompt is displayed if the selected Oracle connection type is SID.
Enter the Operational Reference Store database name [cmx_ors]	Name of the Operational Reference Store database. Default is `cmx_ors`.
Enter the Operational Reference Store database user password	Password to access the Operational Reference Store.
Enter locale name from the list: de, en_US, fr, ja, ko, zh_CN. [en_US]	Operating system locale.
Enter the DBA username [sys]	Name of the user with DBA-level permissions.
Enter the DBA password	Password of the user with DBA-level permissions.
Enter integer code of ORS Timeline Granularity: Year 5, Month 4, Day 3, Hour 2, Minute 1, Second 0 [3]	Specify timeline units to use.

For IBM Db2 environments, provide the following information:

Prompts	Description
Enter database type (ORACLE, MSSQL, DB2)	Database type. Specify `DB2`.
Enter the Operational Reference Store database host name [localhost]	Name of the host that is running the database.
Enter the Operational Reference Store database port number [50000]	Port number that the database listener uses. Default is 50000.
Enter the Operational Reference Store database name [SIP97]	Name of the database. Default is `SIP97`.
Enter the Operational Reference Store database name [cmx_ors]	Name of the Operational Reference Store database. Default is `cmx_ors`.
Enter the Operational Reference Store database user password	Password to access the Operational Reference Store.
Enter locale name from list: de, en_US, fr, ja, ko, zh_CN. [en_US]	Operating system locale. Default is `en_US`.
Enter the DBA username [sys]	Name of the user with DBA-level permissions.
Enter the DBA password	Password of the user with DBA-level permissions.
Enter integer code of ORS Timeline Granularity: Year 5, Month 4, Day 3, Hour 2, Minute 1, Second 0 [3]	Specify timeline units to use.

For Microsoft SQL Server environments, provide the following information:

Prompts	Description
Enter database type (ORACLE, MSSQL, DB2)	Database type. Specify `MSSQL`.
Enter the Operational Reference Store database host name [localhost]	Name of the host that is running the database.
Enter the Operational Reference Store database port number [1433]	Port number that the database listener uses. Default is `1433`.
Enter the Operational Reference Store database name [cmx_ors]	Name of the Operational Reference Store database. Default is `cmx_ors`.
Enter the Operational Reference Store database user password	Password to access the Operational Reference Store.
Enter locale name from list: de, en_US, fr, ja, ko, zh_CN. [en_US]	Operating system locale. Default is `en_US`.
Enter the DBA username [sys]	Name of the user with DBA-level permissions.

Prompts	Description
Enter the DBA password	Password of the user with DBA-level permissions.
Enter integer code of ORS Timeline Granularity: Year 5, Month 4, Day 3, Hour 2, Minute 1, Second 0 [3]	Specify timeline units to use.

6. Save a copy of the CMX_ORS upgrade log files to the upgrade documentation directory. There is a log file for each change script.

   The upgrade process saves the files to the following location:

   - In UNIX.

     ```
     <MDM Hub installation directory>/server/resources/database/db_changes/<database name>/ORS
     ```

   - In Windows.

     ```
     <MDM Hub installation directory>\server\resources\database\db_changes\<database name>\ORS
     ```

   Important: The `sip_ant` log file is overwritten every time you execute `sip_ant` from the command line. You must save a backup copy before you run the `sip_ant` script to upgrade another ORS.

Related Topics:

- "Saving the MDM Hub Environment Report" on page 122

Upgrading Operational Reference Store Databases in Silent Mode

To upgrade an Operational Reference Store database in silent mode, run the upgrade script with the appropriate command for the environment.

Note: To display the silent upgrade process in the command line, set `-Dnoprompt` to `true`. To display the only the start time and end time of the silent upgrade process in the command line, set `-Dnoprompt` to `true -silent`.

1. Open a command prompt.
2. Navigate to the following directory:
   - UNIX. <MDM Hub distribution directory>/database/bin
   - Windows. <MDM Hub distribution directory>\database\bin
3. To upgrade an Operational Reference Store in an Oracle environment, run the following command:

   UNIX.

   ```
   sip_ant.sh updateorsdatabase -Dors.tnsname=<tns name> -Dors.connectiontype=sid -Dors.hostname=<host name> -Dors.port=<port> -Dors.sid=<Oracle SID> -Ddba.username=<Database adminstrator username> -Ddba.password=<Database adminstrator password> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dcmx.server.masterdatabase.type=oracle -Dnoprompt=true
   ```

Windows.

```
sip_ant.bat updateorsdatabase -Dors.tnsname=<tns name> -Dors.connectiontype=sid -
Dors.hostname=<host name> -Dors.port=<port> -Dors.sid=<Oracle SID> -
Ddba.username=<Database adminstrator username> -Ddba.password=<Database adminstrator
password> -Dors.username=<ORS username> -Dors.password=<ORS password> -
Dcmx.server.masterdatabase.type=oracle -Dnoprompt=true
```

4. To upgrade the Operational Reference Store in an IBM Db2 environment, run the following command:

   UNIX.

   ```
   sip_ant.sh updateorsdatabase -Dors.hostname=<host name> -Dors.database=%db2database%
   -Dors.port=<port>
   -Dors.username=<ORS username> -Dors.password=<ORS password> -
   Dcmx.server.masterdatabase.type=db2 -Dnoprompt=true
   ```

   Windows.

   ```
   sip_ant.bat updateorsdatabase -Dors.hostname=<host name> -Dors.database=%db2database
   % -Dors.port=<port>
   -Dors.username=<ORS username> -Dors.password=<ORS password> -
   Dcmx.server.masterdatabase.type=db2 -Dnoprompt=true
   ```

5. To upgrade the Operational Reference Store in a Microsoft SQL Server environment, run the following command:

   UNIX.

   ```
   sip_ant.sh updateorsdatabase -Dors.hostname=<hostname> -Dors.port=<port> -
   Dors.username=<ORS username>
   -Dors.password=<ORS password> -Dors.database=<database name> -
   Dcmx.server.masterdatabase.type=MSSQL
    -Dnoprompt=true
   ```

   Window.

   ```
   sip_ant.bat updateorsdatabase -Dors.hostname=<hostname> -Dors.port=<port> -
   Dors.username=<ORS username>
   -Dors.password=<ORS password> -Dors.database=<database name> -
   Dcmx.server.masterdatabase.type=MSSQL
    -Dnoprompt=true
   ```

Confirm that the Upgrade Scripts Ran Successfully

Check the C_REPOS_DB_CHANGE table to see that the Hub Store upgrade scripts ran successfully.

Scripts run during the upgrade process if they have not run during previous upgrades. If the C_REPOS_DB_CHANGE table indicates that a script failed, contact Informatica Global Customer Support.

CHAPTER 6

Hub Server Upgrade (In-place Upgrade)

This chapter includes the following topics:

Hub Server Upgrade Overview

The Hub Server runs the core and common services for MDM, including access, security, and session management. The Hub Server is deployed in an application server environment.

Note: This chapter is for an in-place upgrade only. For a clean upgrade, install the Hub Server by following the instructions in the *Multidomain MDM Installation Guide* for your application server and database environment.

You can upgrade the Hub Server in graphical mode, console mode, or silent mode. To upgrade the Hub Server in graphical mode or console mode, run the Hub Server Installer provided in the distribution. To upgrade the Hub Server in silent mode, configure the silent installer properties files.

If you use a version of ActiveVOS Server that is not supported, you are prompted to install ActiveVOS Server during the Hub Server upgrade process.

The Hub Server installer differentiates a patch installation from a full installation when, during installation, you point to the existing MDM Hub installation as the target. Before overwriting the existing Hub Server installation, the MDM Hub Installer creates a backup of critical files.

Upgrading the Hub Server in Graphical Mode

To upgrade the Hub Server in graphical mode, run the Hub Server installer.

Tip: If you use embedded ActiveVOS, the upgrade process requires you to install the updated version of ActiveVOS in a new directory. If you prefer to overwrite your existing ActiveVOS installation, before you begin the upgrade, open the `../hub/server/bin/build.properties` file and comment out the following property: `activevos.install.dir = <AVOS installed path>`. For other pre-upgrade tasks, see "Prepare the BPM Upgrade" on page 23.

1. Log in using the user name that was used to install the Hub Server.
2. Start the application server on which the Hub Server is deployed.
3. Open a command prompt and navigate to the Hub Server installer in the distribution directory. By default the installer is in the following directory:
   - On UNIX. `<MDM Hub distribution directory>/<operating system name>/mrmserver`
   - On Windows. `<MDM Hub distribution directory>\windows\mrmserver`
4. Run the following command:
   - On UNIX. `hub_install.bin`
   - On Windows. `hub_install.exe`
5. From the **Introduction** page, click **Next**.

   The **Review the license agreement** page appears.
6. Review and accept the terms of the license agreement. Click **Next**.
7. Select the location of the Hub Server installation. The Hub Server installation folder contains the `siperian-mrm.ear` file. Click **Next**.
8. If use Customer 360 or Supplier 360, perform the following tasks:
   a. Ensure that all the draft records are submitted in Customer 360 or Supplier 360.

      If you do not submit the draft records, you can confirm to delete them and proceed with the upgrade.
   b. Click **Proceed**.
9. If you get the version warning message, click **OK** to upgrade an existing Hub Server installation.
10. On the **Configure server details for the Hub Console** page, enter the following details:
    - Publicly Accessible Host Name. IP address or publicly accessible host name (FQDN) of the server to which the application server binds.
    - HTTP port. HTTP port of the server that the Hub Console must use.
      If HTTPS is enabled for the application server, after the upgrade, configure the Hub Console client by editing the properties in the `build.properties` file.
11. If the previous installation uses WebLogic as the application server, the Hub Server installer prompts you to provide the WebLogic Admin password. Enter the WebLogic password.
12. If you use, or want to use, ActiveVOS for business process management, install the embedded ActiveVOS version that is supported for this version of Multidomain MDM.

    Important: Standalone ActiveVOS is not supported. If you already have a supported version of embedded ActiveVOS installed, you can skip some of the substeps in this step. Perform substep c. to enter

database details and substep 12f. to specify the user name and password for the ActiveVOS Server Administration Console.

a. Select **Yes**.

b. On the **Select an ActiveVOS installation directory** page, accept the default path or select another location. Click **Next**.

c. On the **Configure the ActiveVOS J2EE data source** page, enter the database details that you specified when you created the ActiveVOS database schema, and click **Next**.

Note: If you are moving from standalone ActiveVOS to embedded ActiveVOS, enter the details for the standalone ActiveVOS schema.

d. On the **Specify the ActiveVOS web service URL** page, accept the default URL or specify the URL that you want to use to call ActiveVOS web services. Ensure the URL contains the correct port number for the connection to the application server. Click **Next**.

The post installation setup script uses the URL to call ActiveVOS web services, deploy the predefined MDM workflows to ActiveVOS, and create the URN mapping.

e. On the **Select the ActiveVOS installer** page, click **Choose**. Browse to the ActiveVOS_Server installation file in the distribution package. Click **Next**.

f. Enter the administrative user name and password to create an administrative user for the ActiveVOS Console.

Important: The user name and password must be the same as the ActiveVOS Console user name and password that was created in the application server.

g. Click **Next**.

13. On the **Configure the MDM Product Usage Toolkit** window, select the industry to which the organization belongs and the environment type.

14. If you want to use a proxy server, select **Yes**, and enter the proxy server details. Otherwise, select **No**.

You can enter the following proxy server details:

- Proxy server name/IP
- Proxy server port
- Proxy server domain name. Leave blank if not applicable.
- Proxy server user name. Leave blank if not applicable.
- Proxy server password. Leave blank if not applicable.

15. Click **Next**.

The **Deploy the Hub Server** page appears.

16. Select whether to deploy the Hub Server EAR file automatically or manually, and then click **Next**.

Option	Conditions
Yes, run the script during this installation	Deploys the Hub Server EAR file automatically after a successful installation. Select this option if you use one of the following application server environments with one of the ActiveVOS scenarios: Application server environments: • JBoss standalone environment • WebSphere standalone environment ActiveVOS scenarios: • You chose to install ActiveVOS and no other version of ActiveVOS is installed in this environment. • You chose to install ActiveVOS and the environment includes a supported version of ActiveVOS. Check the product availability matrix (PAM) for supported versions. • You chose not to install ActiveVOS. **Important:** If you chose to install ActiveVOS but you have an unsupported version of ActiveVOS in your environment, select **No, I will run it later**.
No, I will run it later	You package and deploy the Hub Server EAR file manually. Select this option if you use one of the following application server environments, with or without ActiveVOS: • WebLogic standalone environment • WebLogic clustered environment • JBoss clustered environment • WebSphere clustered environment If you are using any application server environment that includes an unsupported version of ActiveVOS, select this option. You must upgrade the ActiveVOS schema to the supported version and then deploy manually.

The **Summary** window appears.

17. To change any options, click the **Previous** button to change your previous selections.
18. After the summary window displays the options you want, click **Install** to start the installation process.

    The Hub Server installer displays the **Please Wait** screen while the installer configures the system. The Hub Server installer backs up critical files to an archive that is stored in the `backup` folder in the MDM Hub installation directory. The file name of the archive uses the format shown in the following example:

    ```
    Informatica MDM Hub Server-2010-09-27_12-13.jar
    ```

    When the installation completes, the **Install Complete** window appears.
19. Click **Done** to exit the Hub Server installer.

    Note: If the upgrade does not complete successfully, a window appears that states that the upgrade failed and displays the location of the log file that contains the failure messages.
20. If you selected **No, I will run it later**, repackage and deploy the Hub Server EAR file.
    a. If you have an unsupported version of ActiveVOS in your environment, update the ActiveVOS schema to the supported version.
    b. Run the following command to repackage the EAR file:

       On UNIX.

       ```
       cd <MDM Hub installation directory>/hub/server/bin
       ./sip_ant.sh repackage
       ```

On Windows.

```
cd <MDM Hub installation directory>\hub\server\bin
sip_ant.bat repackage
```

c. From the application server administration console, manually deploy the Hub Server EAR file. Refer to the application server documentation.

21. Restart the application server.

Upgrading the Hub Server in Console Mode

You can upgrade the Hub Server in console mode in UNIX.

Tip: If you use embedded ActiveVOS for business process management, the upgrade process requires you to install the updated version of ActiveVOS in a new directory. If you prefer to overwrite the existing ActiveVOS installation, before you begin the upgrade, open the `../hub/server/bin/build.properties` file and comment out the following property: `activevos.install.dir = <AVOS installed path>`.

1. Start the application server.
2. Navigate to the following directory in the MDM Hub distribution:

   On UNIX. `<MDM Hub distribution directory>/<operating system name>/mrmserver`
3. Run the following command from the command prompt:

   ```
   ./hub_install.bin -i console
   ```
4. Enter the number of the locale you want to choose for the upgrade, and then press **Enter**.

   The introduction information about the upgrade appears.
5. Press **Enter**.

   The license agreement appears.
6. Read the License Agreement. Type **Y** to accept the terms of the license agreement, or type **N** if you do not want to accept the license agreement and want to exit the installation program.
7. Press **Enter**.

   If you entered **Y** in the preceding step, information about the installation folder appears.
8. Specify the directory where you installed the Hub Server.
   - To choose the default folder, press **Enter**.
   - To change the path, type the absolute path of the installation folder, and press **Enter**.
9. Confirm the location of the installation folder. Type **Y** to confirm the installation folder, or type **N** to change the installation folder.
10. If you use Customer 360 or Supplier 360, perform the following tasks:
    a. Ensure that all the draft records are submitted in Customer 360 or Supplier 360.

       If you do not submit the draft records, you can confirm to delete them and proceed with the upgrade.

    b. Click **Proceed**.
11. Enter **1** to proceed and **2** to cancel. The default option is **2**.
12. Press **Enter** to confirm that you want to proceed.

13. Configure the server details for the Hub Console:
    - IP address or fully qualified host name (FQDN) of the server to which the application server binds.
    - HTTP port of the server that the Hub Console must use.
      If HTTPS is enabled for the application server, after the upgrade, configure the Hub Console client by editing the properties in the `build.properties` file.
14. In WebLogic environments, enter your WebLogic password, and press **Enter**.
15. If you use, or want to use, embedded ActiveVOS for business process management, install the embedded ActiveVOS version that is recommended for this version of Multidomain MDM.

    Important: Standalone ActiveVOS is not supported. If you already have a supported version of embedded ActiveVOS installed, you can skip some of the substeps in this step. Perform substep c. to enter database details and substep 12f. to specify the user name and password for the ActiveVOS Server Administration Console.

    a. Press **Enter** for y=Yes.

    b. Specify the location where you want to install the ActiveVOS Server.

    c. Enter the database details that you specified when you created the ActiveVOS database schema, and click **Next**.

       Note: If you are moving from standalone ActiveVOS to embedded ActiveVOS, enter the details for the standalone ActiveVOS schema.

    d. Specify the URL that you want to use to call MDM and ActiveVOS web services. Ensure the URL contains the correct port number for the connection to the application server.

    e. On the ActiveVOS Installer page, select **Choose** and browse to the ActiveVOS_Server installation file in the distribution package.

    f. Enter a user name and password to create an administrative user for the ActiveVOS Server Administration Console.

       Important: The user name and password must be the same as the ActiveVOS Console user name and password that was created in the application server.
16. Press **Enter**.

    The Informatica platform installation prompt appears.
17. If you want to install the Informatica platform, press **Enter** for Yes. Otherwise, type `2` for No and press **Enter**.

    The prompts for the Informatica platform installation response file and archive file locations appear.
18. Enter the locations of the Informatica platform installation response file and archive file, and press **Enter**.
19. Specify the Product Usage Toolkit options.

    a. Enter the industry to which the organization belongs , and then press **Enter**.

    b. Enter the environment type. Type `1` for Production, type `2` for Test/QA, or type `3` for Development, and then press **Enter**.
20. Select whether you have a proxy server. Press **Enter** for Yes. Otherwise, type `2` for No and then press **Enter**.

    You can enter the following proxy server details:
    - Proxy server name/IP
    - Proxy server port
    - Proxy server domain name. Leave blank if not applicable.

- Proxy server user name. Leave blank if not applicable.
- Proxy server password. Leave blank if not applicable.

The summary of the installation choices appears.

21. Choose whether you want to run the `postInstallSetup` script as part of the installation, or run it manually later.
22. Choose whether you want to run the `postInstallSetup` script as part of the installation, or run it manually later.

Option	Conditions
Yes, run the script during this installation	Deploys the Hub Server EAR file automatically after a successful installation. Select this option if you use one of the following application server environments with one of the ActiveVOS scenarios: Application server environments: • JBoss standalone environment • WebSphere standalone environment ActiveVOS scenarios: • You chose to install ActiveVOS and no other version of ActiveVOS is installed in this environment. • You chose to install ActiveVOS and the environment includes a supported version of ActiveVOS. Check the product availability matrix (PAM) for supported versions. • You chose not to install ActiveVOS. **Important:** If you chose to install ActiveVOS but you have an unsupported version of ActiveVOS in your environment, select **No, I will run it later**.
No, I will run it later	You package and deploy the Hub Server EAR file manually. Select this option if you use one of the following application server environments, with or without ActiveVOS: • WebLogic standalone environment • WebLogic clustered environment • JBoss clustered environment • WebSphere clustered environment If you are using any application server environment that includes an unsupported version of ActiveVOS, select this option. You must upgrade the ActiveVOS schema to the supported version and then deploy manually.

23. Press **Enter**.

    The summary of the upgrade choices appears.
24. Verify the information in the pre-upgrade summary. If the information is correct, press **Enter** to start the upgrade. If you need to make changes, type `BACK` to the specific information and make changes.

    When the process is complete, the upgrade completion information appears.
25. Press **Enter** to exit the installer.

Upgrading the Hub Server in Silent Mode

You can upgrade the Hub Server without user interaction in silent mode. You might want to perform a silent upgrade if you have multiple installations, or if you need to upgrade on a machine cluster. A silent upgrade does not show any progress or failure messages.

Before you run the silent upgrade for the Hub Server, you must configure the properties file for the silent upgrade. The installer reads the file to determine the upgrade options. The silent upgrade process might complete successfully even if you provide incorrect settings, such as an incorrect application server path or port. You must ensure that you provide correct settings in the properties file.

Copy the Hub Server upgrade files to the hard disk on the machine where you plan to install the Hub Server. To upgrade in silent mode, complete the following tasks:

1. Configure the installation properties file and specify the installation options in the properties file.
2. Run the upgrade with the installation properties file.

Configuring the Properties File

Verify the values of the parameters in the properties file that affect the silent upgrade process.

1. Find the properties file that you configured when you installed the Hub Server.
2. If you are installing the bundled, licensed version of the ActiveVOS Server, add the ActiveVOS properties to your properties file.
   a. Open the `silentInstallServer_sample.properties` file that ships with this release.
   b. Search for ActiveVOS.
   c. Copy the ActiveVOS Installation section to your properties file.

```
################################################################################
#################### ActiveVOS Server installation ############################
################################################################################

## Do you want to install ActiveVOS (Yes/No)
AVOS.INSTALL=Yes
## Path to ActiveVOS Installer (ActiveVOS_Server_windows_9.2.4.3.exe for Windows
or ActiveVOS_Server_unix_9.2.4.3.sh for Linux/UNIX)
AVOS_INSTALLER_PATH=c\:\\ActiveVOS_Server_windows_9.2.4.3.exe
## ActiveVOS server install directory
AVOS_INSTALL_DIR=C\:\\infamdm\\avos\\server

## Database type is the same as for HUB (There is no ability to set a different
database type for ActiveVOS)
## Oracle connection data
## Connection Type SID or Service Name
AVOS.ORACLE.CONNECTION.TYPE="Service Name"
AVOS.DB.SERVER=localhost
AVOS.DB.PORT=1521
## Oracle SID name or service name
AVOS.DB.SID=orcl
AVOS.DB.SCHEMA_NAME=avos
AVOS.DB.PASSWORD=!!cmx!!

## DB2 connection data
AVOS.DB.SERVER=localhost
AVOS.DB.PORT=50000
AVOS.DB.DBNAME=AVOS
AVOS.DB.SCHEMA_NAME=AVOS
AVOS.DB.USER=avos
```

```
AVOS.DB.PASSWORD=!!cmx!!

## MSSQL connection data
AVOS.DB.SERVER=localhost
AVOS.DB.PORT=1433
AVOS.DB.DBNAME=avos
AVOS.DB.USER=avos
AVOS.DB.PASSWORD=!!cmx!!

##If you are moving from standalone ActiveVOS to embedded ActiveVOS, enter the
details for the standalone ActiveVOS schema.
```

d. In your properties file, specify the information for your ActiveVOS database and remove the properties for the other supported databases. If you need help with properties, see the *Multidomain MDM Installation Guide* for your environment.

3. If you use the licensed version of ActiveVOS server, add and configure the following properties in the silent installation properties file:

```
SIP.APPSERVER.WEB.URL=http://localhost:8080
## Avos console's administrator username
AVOS.CONSOLE.USER=aeadmin
## Avos console's administrator password
AVOS.CONSOLE.PASSWORD=admin
##The user name and password must be the same as the ActiveVOS Console user name and
password
that was created in the application server during the pre-installation process.
```

Running the Silent Upgrade

After you configure the properties file, you can start the silent upgrade.

1. Ensure that the application server is running.
2. Open a command window.
3. Run the following command:

   On UNIX. `./hub_install.bin -f <location of silent properties file for hub server>`

   On Windows. `hub_install.exe -f <location of silent properties file for hub server>`

   The silent upgrade runs in the background. The process can take a while. If you ran the postInstallSetup script for the Hub Server as part of the silent installation, check the postinstallSetup.log files to verify that the upgrade was successful.

   The log file is available in the following directory:
   On UNIX. `<MDM Hub installation directory>/hub/server/logs/`

   On Windows. `<MDM Hub installation directory>\hub\server\logs\`

Run the patchInstallSetup Script

If you chose to deploy manually during the Hub Server installation, you must run the patchInstallSetup script.

1. Navigate to the following directory: `<MDM Hub installation directory>/hub/server`
2. Run the following command to deploy the Hub Server application and apply changes to the application server configuration.

On UNIX

Note: On UNIX, if you include an exclamation mark (!) character in the password, you must include a backslash before the exclamation mark (!) character. For example, if the password is `!!cmx!!`, enter `\!\!cmx\!\!`.

WebLogic

```
patchInstallSetup.sh -Dweblogic.password=<WebLogic password> -
Ddatabase.password=<MDM Hub Master database password>
-Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

Important: In a WebLogic 12.2.1.3 or later environment, if you decided to install ActiveVOS or if you decided to use the WebLogic T3S protocol, add the options that match your decisions:

- ActiveVOS installed. `-Dinstall.avos.patch=true`
- T3S protocol used. `-Dweblogic.naming.protocol=t3s`

WebSphere with security enabled

```
patchInstallSetup.sh -Dwebsphere.password=<WebSphere password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere with security disabled

```
patchInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

JBoss

```
patchInstallsetup.sh -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

On Windows

WebLogic

```
patchInstallSetup.bat -Dweblogic.password=<WebLogic password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

Important: In a WebLogic 12.2.1.3 or later environment, if you decided to install ActiveVOS or if you decided to use the WebLogic T3S protocol, add the options that match your decisions:

- ActiveVOS installed. `-Dinstall.avos.patch=true`
- T3S protocol used. `-Dweblogic.naming.protocol=t3s`

WebSphere with security enabled

```
patchInstallSetup.bat -Dwebsphere.password=<WebSphere password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere with security disabled

```
patchInstallSetup.bat -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

JBoss

```
patchInstallsetup.bat -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

Copy Hub Server Log Files to the Upgrade Documentation Folder

Save a copy of the Hub Server log files. Use these log files assist if you need to troubleshoot the upgrade.

Copy the Hub Server log files to the `upgradedoc` upgrade documentation folder. Save these files in a separate subfolder, such as `hub_server_upgrade`. If you upgraded multiple Hub Servers in a cluster, save the files for each Hub Server instance in a separate folder.

The following table describes the log files to copy:

File	Description
`<MDM Hub installation directory>/hub/server/Infamdm_Hub_Server_InstallLog.xml`	Contains log messages for the Hub Server installation.
`<MDM Hub installation directory>/hub/server/infamdm_installer_debug.txt`	Contains debug messages and all the options that you selected when you ran the upgrade process.
`<MDM Hub installation directory>/hub/server/logs/patchInstallSetup.log`	Contains the `patchInstallSetup` script results.
`<MDM Hub installation directory>/hub/server/logs/cmxserver.log`	Contains the Hub Server log. The Hub Server creates this file when you start the Hub Server.
Application server log files.	Located in the tree under the installation directory for the application server.

Reapplying the Hub Server Upgrade (Optional)

If you complete the Hub Server upgrade, the upgrade process does not allow you to reapply the Hub Server upgrade. For example, you might want to reapply the Hub Server upgrade if hardware fails during the upgrade process. You can also perform this procedure if you test an upgrade and then want to revert to an earlier version of the software.

1. Back up the `siperian-mrm.ear` file in the following directory:
   - On UNIX. `<MDM Hub installation directory>/hub/server`
   - On Windows. `<MDM Hub installation directory>\hub\server`
2. Repeat the upgrade steps. Add the parameter `-DSIPERIAN_FORCED_PATCH_INSTALL=true` to the install command.

   For example, if you reapply the upgrade in graphical mode in UNIX, run the following command:

   ```
   hub_install.bin -DSIPERIAN_FORCED_PATCH_INSTALL=true
   ```

CHAPTER 7

Process Server Upgrade (In-place Upgrade)

This chapter includes the following topics:

Process Server Upgrade Overview

The Process Server is a servlet that handles data cleansing operations, match operations, and batch jobs. To upgrade the Process Server, run the Process Server installer provided in the distribution. The Process Server installer differentiates an upgrade from a full installation when you select the existing MDM Hub installation location as the target location during the upgrade process. Before overwriting the existing Process Server installation, the Process Server Installer creates a backup of critical files.

Note: This chapter is for an in-place upgrade only. For a clean upgrade, install the Process Server by following the instructions in the *Multidomain MDM Installation Guide* for your application server and database environment.

Upgrading the Process Server in Graphical Mode

To upgrade the Process Server in graphical mode, run the Process Server installer.

1. Log in using the user name that was used to install the Process Server.
2. Start the application server on which the Process Server is deployed.

3. Open a command prompt and navigate to the Process Server installer in the distribution directory. By default the installer is in the following directory:
   - On UNIX. `<MDM Hub distribution directory><operating system name>/mrmcleanse`
   - On Windows. `<MDM Hub distribution directory>\windows\mrmcleanse`
4. Run the following command:
   - On UNIX. `hub_cleanse_install.bin`
   - On Windows. `hub_cleanse_install.exe`
5. From the **Introduction** window, click **Next**.

   The **License Agreement** window appears.
6. Select the **I accept the terms of the License Agreement** option, and then click **Next**.

   The **Choose Install Folder** window appears.
7. Select the location of the Process Server installation. The Process Server installation folder contains the `siperian-mrm-cleanse.ear` file.
   - To choose the default location, click **Next**.
   - To choose another location, click **Choose**, and then click **Next**.

   The **Version Warning** message appears.
8. Click **OK** to confirm that you want to proceed.

   The **Enter Location of License File** window appears.
9. Select the location of the license file, and then click **Next**.
10. If the previous installation uses WebLogic as the application server, the Process Server Installer prompts you to provide the WebLogic Admin password. Enter the WebLogic password.
11. On the Product Usage Toolkit page, select the **Environment Type**.
12. If you have a proxy server, select **Yes**, and enter the proxy server details. Otherwise, select **No**, and click **Next**.

    You can enter the following proxy server details:
    - Proxy server name/IP
    - Proxy server port
    - Proxy server domain name. Leave blank if not applicable.
    - Proxy server user name. Leave blank if not applicable.
    - Proxy server password. Leave blank if not applicable.
13. Click **Next**.

    The Deploy page appears.
14. Select whether to deploy automatically or manually. click **Yes** to deploy automatically, or click **No** to deploy manually, and then click **Next**.
    - On WebSphere standalone environments or JBoss standalone environments, click **Yes** to deploy automatically, and then click **Next**.
    - On WebLogic or clustered environments, click **No** to deploy manually, and then click **Next**.

    The **Pre-Installation Summary** window appears.
15. To change any options, click the **Previous** button to change your previous selections.
16. After the summary window displays the options you want, click **Install** to start the installation process.

The Process Server installer displays the **Please Wait** screen while the installer configures the system. The Process Server installer backs up critical files to an archive that is stored in the `backup` folder in the MDM Hub installation directory. The file name of the archive uses the format shown in the following example:

```
Siperian Hub Cleanse Match Server-2010-05-12_18-09.jar
```

When the installation completes, the **Install Complete** window appears.

17. Click **Done** to exit the Process Server installer.

    Note: If the upgrade does not complete successfully, a window appears that states that the upgrade failed and displays the location of the log file that contains the failure messages.

18. If you selected **No** in step 14, repackage and manually deploy the EAR file.

    a. Run the following command to repackage the EAR file:

    On UNIX.

    ```
    cd <MDM Hub installation directory>/hub/cleanse/bin
    ./sip_ant.sh repackage
    ```

    On Windows.

    ```
    cd <MDM Hub installation directory>\hub\cleanse\bin
    sip_ant.bat repackage
    ```

    b. From the application server administration console, manually deploy the Process Server EAR file. Refer to the application server documentation.

19. Copy the SSA-Name3 library files from `<MDM Hub installation directory>/hub/cleanse/lib/upgrade/SSA` to `<MDM Hub installation directory>hub/cleanse/lib`.

20. Restart the application server.

Upgrading the Process Server in Console Mode

You can upgrade the Process Server in console mode on UNIX.

Note: Do not use the root user when you upgrade the Process Server on RedHat Linux. The root user does not have a .profile, which InstallAnywhere requires. Instead, create and use a separate user profile to upgrade the Process Server.

1. Start the application server.
2. Navigate to the following directory in the MDM Hub distribution:

   On Solaris. `<MDM Hub distribution directory>/solaris/mrmcleanse`

   On HP-UX. `<MDM Hub distribution directory>/hpux/mrmcleanse`

   On Linux. `<MDM Hub distribution directory>/linux/mrmcleanse`

   On AIX. `<MDM Hub distribution directory>/aix/mrmcleanse`

3. Run the following command from the command prompt:

   ```
   ./hub_cleanse_install.bin -i console
   ```

4. Enter the number of the locale you want to choose for the installation, and then press **Enter**.

   The introduction information about the installation appears.

5. Press **Enter**.

   The license agreement appears.

6. Read the License Agreement. Type **Y** to accept the license agreement, or type **N** if you do not want to accept the license agreement and want to exit the installation program.
7. Press **Enter**.

   If you entered **Y** in the preceding step, information about the installation folder appears.
8. Specify the directory where you installed the Process Server.
   - To choose the default location, press **Enter**.
   - To change the path, type the absolute path of the installation folder, and press **Enter**.
9. Confirm the location of the installation folder. Type **Y** to confirm the installation folder, or type **N** to change the installation folder.

   The version warning message appears.
10. Press **Enter** to confirm that you want to proceed.

    The prompt for the license file location appears.
11. Enter the absolute path of the license file, and press **Enter**.
12. In WebLogic environments, enter your WebLogic password, and press **Enter**.
13. From the Product Usage Toolkit options, select the environment type. Type `1` for Production, type `2` for Test/QA, or type `3` for Development, and then press **Enter**.
14. Select whether you have a proxy server. Press **Enter** for Yes. Otherwise, type `2` for No and then press **Enter**.

    You can enter the following proxy server details:
    - Proxy server name/IP
    - Proxy server port
    - Proxy server domain name. Leave blank if not applicable.
    - Proxy server user name. Leave blank if not applicable.
    - Proxy server password. Leave blank if not applicable.

    The summary of the installation choices appears.
15. Choose whether you want to run the `postInstallSetup` script as part of the installation, or run it manually later.
16. Press **Enter**.

    The summary of the upgrade choices appears.
17. Verify the information in the pre-upgrade summary. If the information is correct, press **Enter** to start the upgrade. If you need to make changes, type `BACK` to the specific information and make changes.

    When the process is complete, the upgrade completion information appears.
18. Press **Enter** to exit the installer.

Upgrading the Process Server in Silent Mode

You can upgrade the Process Server without user interaction in silent mode. You might want to perform a silent upgrade if you have multiple installations, or if you need to upgrade on a machine cluster. A silent upgrade does not show any progress or failure messages.

Before you run the silent upgrade for the Process Server, you must configure the properties file for the silent upgrade. The installer reads the file to determine the upgrade options. The silent upgrade process might complete successfully even if you provide incorrect settings, such as an incorrect application server path or port setting. You must ensure that you provide correct settings in the properties file.

Copy the Process Server upgrade files to the hard disk on the machine where you plan to upgrade the Process Server. To upgrade in silent mode, complete the following tasks:

1. Configure the installation properties file and specify the installation options in the properties file.
2. Run the upgrade with the installation properties file.

Configuring the Properties File

Verify the values of the parameters in the properties file that affect the silent upgrade process.

1. Find the properties file that you configured when you installed the Process Server.
2. Use a text editor to open the file and verify the values of the parameters that affect the silent upgrade process.

   The following table describes the upgrade parameters to verify:

Property Name	Description
USER_INSTALL_DIR	Directory where you installed the Process Server. For example, `C\:\\<MDM Hub installation directory>\\cleanse`. You must escape backslash characters in the properties file. Use double backslashes when you specify the installation directory path.
SIP.APPSERVER.PASSWORD	Password to access WebLogic. For WebLogic environments.
RUN_DEPLOYMENT_FLAG	Runs the postInstallSetup script as part of the silent upgrade. Set to `1` if you want to run postInstallSetup at the end of the silent upgrade. Set to `0` if you do not want to run postInstallSetup.

3. Add and configure the following product usage toolkit properties to the silent installation properties file:

```
#Product Usage Toolkit Installation
#CSM_TYPE is the type of Product Usage Toolkit installation.
# valid values are:Production,Test,Development. Should not be blank.
CSM_TYPE=Production

# If the network haa a proxy server, fill in the following parameters (leave empty
if no proxy):
# proxy server host
CSM_HOST=
# proxy server port
CSM_PORT=
# Proxy server domain name (leave blank, if not applicable)
CSM_DOMAIN=
# Proxy server user name (leave blank, if not applicable)
CSM_PROXY_USER_NAME=
#Proxy server password (leave blank, if not applicable)
CSM_PROXY_PASSWORD=
```

Running the Process Server Silent Upgrade

After you configure the properties file, you can start the silent upgrade.

1. Ensure that the application server is running.
2. Open a command window.
3. Run the following command:

   On UNIX. `./hub_cleanse_install.bin -f <location of silent properties file for process server>`

   On Windows. `hub_cleanse_install.exe -f <location of silent properties file for process server>`

   The silent upgrade runs in the background. The process can take a while. If you ran the post install script for the Process Server as part of the silent installation, check the postinstallSetup.log files to verify that the upgrade was successful.

   The log file is available in the following directory:
   On UNIX. `<MDM Hub installation directory>/hub/cleanse/logs/`

   On Windows. `<MDM Hub installation directory>\hub\cleanse\logs\`

Steps to Upgrade to Informatica Address Verification 5 Integration

This section describes the upgrade process required for the MDM Hub implementation to use Informatica Address Verification 5.

Note: This section is applicable to users with a license for using Informatica Address Verification.

You must perform the following steps to upgrade to Informatica Address Verification 5 integration:

1. Open the `cmxcleanse.properties` file.This file is located at:

   Windows: *<infamdm_install_directory>*\hub\cleanse\resources

   UNIX: *<infamdm_install_directory>*/hub/cleanse/resources

2. Ensure that the following Informatica Address Verification 5 properties are set in the `cmxcleanse.properties` files:

   Windows:

   ```
   cleanse.library.addressDoctor.property.SetConfigFile=C:\infamdm\hub\cleanse\resources
   \AddressDoctor\5\SetConfig.xml
   cleanse.library.addressDoctor.property.ParametersFile=C:\infamdm\hub\cleanse
   \resources
   \AddressDoctor\5\Parameters.xml
   cleanse.library.addressDoctor.property.DefaultCorrectionType=PARAMETERS_DEFAULT
   ```

   UNIX:

   ```
   cleanse.library.addressDoctor.property.SetConfigFile=/u1/infamdm/hub/cleanse/
   resources/
   AddressDoctor/5/SetConfig.xml
   cleanse.library.addressDoctor.property.ParametersFile=/u1/infamdm/hub/cleanse/
   resources/
   AddressDoctor/5/Parameters.xml
   cleanse.library.addressDoctor.property.DefaultCorrectionType=PARAMETERS_DEFAULT
   ```

3. Save and close the properties file.

4. Copy `SetConfig.xml` and `Parameters.xml` to the location specified in the `cmxcleanse.properties` file.

   The following is a sample `SetConfig.xml` file:

```
<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'>
<SetConfig>
    <General WriteXMLEncoding="UTF-16" WriteXMLBOM="NEVER"
    MaxMemoryUsageMB="600" MaxAddressObjectCount="10" MaxThreadCount="10" />

    <UnlockCode>79FYL9UAXAVSR0KLV1TDC6PAQVVC3KM14FZC</UnlockCode>

    <DataBase CountryISO3="ALL" Type="BATCH_INTERACTIVE" Path="c:\addressdoctor\5"
        PreloadingType="NONE" />

    <DataBase CountryISO3="ALL" Type="FASTCOMPLETION" Path="c:\addressdoctor\5"
        PreloadingType="NONE" />

    <DataBase CountryISO3="ALL" Type="CERTIFIED" Path="c:\addressdoctor\5"
        PreloadingType="NONE" />

    <DataBase CountryISO3="ALL" Type="GEOCODING" Path="c:\addressdoctor\5"
        PreloadingType="NONE" />

        <DataBase CountryISO3="ALL" Type="SUPPLEMENTARY" Path="c:\addressdoctor\5"
        PreloadingType="NONE" />
</SetConfig>
```

   The following is a sample `Parameters.xml` file:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE Parameters SYSTEM 'Parameters.dtd'>
<Parameters
    WriteXMLEncoding="UTF-16"
    WriteXMLBOM="NEVER">
    <Process
        Mode="BATCH"
        EnrichmentGeoCoding="ON"
        EnrichmentCASS="ON"
        EnrichmentSERP="ON"
        EnrichmentSNA="ON"
        EnrichmentSupplementaryGB="ON"
        EnrichmentSupplementaryUS="ON" />
    <Input
        Encoding="UTF-16"
        FormatType="ALL"
        FormatWithCountry="ON"
        FormatDelimiter="PIPE" />
    <Result
        AddressElements="STANDARD"
        Encoding="UTF-16"
        CountryType="NAME_EN"
        FormatDelimiter="PIPE" />
</Parameters>
```

5. Specify the Informatica Address Verification 5 unlock code in the configuration file, `SetConfig.xml`.

   For more information about the `SetConfig.xml` file and `Parameters.xml` file, refer to your Informatica Address Verification 5 documentation.

6. Copy the Informatica Address Verification 5 library from the following location:

   Windows: <*infamdm_install_directory*>\hub\cleanse\lib\upgrade\AddressDoctor

   UNIX: <*infamdm_install_directory*>/hub/cleanse/lib/upgrade/AddressDoctor

7. Replace `JADE.dll` (or equivalent Informatica Address Verification 4 library) with the Informatica Address Verification 5 library at the following location:

   Windows: <*infamdm_install_directory*>\hub\cleanse\lib

   UNIX: <*infamdm_install_directory*>/hub/cleanse/lib

For more information, refer to the `libupdate_readme.txt` document available at:

Windows: `<infamdm_install_directory>\hub\cleanse\lib\upgrade`

UNIX: `<infamdm_install_directory>/hub/cleanse/lib/upgrade`

8. Restart the application server.

   Ensure that you are logged in with the same user name that is currently running the application server and that no exceptions occur while starting the application server.

9. Restart the Process Server.

   During the Process Server initialization, you should see a message similar to the following in the terminal console:

   ```
   [INFO ] com.siperian.mrm.cleanse.addressDoctor.Library: Initializing AddressDoctor5
   ```

10. Start the Cleanse Functions tool.
11. Obtain a write lock (**Write Lock** > **Acquire Lock**).
12. Select the Informatica Address Verification cleanse function.
13. Click the **Refresh** button.

    The Informatica Address Verification 5 cleanse function is added to the Informatica Address Verification cleanse functions node.

Configure Match Population

The match population contains the standard population set to use for the match process. Each supported country, language, or population has a standard population set. You must enable the match population to use for the match rules.

The match population is available as a *population*.ysp file with the Informatica MDM Hub installation. The population name is the same as the ysp file name. If you add a Japanese population, and want to use the Person_Name_Kanji match field, add _Kanji to the population name. For example, Japan_Kanji or Japan_i_Kanji. If you do this, the standard Person_Name match field is not available.

The population that you use must be compatible with the SSA-Name3 version of the MDM Hub. If you need additional population files or if you need an updated population file to upgrade to a later version, contact Informatica Global Customer Support. The first population file that you request with the product is free. You might need population files for other countries or you might need an updated population file to upgrade to a later version of the MDM Hub.

Upgrading custom population files

During the upgrade process, the population files are upgraded. If you have customized your population files, contact Informatica Global Customer Support to have the customizations applied to the upgraded properties file.

Enabling Match Population

You must enable the match population to use for the match rules.

1. Copy the `<population>.ysp` files to the following location:

   On UNIX. `<MDM Hub installation directory>/hub/cleanse/resources/match`

   On Windows. `<MDM Hub installation directory>\hub\cleanse\resources\match`

2. In the C_REPOS_SSA_POPULATION metadata table, verify that the population is registered.

   The seed database for the MDM Hub installation has some populations registered in the C_REPOS_SSA_POPULATION table, but not enabled.

3. If the C_REPOS_SSA_POPULATION table does not contain the population, add it to the table and enable it.

   The population name is the same as the ysp file name. For example, if the ysp file name is `US.ysp`, the population name is US.
   To add the population to an Operational Reference Store, use the following steps:

   On IBM Db2 or Oracle.

   a. Connect to the Operational Reference Store schema that you want to add the population to.

   b. In SQL*Plus, run the `add_std_ssa_population.sql` script in the following directory:

      On UNIX. `<MDM Hub installation directory>/server/resources/database/custom_scripts/oracle`

      On Windows. `<MDM Hub installation directory>\server\resources\database\custom_scripts\oracle`

   c. Answer the prompts described in the following table:

Prompt	Description
Enter the population to add	Name of the population.
Enter a value for ROWID_SSA_POP (example: INFA.0001) DEFAULT [INFA.0001]	Unique value for the ROWID_SSA_POP column of the C_REPOS_SSA_POPULATION metadata table. Default is `INFA.0001`

      The population is registered in the C_REPOS_SSA_POPULATION table.

   d. Run the following command to enable the population:

```
UPDATE c_repos_ssa_population SET enabled_ind = 1 WHERE population_name = '<Your Population> ';
COMMIT;
```

   On Microsoft SQL Server

   a. Run the `add_std_ssa_population.bat` script in the following directory:

      `<MDM Hub installation directory>\server\resources\database\custom_scripts\MSSQL`

   b. Answer the prompts described in the following table:

Prompt	Description
Hostname with MSSQL instance for CMX_ORS DB ("localhost")	Host name of the Microsoft SQL Server instance.
cmx_ors user name ("cmx_ors")	User name of the Operational Reference Store.
cmx_ors user password	Password of the Operational Reference Store.

Prompt	Description
Enter the population name (Note: If you use Person_Name_Kanji for the Japan population or Japan_i population, add the suffix '_Kanji' to the end of the name) DEFAULT (" ")	Name of the population.
Enter a value for ROWID_SSA_POP (example: INFA.0001) DEFAULT (INFA.0001)	Unique value for the ROWID_SSA_POP column of the C_REPOS_SSA_POPULATION metadata table.

The population is registered in the C_REPOS_SSA_POPULATION table.

c. Run the following command to enable the population:

```
USE <Operational Reference Store user>
GO
UPDATE [dbo].[C_REPOS_SSA_POPULATION] SET ENABLED_IND = 1 WHERE POPULATION_NAME
= '<population>'
```

4. Restart the Process Server.
5. Log in to the Hub Console to verify that the population is enabled.

   The population appears in the **Match/Merge Setup** user interface for base objects.

Copy Process Server Log Files to the Upgrade Documentation Directory

Save a copy of the Hub Server log files. Use these log files assist if you need to troubleshoot the upgrade.

Copy the Process Server log files to the upgrade documentation folder. Save these files in a separate subfolder, such as `cleanse_match_server_upgrade`. If you upgraded multiple Process Servers in a cluster, save the files for each Process Server instance in a separate folder.

The following table describes the log files to copy:

File	Description
`<cleanse installation directory>/hub/cleanse/Infamdm_Hub_Cleanse_Match_Server_InstallLog.xml`	Contains log files for the Process Server installation.
`<cleanse installation directory>/hub/cleanse/infamdm_installer_debug.txt`	Contains debug messages and all the options that you selected when you ran the upgrade process.
`<cleanse installation directory>/hub/cleanse/logs/patchInstallSetup.log`	Contains the patchInstallSetup script results.
`<cleanse installation directory>/hub/cleanse/logs/cmxserver.log`	Contains the Process Server logs.
Application server log files.	Located in the tree under the installation directory for the application server.

Reapplying the Process Server Upgrade (Optional)

If you complete the Process Server upgrade, the upgrade process does not allow you to reapply the Process Server upgrade. For example, you might want to reapply the Process Server upgrade if hardware fails during the upgrade process. You can also perform this procedure if you test an upgrade and then want to revert to an earlier version of the software.

1. Back up the `siperian-mrm.ear` file in the following directory:
   - On UNIX. `<MDM Hub installation directory>/hub/cleanse`
   - On Windows. `<MDM Hub installation directory>\hub\cleanse`
2. Repeat the upgrade steps. Add the parameter `-DSIPERIAN_FORCED_PATCH_INSTALL=true` to the install command.

   For example, if you reapply the upgrade in graphical mode in UNIX, run the following command:

   ```
   hub_cleanse_install.bin -DSIPERIAN_FORCED_PATCH_INSTALL=true
   ```

CHAPTER 8

Post-Upgrade Tasks

This chapter includes the following topics:

Post-Upgrade Tasks

Whether you perform a clean upgrade or an in-place upgrade, perform the post-upgrade tasks to ensure your environment is properly configured.

Configure JDBC Driver for Microsoft SQL Server 2017

If you installed the Hub Server in an environment that uses Microsoft SQL Server 2017, copy the correct version of the JDBC driver file to the Hub Server `lib` directory.

1. Find latest supported version of the Microsoft JDBC driver in the `Binn` directory on the machine on which Microsoft SQL Server is installed.

   When you perform the pre-installation tasks, you download and copy the driver file from the Microsoft website to the `Binn` directory.

2. Copy the Microsoft JDBC JAR file to the following directory:

   `<MDM Hub installation directory>/hub/server/lib`

3. Restart the application server.

Update Properties

Upgrade Task	Details
Update the Application Server Settings in the Properties Files	If you upgrade the application server, you must manually update the application server settings in the properties files. In the `build.properties` file located in `<MDM Hub installation directory>\hub\server\bin`, update the following settings: - `SIP.AS.HOME` - `SIP.AS.SERVER_FOLDER` - `SIP.AS.DEPLOY_FOLDER` In the `setSiperianEnv.bat` file located in `<MDM Hub installation directory>\hub\server`, update the following settings: - `SET JBS_HOME` - `SET JBS_SERVER_DIR` - `SET JBS_DEPLOY_DIR` - `SET JBS_CLIENT_CLASSPATH` In the `cmxserver.properties` file located in `<MDM Hub installation directory>\hub\server\resources`, update the following settings: - `cmx.appserver.version` If you have changed any other application server configuration settings, such as port numbers, you must also update the settings in `cmxserver.properties`.

JBoss Post-Upgrade Tasks

In JBoss environments, perform the following tasks:

Task	Action

Remove the `odjbc6.jar` file	Remove the JAR file from the following locations:
	`<JBoss install location>\modules\com\activevos\main`
	`<JBoss install location>\modules\com\informatica\mdm\jdbc\main`
	`<MDM Hub installation directory>\hub\server\lib`

Optionally, update the JDBC driver to `sqljdbc42.jar`	1. Install the JDBC driver as a core module. For instructions, see the JBoss documentation. 2. Update the `datasource` definitions in the JBoss `standalone-full.xml` file to use the module.

Perform Post-Upgrade Tasks for In-place Upgrade

After an in-place upgrade, you must perform some post-upgrade tasks.

The following table provides details of the post-upgrade tasks:

Upgrade Task	Details
Clear the Java cache	1. Clear the Java cache. For instructions, see the Java documentation. 2. Launch the Hub Console.

Configure logging	To optimize the performance of Data Director, configure logging by editing the `log4j.xml` file.
	1. Open `log4j.xml` in the following directory:
	`<MDM Hub installation directory>/hub/server/conf`
	2. Add the following logging configuration:
	`<category name="org.eclipse.persistence.sdo"> <priority value="WARN"/> </category> <category name="org.eclipse.persistence.default"> <priority value="WARN"/> </category>`
	3. Save and close the `log4j.xml` file.

Drop Objects, Columns, and References to Deprecated Objects

Upgrade Task	Details
Update references to REL_START_DATE, REL_END_DATE, and HUID tables	Update references to the REL_START_DATE and REL_END_DATE system columns and to the HUID table. The upgrade process removes references to the REL_START_DATE and REL_END_DATE columns in the packages and views used in a Hierarchy Manager base object. Change references from REL_START_DATE to PERIOD_START_DATE. Change references from REL_END_DATE to PERIOD_END_DATE.

Run the PostInstall Script for Deploying the Hub Server (Conditional)

If you skipped the `postInstallSetup` script during the installation, run the script. The post-installation process deploys the Hub Server applications, creates data sources, and configures JMS message queues.

If you use a WebLogic Server environment with Managed Servers where the Administration Server and the Managed Servers are on different machines, copy all the deployment files to the MDM Hub installation directory of the Administration Server. For more information, see the *Multidomain MDM Installation Guide*.

- Open a command prompt and run the `postInstallSetup` script the following directory: `<MDM Hub installation directory>/hub/server`

  Note: If you did not install the ActiveVOS version that is bundled with the MDM Hub installer, do not include the ActiveVOS user names and passwords in the command. On UNIX, if you include the exclamation mark (!) in your password, you must include a backslash before the exclamation mark. For example, if your password is !!cmx!!, enter the following password: \!\!cmx\!\!

  For WebSphere with security disabled:

  ```
  ./postInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password>
  -Davos.username=<ActiveVOS Console username>
  -Davos.password=<ActiveVOS Console password>
  -Davos.jdbc.database.username=<ActiveVOS database username>
  -Davos.jdbc.database.password=<ActiveVOS database password>
  ```

  For WebSphere with security-enabled:

  ```
  ./postInstallSetup.sh -Dwebsphere.password=<WebSphere password>
  -Ddatabase.password=<MDM Hub Master database password>
  -Davos.username=<ActiveVOS Console username>
  -Davos.password=<ActiveVOS Console password>
  -Davos.jdbc.database.username=<ActiveVOS database username>
  -Davos.jdbc.database.password=<ActiveVOS database password>
  ```

  For JBoss:

  ```
  ./postInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password>
  -Davos.username=<ActiveVOS Console username>
  -Davos.password=<ActiveVOS Console password>
  -Davos.jdbc.database.username=<ActiveVOS database username>
  -Davos.jdbc.database.password=<ActiveVOS database password>
  ```

For WebLogic:

```
./postInstallSetup.sh -Dweblogic.password=<WebLogic password>
-Ddatabase.password=<MDM Hub Master database password>
-Davos.username=<ActiveVOS Console username>
-Davos.password=<ActiveVOS Console password>
-Davos.jdbc.database.username=<ActiveVOS database username>
-Davos.jdbc.database.password=<ActiveVOS database password>
```

Important: In a WebLogic 12.2.1.3 or later environment, if you decided to install ActiveVOS or if you decided to use the WebLogic T3S protocol, add the options that match your decisions:

- ActiveVOS installed. `-Dinstall.avos.patch=true`
- T3S protocol used. `-Dweblogic.naming.protocol=t3s`

The ActiveVOS Console credentials are the same credentials as the administrative user in the application server.

The ActiveVOS database credentials are the same credentials that were used to run the `create_bpm` script.

If you deploy in a WebLogic environment with Managed Servers, ensure that you specify all the Managed Servers as the targets for deployments in the WebLogic Server Administration Console.

For more information, see the *Multidomain MDM Installation Guide* and the WebLogic documentation.

Configure the Hub Console Client

To configure the Hub Console client to connect to the Hub Server machine, edit the `cmxserver.properties` file and repackage the Hub Server application.

Note: You can override the host name and port number when you launch the Hub Console.

1. Set relevant values for the`cmx.appserver.console.mode` property in the `cmxserver.properties` file in the following directory:

   `<MDM Hub installation directory>/hub/server/resources`

   Set the value to the communication protocol that you use, either HTTP or HTTPS.
2. Navigate to the following directory:

   `<MDM Hub installation directory>/hub/server/bin`
3. Run the following command:

   On UNIX.

   ```
   ./sip_ant.sh repackage -Dconsole.hostname=<host name> -Dconsole.webport=<port>
   ```

   On Windows.

   ```
   sip_ant.bat repackage -Dconsole.hostname=<host name> -Dconsole.webport=<port>
   ```

   Where host name is the IP address or publicly accessible host name of the server that the application server binds to and port is the HTTP or HTTPS port of the current node that the Hub Console must use.

Configure WebSphere Administrative Security

You can configure WebSphere administrative security to control MDM Hub access to the WebSphere administrative console.

To configure WebSphere administrative security, perform the following steps:

1. Unregister the Operational Reference Stores (ORS).
2. Uninstall the EAR files and remove data sources from WebSphere.
3. Enable WebSphere administrative security in the WebSphere administrative console.
4. Configure the Hub Server and Process Server properties.
5. Run the Hub Server and Process Server PostInstallSetup scripts.
6. Register the ORS.

Enter your WebSphere credentials when you register the ORS. You do not need to enter your credentials after you verify the ORS.

Unregister the Operational Reference Store

To unregister the Operational Reference Store (ORS), use the Databases tool in the MDM Hub Console.

1. From the MDM Hub Console, click **Write Lock** > **Acquire Lock**.
2. From the **Configuration** workbench, select the **Databases** tool.

   The **Database Information** page appears.
3. From the list of databases, select the ORS to unregister.
4. Click **Unregister database**.

   The Database tool prompts you to confirm that you want to unregister the ORS.
5. Click **Yes.**

Uninstall the EAR files and Remove Data Sources

To uninstall the EAR files and remove data sources, use the WebSphere administrative console.

1. Use the WebSphere administrative console to undeploy the following deployment files:

Deployment File Name	Description
siperian-mrm.ear	Required. The Hub Server application.
provisioning-ear.ear	Required. The Provisioning tool application.
entity360view-ear.ear	Optional. The Entity 360 framework.

2. Use the WebSphere administrative console to remove all data sources for the MDM Hub Master Database and the Operational Reference Stores.
3. Restart the application server.

For more information, see the WebSphere documentation.

Enable WebSphere Administrative Security in the WebSphere Administrative Console

You must enable WebSphere administrative security in the WebSphere administrative console. When you enable WebSphere administrative security, disable WebSphere application security.

For more information, see the WebSphere documentation.

Configure the Hub Server and Process Server Properties

You must configure the Hub Server and Process Server property files to enable WebSphere administrative security.

1. Stop the application server.
2. Enable WebSphere security on the Hub Server.
   a. Open `cmxserver.properties` in the following directory:

      On UNIX. `<MDM Hub installation directory>/hub/server/resources`

      On Windows. `<MDM Hub installation directory>\hub\server\resources`

   b. Set `cmx.websphere.security.enabled` to `true`.
3. Enable WebSphere security on the Process Server.
   a. Open `cmxcleanse.properties` in the following directory:

      On UNIX. `<MDM Hub installation directory>/hub/cleanse/resources`

      On Windows. `<MDM Hub installation directory>\hub\cleanse\resources`

   b. Set `cmx.websphere.security.enabled` to `true`.
4. Configure the WebSphere user name on the Hub Server.
   a. Open `build.properties` in the following directory:

      On UNIX. `<MDM Hub installation directory>/hub/server/bin`

      On Windows. `<MDM Hub installation directory>\hub\server\bin`

   b. Set `websphere.username` to the WebSphere administrative user name.
5. Configure the WebSphere user name on the Process Server.
   a. Open `build.properties` in the following directory:

      On UNIX. `<MDM Hub installation directory>/hub/cleanse/bin`

      On Windows. `<MDM Hub installation directory>\hub\cleanse\bin`

   b. Set `websphere.username` to the WebSphere administrative user name.
6. In SQL* Plus, run the following command to set `c_repos_cleanse_match_server.is_secured` to `1`.

```
UPDATE c_repos_cleanse_match_server set is_secured = 1 where
rowid_cleanse_match_server='<Insert value here>';
COMMIT;
```

7. Start the application server.

Run the Hub Server PostInstallSetup Script Manually

You must run the Hub Server PostInstallSetup script.

1. Open a command prompt.

2. Navigate to the PostInstallSetup script in the following directory:

   On UNIX. `<MDM Hub installation directory>/hub/server`

   On Windows. `<MDM Hub installation directory>\hub\server`

3. Run the following command:

   On UNIX. postinstallsetup.sh -Ddatabase.password=<MDM Hub Master Database> -Dwebsphere.password=<WebSphere administrative user password>

   On Windows. postinstallsetup.bat -Ddatabase.password=<MDM Hub Master Database> -Dwebsphere.password=<WebSphere administrative user password>

4. Restart the application server.

Run the Process Server PostInstallSetup Script

You must run the Process Server PostInstallSetup script.

1. Open a command prompt.
2. Navigate to the PostInstallSetup script in the following directory:

   On UNIX. `<MDM Hub installation directory>/hub/cleanse`

   On Windows. `<MDM Hub installation directory>\hub\cleanse`

3. Run the following command:

   On UNIX. `postinstallsetup.sh -Dwebsphere.password=<websphere administrative user password>`

   On Windows. `postinstallsetup.bat -Dwebsphere.password=<websphere administrative user password>`

4. Restart the application server.

Register the Operational Reference Stores

To register the Operational Reference Stores, use the Hub Console.

1. Start the **Databases** tool under the **Configuration** workbench.
2. Click **Write Lock** > **Acquire Lock**.
3. Click **Register database**.

   The **Informatica MDM Hub Connection Wizard** appears and prompts you to select the database type.
4. Select Microsoft SQL Server, Oracle, or IBM Db2, and click **Next**.
5. In Microsoft SQL Server, configure connection properties for the database.
   a. In the Connection Properties page, specify the connection properties, and then click **Next**.

The following table lists and describes the connection properties:

Property	Description
Database Display Name	Name for the Operational Reference Store that must appear in the Hub Console.
Machine Identifier	Prefix given to keys to uniquely identify records from the Hub Store instance.
Database hostname	IP address or name of the server that hosts the Microsoft SQL Server database.
Port	Port of the Microsoft SQL Server database. The default is 1433.
Schema Name	Name of the Operational Reference Store.
Password	Password associated with the user name for the Operational Reference Store.
Dynamic Data Masking host	IP address or name of the server that hosts Dynamic Data Masking. Leave empty if you do not use Dynamic Data Masking.
DDM connection URL	Optional. URL for the Dynamic Data Masking server. Leave empty if you do not use Dynamic Data Masking.

b. In the Connection Properties page, specify the connection properties, and then click **Next**.

c. Review the summary, and specify additional connection properties.

The following table lists additional connection properties that you can configure:

Property	Description
Connection URL	Connect URL. The Connection Wizard generates the connect URL by default.
Create datasource after registration	Select to create the datasource on the application server after registration.

6. In Oracle environments, configure connection properties for the database.

a. Select an Oracle connection method, and click **Next**.

The following table describes the Oracle connection methods that you can select:

Connection Method	Description
Service	Connect to Oracle by using the service name.
SID	Connect to Oracle by using the Oracle System ID.

For more information about SERVICE and SID names, see the Oracle documentation.

The **Connection Properties** page appears.

b. Specify the connection properties for the connection type that you select, and click **Next**.

The following table lists and describes the connection properties:

Property	Description
Database Display Name	Name for the Operational Reference Store that must appear in the Hub Console.
Machine Identifier	Prefix given to keys to uniquely identify records from the Hub Store instance.
Database hostname	IP address or name of the server that hosts the Oracle database.
SID	Oracle System Identifier that refers to the instance of the Oracle database running on the server. The **SID** field appears if you selected the **SID** connection type.
Service	Name of the Oracle SERVICE used to connect to the Oracle database. The **Service** field appears if the you selected the **Service** connection type.
Port	The TCP port of the Oracle listener running on the Oracle database server. The default is `1521`.
Oracle TNS Name	Name by which the database is known on your network as defined in the `TNSNAMES.ORA` file of the application server. For example: `mydatabase.mycompany.com`. You set the Oracle TNS name when you install the Oracle database. For more information about the Oracle TNS name, see the Oracle documentation.
Schema Name	Name of the Operational Reference Store.
User name	User name for the Operational Reference Store. By default, this is the user name that you specify in the script that you use to create the Operational Reference Store. This user owns all of the Operational Reference Store database objects in the Hub Store. If a proxy user is configured for the Operational Reference Store, then you can specify the proxy user instead.
Password	Password associated with the user name for the Operational Reference Store. For Oracle, the password is not case sensitive. By default, this is the password that you specify when you create the Operational Reference Store. If a proxy user is configured for the Operational Reference Store, then you specify the password for the proxy user instead.
Dynamic Data Masking host	IP address or name of the server that hosts Dynamic Data Masking. Leave empty if you do not use Dynamic Data Masking.
DDM connection URL	Optional. URL for the Dynamic Data Masking server. Leave empty if you do not use Dynamic Data Masking.

Note: The **Schema Name** and the **User Name** are both the names of the Operational Reference Store that you specified when you created the Operational Reference Store. If you need this information, consult your database administrator.

The **Summary** page appears.

c. Review the summary, and specify additional connection properties.

The following table lists additional connection properties that you can configure:

Property	Description

Connection URL	Connect URL. The Connection Wizard generates the connect URL by default. The following list shows the format of the connect URL for the Oracle connection types:
	Service connection type
	`jdbc:oracle:thin:@//database_host:port/service_name`
	SID connection type
	`jdbc:oracle:thin:@//database_host:port:sid`
	For a service connection type only, you have the option to customize and later test a different connection URL.

Create datasource after registration	Select to create the datasource on the application server after registration. **Note:** If you do not select the option, you must manually configure the data source.

d. For a service connection type, if you want to change the default URL, click the **Edit** button, specify the URL, and then click **OK**.

7. In IBM Db2 environments, configure connection properties for the database.

a. Specify the connection properties, and click **Next**.

The following table lists and describes the connection properties:

Property	Description
Database Display Name	Name for the Operational Reference Store that must appear in the Hub Console.
Machine Identifier	Prefix given to keys to uniquely identify records from the Hub Store instance.
Database server name	IP address or name of the server that hosts the IBM Db2 database.
Database name	Name of the database that you create.
Database hostname	IP address or name of the server that hosts the IBM Db2 database.
Schema Name	Name of the Operational Reference Store.
User name	User name for the Operational Reference Store. By default, this is the user name that you specify in the script that you use to create the Operational Reference Store. This user owns all the Operational Reference Store database objects in the Hub Store. If a proxy user is configured for the Operational Reference Store, then you can specify the proxy user instead.

Property	Description
Password	Password associated with the user name for the Operational Reference Store. For IBM Db2, the password is case sensitive. By default, this is the password that you specify when you create the Operational Reference Store. If a proxy user is configured for the Operational Reference Store, then you specify the password for the proxy user instead.
Dynamic Data Masking host	IP address or name of the server that hosts Dynamic Data Masking. Leave empty if you do not use Dynamic Data Masking.
DDM connection URL	Optional. URL for the Dynamic Data Masking server. Leave empty if you do not use Dynamic Data Masking.

Note: The **Schema Name** and the **User Name** are both the names of the Operational Reference Store that you specified when you created the Operational Reference Store. If you need this information, consult your database administrator.

The **Summary** page appears.

b. Review the summary, and specify additional connection properties.

The following table lists additional connection properties that you can configure:

Property	Description
Connection URL	Connect URL. The Connection Wizard generates the connect URL by default. The following example shows the format of the connect URL: `jdbc:db2://database_host:port/db_name`
Create datasource after registration	Select to create the datasource on the application server after registration. **Note:** If you do not select the option, you must manually configure the data source.

8. Click **Finish**.

   The **Registering Database** dialog box appears.

9. Click **OK**.

   The **Application Server Login** dialog box appears.

10. Enter the WebSphere administrative user name and password.
11. Click **OK**.

    The MDM Hub registers the ORS.

12. Restart the application server.
13. Select the Operational Reference Store that you registered, and click the **Test database connection** button to test the database settings.

    The Test Database dialog box displays the result of the database connection test.

14. Click **OK**

    The ORS is registered, and the connection to the database is tested.

Configure Class Loaders on WebSphere

After you run any of the PostInstallSetup scripts that are required, use the WebSphere deployment manager to configure class loaders for the Hub Server and Process Server applications.

1. Configure class loaders for the following applications: `siperian-mrm.ear`, `provisioning-ear.ear`, `entity360view-ear.ear`, and `siperian-mrm-cleanse.ear`.
   a. Select **Applications** > **Application Types** > **WebSphere enterprise applications**.
   b. On the **Enterprise Applications** page, click one of the applications.
   c. On the page for configuring applications, click the **Class loading and update detection** link.
   d. On the **Class loader** configuration page, select the **Classes loaded with local class loader first (parent last)** class loader order option.
   e. Click **Apply**, and then click **OK**.
2. Configure class loaders for the web modules of the following application EAR files:

Application EAR File	Web Module	Class Loader Order
`siperian-mrm.ear`	`zds-gui.war`	Classes loaded with local class loader first (parent last)
`provisioning-ear.ear`	`provisioning.war`	Classes loaded with local class loader first (parent last)
`siperian-mrm-cleanse.ear`	`siperian-mrm-cleanse.war`	Classes loaded with local class loader first (parent last)

   a. Select **Applications** > **Application Types** > **WebSphere enterprise applications**.
   b. On the **Enterprise Applications** page, click the name of the application EAR file.
   c. On the page for configuring the application, click the **Manage Modues** link.
   d. From the list of modules, click the link for the web module.
   e. On the web module configuration page, select the class loader order.
   f. Click **Apply**, and then click **OK**.
3. Restart WebSphere, and then start the Hub Server and Process Server applications.

Register the Operational Reference Stores

To register the Operational Reference Stores, use the Hub Console.

1. Start the **Databases** tool under the **Configuration** workbench.
2. Click **Write Lock** > **Acquire Lock**.
3. Click **Register database**.
   The **Informatica MDM Hub Connection Wizard** appears and prompts you to select the database type.
4. Select Microsoft SQL Server, Oracle, or IBM Db2, and click **Next**.

5. In Microsoft SQL Server, configure connection properties for the database.
   a. In the Connection Properties page, specify the connection properties, and then click **Next**.

      The following table lists and describes the connection properties:

Property	Description
Database Display Name	Name for the Operational Reference Store that must appear in the Hub Console.
Machine Identifier	Prefix given to keys to uniquely identify records from the Hub Store instance.
Database hostname	IP address or name of the server that hosts the Microsoft SQL Server database.
Port	Port of the Microsoft SQL Server database. The default is 1433.
Schema Name	Name of the Operational Reference Store.
Password	Password associated with the user name for the Operational Reference Store.
Dynamic Data Masking host	IP address or name of the server that hosts Dynamic Data Masking. Leave empty if you do not use Dynamic Data Masking.
DDM connection URL	Optional. URL for the Dynamic Data Masking server. Leave empty if you do not use Dynamic Data Masking.

   b. In the Connection Properties page, specify the connection properties, and then click **Next**.
   c. Review the summary, and specify additional connection properties.

      The following table lists additional connection properties that you can configure:

Property	Description
Connection URL	Connect URL. The Connection Wizard generates the connect URL by default.
Create datasource after registration	Select to create the datasource on the application server after registration.

6. In Oracle environments, configure connection properties for the database.
   a. Select an Oracle connection method, and click **Next**.

      The following table describes the Oracle connection methods that you can select:

Connection Method	Description
Service	Connect to Oracle by using the service name.
SID	Connect to Oracle by using the Oracle System ID.

      For more information about SERVICE and SID names, see the Oracle documentation.

      The **Connection Properties** page appears.

b. Specify the connection properties for the connection type that you select, and click **Next**.

The following table lists and describes the connection properties:

Property	Description
Database Display Name	Name for the Operational Reference Store that must appear in the Hub Console.
Machine Identifier	Prefix given to keys to uniquely identify records from the Hub Store instance.
Database hostname	IP address or name of the server that hosts the Oracle database.
SID	Oracle System Identifier that refers to the instance of the Oracle database running on the server. The **SID** field appears if you selected the **SID** connection type.
Service	Name of the Oracle SERVICE used to connect to the Oracle database. The **Service** field appears if the you selected the **Service** connection type.
Port	The TCP port of the Oracle listener running on the Oracle database server. The default is `1521`.
Oracle TNS Name	Name by which the database is known on your network as defined in the `TNSNAMES.ORA` file of the application server. For example: `mydatabase.mycompany.com`. You set the Oracle TNS name when you install the Oracle database. For more information about the Oracle TNS name, see the Oracle documentation.
Schema Name	Name of the Operational Reference Store.
User name	User name for the Operational Reference Store. By default, this is the user name that you specify in the script that you use to create the Operational Reference Store. This user owns all of the Operational Reference Store database objects in the Hub Store. If a proxy user is configured for the Operational Reference Store, then you can specify the proxy user instead.
Password	Password associated with the user name for the Operational Reference Store. For Oracle, the password is not case sensitive. By default, this is the password that you specify when you create the Operational Reference Store. If a proxy user is configured for the Operational Reference Store, then you specify the password for the proxy user instead.
Dynamic Data Masking host	IP address or name of the server that hosts Dynamic Data Masking. Leave empty if you do not use Dynamic Data Masking.
DDM connection URL	Optional. URL for the Dynamic Data Masking server. Leave empty if you do not use Dynamic Data Masking.

Note: The **Schema Name** and the **User Name** are both the names of the Operational Reference Store that you specified when you created the Operational Reference Store. If you need this information, consult your database administrator.

The **Summary** page appears.

c. Review the summary, and specify additional connection properties.

The following table lists additional connection properties that you can configure:

Property	Description

Connection URL	Connect URL. The Connection Wizard generates the connect URL by default. The following list shows the format of the connect URL for the Oracle connection types:
	Service connection type
	`jdbc:oracle:thin:@//database_host:port/service_name`
	SID connection type
	`jdbc:oracle:thin:@//database_host:port:sid`
	For a service connection type only, you have the option to customize and later test a different connection URL.

Create datasource after registration	Select to create the datasource on the application server after registration. **Note:** If you do not select the option, you must manually configure the data source.

d. For a service connection type, if you want to change the default URL, click the **Edit** button, specify the URL, and then click **OK**.

7. In IBM Db2 environments, configure connection properties for the database.

a. Specify the connection properties, and click **Next**.

The following table lists and describes the connection properties:

Property	Description
Database Display Name	Name for the Operational Reference Store that must appear in the Hub Console.
Machine Identifier	Prefix given to keys to uniquely identify records from the Hub Store instance.
Database server name	IP address or name of the server that hosts the IBM Db2 database.
Database name	Name of the database that you create.
Database hostname	IP address or name of the server that hosts the IBM Db2 database.
Schema Name	Name of the Operational Reference Store.

Property	Description
User name	User name for the Operational Reference Store. By default, this is the user name that you specify in the script that you use to create the Operational Reference Store. This user owns all the Operational Reference Store database objects in the Hub Store. If a proxy user is configured for the Operational Reference Store, then you can specify the proxy user instead.
Password	Password associated with the user name for the Operational Reference Store. For IBM Db2, the password is case sensitive. By default, this is the password that you specify when you create the Operational Reference Store. If a proxy user is configured for the Operational Reference Store, then you specify the password for the proxy user instead.
Dynamic Data Masking host	IP address or name of the server that hosts Dynamic Data Masking. Leave empty if you do not use Dynamic Data Masking.
DDM connection URL	Optional. URL for the Dynamic Data Masking server. Leave empty if you do not use Dynamic Data Masking.

Note: The **Schema Name** and the **User Name** are both the names of the Operational Reference Store that you specified when you created the Operational Reference Store. If you need this information, consult your database administrator.

The **Summary** page appears.

b. Review the summary, and specify additional connection properties.

The following table lists additional connection properties that you can configure:

Property	Description
Connection URL	Connect URL. The Connection Wizard generates the connect URL by default. The following example shows the format of the connect URL: `jdbc:db2://database_host:port/db_name`
Create datasource after registration	Select to create the datasource on the application server after registration. **Note:** If you do not select the option, you must manually configure the data source.

8. Click **Finish**.

   The **Registering Database** dialog box appears.

9. Click **OK**.

   The **Application Server Login** dialog box appears.

10. Enter the WebSphere administrative user name and password.
11. Click **OK**.

    The MDM Hub registers the ORS.

12. Restart the application server.
13. Select the Operational Reference Store that you registered, and click the **Test database connection** button to test the database settings.

The Test Database dialog box displays the result of the database connection test.

14. Click **OK**

    The ORS is registered, and the connection to the database is tested.

Validate the Upgraded Metadata

Ensure the Operational Reference Stores (ORS) do not have any validation errors. Compare the results with the previous validation results that you obtained in before the upgrade. Use the Repository Manager in the Hub Console to validate metadata.

Note: After you upgrade from a previous version of the MDM Hub, you might have validation errors for old databases that had staging tables without any mappings. In the Repository Manager tool in the Hub Console, click the **Repair** button to fix these repairable issues.

Validating Metadata

To validate the metadata of an Operational Reference Store (ORS), use the Repository Manager tool in the Hub Console.

1. From the **Configuration** workbench, click **Repository Manager**.
2. On the **Validate** tab, select the repository to validate.
3. Click **Validate**.

   The **Select Validation Checks** dialog box appears.
4. Select the validation checks to perform, and click **OK**.

   The Repository Manager tool validates the repository and displays any issues in the **Issues Found** pane.
5. To fix repairable issues, click **Repair**.
6. If you validate a Customer 360 or Supplier 360 repository, click **Restore** to resolve errors that appear on the **Domain Results** tab.
7. If the ORS remains in the **Unknown** state, synchronize the system clocks of the application server and the database machine.

Saving the Validation Results

After you run the validation process, you can save the validation results as an HTML file.

1. From the **Repository Manager** tool in the **Hub Console**, select the **Validate** tab.
2. Click the **Save** button.
3. From the **Save** dialog box, navigate to the directory where you want to save the validation results.
4. Specify a descriptive file name for the HTML file. Click **Save**.

   The Repository Manager saves the validation results as an HTML file in the specified location.

Resolving Metadata Validation Messages

After you run the validation tool, you might receive validation messages.

The following error messages are some of the most common validation messages.

`Warning SIP-PV-10703 Package 'EMPLOYEE_DETAILS_PKG' is not synchronized with its database view.`

To synchronize with the database view, run the Repair process from the Repository Manager.

`View 'C_EMPLOYEE_DETAILS_MTIP' - SELECT privilege for proxy user role is not granted.` **or** `SIP-MV-11410- SQL of the root MTIP is incorrect.`

Rebuild the MTIP views.

1. In the Hub Console, open the Configuration workbench and click **Enterprise Manager**.
2. Acquire a write lock.
3. Select the **ORS databases** tab.
4. Select the database.
5. Select the **Properties** tab.
6. Find the property called **MTIP regeneration required**, and click the **Regenerate MTIPs** button.

`View 'EMPLOYEE_DETAILS_PKG' - SELECT privilege for proxy user role is not granted.`

Update the proxy user role to include this privilege.

1. In the Hub Console, open the Security Access Manager workbench and click **Roles**.
2. Acquire a write lock.
3. Select the proxy user role.
4. Select the **Resource Privileges** tab.
5. Find the package or table that was named in the message.
6. Select the **Read** check box.

`SIP-PV-11105 - SELECT privilege for proxy user role has not been granted for a view.`

`View 'C_REPOS_USER_GROUP_ALL' - SELECT privilege for proxy user role is not granted.`

The database migration script created the proxy user role but did not grant privileges to the proxy user on the repository views. From the database, grant the proxy user SELECT privileges on the repository views.

Updating a Localized Schema

If the pre-upgrade schema includes localized lookup tables, you might see validation errors after you upgrade.

For lookup tables that were localized in version 10.2.x, the metatdata validation process generates the following types of errors:

`SIP-MV-22000 The name attribute for DB bundle [<lookup table name>.dbBundleMapping] is not defined in the configuration`

To fix the errors, perform the following steps:

1. Download the `dbBundleConfig.xml` file from the C_REPOS_CO_CS_CONFIG repository table.
2. In the `dbBundleConfig.xml` file, and copy the `name=<lookup table name>` parameter from the `bundle` element to the `mapping` element.

For example, the following snippet shows the edited `mapping` elements:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dbBundleConfiguration xmlns="http://www.example.com/mdm/db-bundle-configuration">
<bundle name="LUCountry" hubObject="C_LU_COUNTRY_LCL">
<mapping name="LUCountry" keyColumn="COUNTRY_CD" countryColumn="COUNTRY_CODE"
languageColumn="LANGUAGE_CODE" valueColumn="LOCALIZED_STRING"/>
</bundle>
<bundle name="LUState" hubObject="C_LU_STATE_LCL">
<mapping name="LUState" keyColumn="STATE_CD" countryColumn="COUNTRY_CODE"
languageColumn="LANGUAGE_CODE" valueColumn="LOCALIZED_STRING"/>
</bundle>
<bundle name="LUCountry.LUState" hubObject="C_LU_STATE_LCL">
<mapping name="LUCountry.LUState" keyColumn="STATE_CD" countryColumn="COUNTRY_CODE"
languageColumn="LANGUAGE_CODE" valueColumn="LOCALIZED_STRING"/>
</bundle>
</dbBundleConfiguration>
```

3. Upload the edited `dbBundleConfig.xml` file to the C_REPOS_CO_CS_CONFIG repository table.
4. Run the metadata validation.

Customize the Content Security Policy

Content Security Policy (CSP) is a standard that prevents code injection attacks, such as cross-site scripting. A website declares approved origins of content that a browser can load to display website content. The upgrade process checks for custom user interface components in the registered Operational Reference Store (ORS) databases. If any components are found, the appropriate content security policy is applied to ensure that no custom interface components are blocked.

If custom user interface components are found, the content security policy is set to the defaults required for Multidomain MDM to function. The upgrade process then adds the following rules: `script-src *; font-src *; style-src *; frame-src *; image-src *; default-src *;`

After you upgrade, customize the content security policy to secure your system and prevent code injection attacks. For more information about configuring the content security policy, see the *Multidomain MDM Provisioning Tool Guide*.

Perform Post-Upgrade Tasks for Clean Upgrade

After a clean upgrade, if a decryption error occurs when you launch the Hub Console, you must perform some post-upgrade tasks.

Perform the following post-upgrade tasks:

1. Encrypt passwords for schemas.
2. Update passwords for schemas.
3. Test and update the Operational Reference Store connections.
4. Test and update the ActiveVOS connection.
5. Test cleanse functions and add Process Servers.

Encrypt Passwords for Schemas

Encrypt the database schema passwords to secure them.

- To encrypt a database schema password, run the following command from a command prompt:

  ```
  java -classpath siperian-api.jar;siperian-common.jar;siperian-server.jar
  com.delos.util.PublicKeyBasedEncryptionHelper <plain text password> <Hub Server
  installation directory>
  ```

  The results are echoed to the terminal window:

  ```
  Plaintext Password: password
  Encrypted Password: encrypted password
  ```

Update Passwords for Schemas

You can update the MDM Hub Master Database password or Operational Reference Store password.

1. To update your Master Database password or Operational Reference Store password, connect as the cmx_system user and run the following statement:

   On Oracle and IBM Db2.

   ```
   UPDATE C_REPOS_DATABASE SET PASSWORD = '<new_password>' WHERE USER_NAME =
   <user_name>;
   COMMIT;
   ```

   On Microsoft SQL Server.

   ```
   UPDATE [dbo].[C_REPOS_DATABASE] SET PASSWORD = '<new_password>' WHERE USER_NAME =
   <user_name>
   ```

2. Restart the application server.

Test and Update the Operational Reference Store Connections

Test the connection to an Operational Reference Store. If the connection fails, edit the Operational Reference Store password in the Hub Console.

1. In the Configuration workbench, click **Databases**.
2. Acquire a write lock.
3. Select the Operational Reference Store that you want to test.
4. Click the **Test database connection** button.

   The **Test Database** dialog box appears.
5. Click **OK**.
6. If the connection test fails, click the **Edit database connection properties** icon.

   The **Register Database** dialog box appears for the selected Operational Reference Store.
7. Edit the password of the user name associated with the Operational Reference Store, and click **OK**.

Test and Update the ActiveVOS Connection

Test the connection of the ActiveVOS workflow engine to the MDM Hub. If the connection fails, edit the password of the ActiveVOS trusted user in the Hub Console.

1. In the Configuration workbench, click **Workflow Manager**.
2. Acquire a write lock.

3. Click the **Workflow Engines** tab, and select the ActiveVOS workflow engine.
4. To test the workflow engine connection, click **Test**.
5. If the workflow engine connection test fails, click **Edit**.
6. In the **Edit Workflow** dialog box, edit the password of the ActiveVOS trusted user, and click **OK**.

Test and Add Process Servers

Test a built-in cleanse function, such as a string function. If the test fails with a decryption error, delete and add the Process Server in the Hub Console.

1. Test a built-in cleanse function.
   a. In the Model workbench, click **Cleanse Functions**.
   b. Acquire a write lock.
   c. Select the cleanse function that you want to test.
   d. Click the **Test** tab.
   e. Enter an input value, and click **Test**.
2. If the test fails, check the `cmxserver.log` file for `SIP-09131: General Decryption failure` error.

   The `cmxserver.log` file is in the following directory:

   `<MDM Hub installation directory>/hub/server/logs`
3. If the error log contains the `SIP-09131` error, delete and add the Process Server.
   a. In the Utilities workbench, click **Process Server**.
   b. Acquire a write lock.
   c. Select the Process Server that you want to delete, and click the **Delete Process Server** icon.
   d. Confirm deletion, and click **OK**.
   e. Click the **Add Process Server** icon.
   f. In the **Add/Edit Process Server** dialog box, configure the Process Server properties.
   g. Save the configuration.

Configure Cleanse Functions for Platform Transformations

If you want to use the platform transformations that you configured, add an IDQ library in the Cleanse Functions tool. You can then use the cleanse functions in the library in place of the platform transformations. Multidomain MDM 10.5 includes an upgrade of Apache Axis2 to version 1.8.0. Update the custom code of cleanse functions so that all Apache Axis2 libraries point to version 1.8.0. Refresh existing SOAP cleanse functions by completing steps 1 to 3 below.

Note: Multidomain MDM 10.5 includes an upgrade of Apache Axis2 to version 1.8.0. Update the custom code of cleanse functions so that all Apache Axis2 libraries point to version 1.8.0 by refreshing existing SOAP cleanse functions.

1. Launch the Hub Console and start the **Cleanse Functions** tool.
2. Acquire a write lock.

3. Right-click **Cleanse Functions**, and then click **Add IDQ Library**.

   The **Add IDQ Library** dialog box appears.

4. Specify the following properties:

Property	Description
Library Name	Name of the IDQ library. The name appears as the folder name in the Cleanse Functions list.
IDQ WSDL URI	URI of the WSDL associated with the platform transformation.
IDQ WSDL Service	Service of the WSDL associated with the platform transformation.
IDQ WSDL Port	Port of the WSDL associated with the platform transformation.
Description	Descriptive text for the library that you want displayed in the Cleanse Functions tool.

5. Click **OK**.

   The IDQ library appears in the Cleanse Functions navigator.

6. Click **Refresh** to generate the IDQ library.

   The Cleanse Functions tool retrieves the WSDL associated with the platform transformation, generates the IDQ library, and displays the available cleanse functions in the Cleanse Functions list.

7. Test the cleanse functions.

   You can now use the cleanse functions in place of the platform transformations. The cleanse functions in the IDQ library can call the web services associated with the platform transformations.

Review the MDM Hub Environment Report

Use the Enterprise Manager tool in the Hub Console to review the current MDM Hub configuration for Hub Servers, Process Servers, the MDM Hub Master Database, and Operational Reference Store databases. Note the version history of the components.

Save a copy of the environment report in the `upgradedoc` upgrade documentation folder.

Saving the MDM Hub Environment Report

To save the MDM Hub environment report, use the Enterprise Manager tool in the Hub Console.

1. From the **Configuration** workbench in the Hub Console, select the **Enterprise Manager** tool.
2. From the **Enterprise Manager** tool, select the **Environment Report** tab.
3. Click **Save**.
4. From the **Save Hub Environment Report** dialog box, navigate to the directory where you want to save the environment report.
5. Click **Save**.

Upgrade External Calls and Applications

Effective in version 10.4, Multidomain MDM uses certificate-based authentication to authenticate external calls and applications. To use external calls and custom applications, you must configure a trusted application user. Also, EJB is not supported for external calls. You must use the HTTP communication protocol instead.

If your business entity services and custom applications use the BESExternalCall sample code and libraries included in the Resource Kit from a version prior to 10.4, perform the following upgrade steps:

1. Configure a trusted application user for the custom application.
2. Configure the following connection properties in the `bes-client.properties` file:

Connection Property	Description
siperian-client.protocol	Communication protocol that you want to use. Default is HTTP. Do not change the default value.
bes-client.http.url	URL for the custom application to connect to MDM. Use the following syntax: `http://<MDM host>:<port number>/cmx` Default is `http://localhost:8080/cmx`.

   The sample `bes-client.properties` file is in the following directory:
   `<Resource Kit installation directory>/samples/BESExternalCall/source/resources`

3. Review the CustomLogicService class in the `CustomLogicService.java` file in the BESExternalCall sample and implement the use of a trusted application user.

   The following code sample shows the CustomLogicService class:

```
public class CustomLogicService implements Provider<Source> {

    @Override
    public Source invoke(Source request) {

        CompositeServiceClient compositeServiceClient =
createCompositeServiceClient();
        CustomLogicFactory customLogicFactory = new
CustomLogicFactoryImpl(compositeServiceClient);
        String appName = "<trusted application user>";
        ExternalCallProcessor externalCallProcessor =
                new ExternalCallProcessor(compositeServiceClient, appName,
customLogicFactory);

        return externalCallProcessor.invoke(request);
    }

    private static CompositeServiceClient createCompositeServiceClient() {
        InputStream resourceAsStream =
CustomLogicService.class.getResourceAsStream("/bes-client.properties");
        Properties config = new Properties();
        try {
            config.load(resourceAsStream);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        return CompositeServiceClient.newCompositeServiceClient(config);
    }
```

4. Build the custom application to use the following updated JAR files:
   - siperian-api.jar
   - siperian-common.jar
   - mdm-spi.jar

   After you build the application, the external calls and applications use the HTTP communication protocol and certificate-based authentication.
5. Redeploy the custom application on the application server.

Upgrade the SiperianClient Library Classes for the EJB Protocol

If you use the EJB protocol to communicate with the MDM Hub through the Services Integration Framework (SIF) requests, you must use the latest version of the SiperianClient library classes. If you use custom JNDI lookup methods, update the lookup methods so that the methods conform to the EJB3 conventions.

1. Replace the existing SiperianClient library classes with the latest version of the SiperianClient library classes.

   The `siperian-api.jar` file located in the following directories contains the SiperianClient library classes:
   - `<Resource Kit Installation Directory>\sdk\sifsdk\lib`
   - `<MDM Hub Installation Directory>\hub\server\lib`
2. If you use custom JNDI lookup methods, update the lookup methods so that the methods conform to the EJB3 conventions.

Prepare the MDM Hub Metadata

Upgrade Task	Details
Regenerate match tokens.	Run the Generate Match Tokens batch job for each base object. The Generate Match Tokens batch job creates the match tokens based on the SSA-Name3 library files that you update during the Process Server upgrade.
Reindex the search data.	If the search data contains any accented characters, such as â and î, you can run the Initially Index Smart Search Data batch job to reindex the data. After you reindex the data, a search request can return records that contain accented characters.

Upgrade Task	Details
Configure metadata caching (Optional)	In version 10.1 and earlier, the MDM Hub used JBoss Cache for metadata caching. After you upgrade from one of these versions, the MDM Hub Server uses the Infinispan configuration file instead of the JBoss Cache configuration file. You might need to configure Infinispan caching to achieve similar results as JBoss Cache. For more information, see "Configuring Metadata Caching (Optional)" on page 179.
Reregister custom indexes.	You must reregister custom indexes after the migration. Use the registerCustomIndex SIF API to reregister the custom indexes. For more information about the RegisterCustomIndex SIF API, see the*Multidomain MDM Services Integration Framework Guide*. For SOAP and Java code samples to run the registerCustomIndex SIF API, see KB 500116.https://kb.informatica.com/howto/6/Pages/19/500116.aspx?myk=500116.

Upgrade Tests

Test the upgraded MDM Hub. Each Multidomain MDM implementation is unique and the testing requirements vary between the development, test, and production environments. If a suggested upgrade test is not appropriate for your environment, you can design your own tests. Design the test activities to meet the unique requirements of your implementation.

MDM Hub Upgrade Tests

Perform the following Hub Console upgrade tests that apply to your environment:

1. Launch the Hub Console.
2. Select the **Users** tool in the **Configuration** workbench to view the properties of an existing user.
3. Select the **Schema Viewer** tool in the **Model** workbench, and then connect to an Operational Reference Store. Review the schema in the **Schema Viewer**.
4. Select the **Schema** tool in the **Model** workbench to view the **Match/Merge Setup** for a base object.
5. Select the **Batch Viewer** tool in the **Utilities** workbench. If possible, run test batch jobs for the Stage batch job, the Load batch job, the Match batch job, and the Merge batch job.
6. Select the Process Server tool in the **Utilities** workbench. Test the connection to a registered Process Server.
7. Select the **Cleanse Functions** tool in the **Model** workbench. Run a test cleanse function for each external cleanse engine.
8. Select the **Data Manager** tool in the **Data Steward** workbench. Create two matching test records.
9. Select the **Merge Manager** tool in the **Data Steward** workbench. Find the two test records, merge the test records, and then unmerge the test records.

Custom Code Upgrade Tests

If you have custom code such as custom client applications, run tests to verify that the custom code works as expected.

Provisioning Tool Upgrade Test

Log in to the Provisioning tool. The tool validates the XML files that contain your configuration for business entities, reference entities, applications, custom views, tasks, and so on.

If the validation process is successful, continue to the next upgrade test. Optionally, you can spot check your configuration to verify the settings.

If the validation process detects some errors, review the list of errors and proposed fixes. A fix might include removing some settings. You can choose to accept all the fixes or cancel without making changes. If you choose to cancel, you must fix the errors in the XML files yourself. The XML files are stored in the C_REPOS_CO_CS_CONFIG and C_REPOS_COMPONENT_INSTANCE repository tables.

Caution: If you exit without fixing the errors, you might be locked out of the Provisioning tool.

1. Log in to the Provisioning tool.
2. Select an Operational Reference Store database.
3. If you see validation errors, review the proposed fixes.
   - To apply the fixes, click **Fix**.
   - To exit without applying the fixes, click **Cancel**.
     You are logged out of the Provisioning tool. Open the XML files and fix the errors.
4. Optionally, after the XML is valid, you can verify the configuration settings.

Data Director with Business Entities Upgrade Tests

If you use Data Director with business entities, open the application and test it.

Perform the following upgrade tests that apply to your environment:

1. Log in to Data Director.
2. Run multiple searches.
3. Create and process multiple tasks.
4. Insert a test record.
5. Copy the test record to create a second test record.
6. Run a search to find the two test records.
7. Merge and unmerge the two test records.

Data Director with Subject Areas Upgrade Tests

If you use Data Director with subject areas, you need to deploy the application before you begin the tests.

Perform the following upgrade tests that apply to your environment:

1. Launch the Data Director Configuration Manager, and then deploy a Data Director application instance.
2. Log in to Data Director.
3. Run multiple searches.

4. Create and process multiple tasks.
5. Insert a test record.
6. Copy the test record to create a second test record.
7. Run a search to find the two test records.
8. Merge and unmerge the two test records.

Configure General Hub Server Properties

The upgrade process preserves the values of the Hub Server properties. When you upgrade from earlier versions, the `cmxserver.properties` file does not contain the new properties and the changes to property defaults that were added for the release.

If you installed in a JBoss environment, change the value of the `cmx.jboss7.management.port` property from `9999` to `9990`.

Ensure that you configure the new properties that are added for the release. For more information, see the*Multidomain MDM Release Guide* and *Multidomain MDM Configuration Guide*.

Data Director and Hub Server Properties

The upgrade process preserves the values of the Hub Server properties that affect Data Director.

When you upgrade from earlier versions, the pre-upgrade `cmxserver.properties` file does not contain some properties that were added in version 10.0.0 and later. Verify that the upgrade process added the properties to the `cmxserver.properties` file. If necessary, add any missing properties to the end of the `<MDM Hub installation directory>/hub/server/resources/cmxserver.properties` file.

The default values for the properties retain the current behavior of Data Director applications. Before you customize the properties, read the property descriptions in the "Hub Server Properties" chapter of the *Multidomain MDM Configuration Guide*.

Properties Added in Version 10.3

Added new properties to support file upload, task manager, and workflow diagrams.

```
# File upload properties
# ----------------------
# Maximum upload size.
cmx.file.max_file_size_mb=20
# Maximum number of concurrent uploads.
cmx.file.max_concurrent_uploads=20
# Type of files that can be uploaded.
cmx.file.allowed_file_extensions=pdf,jpg
# Number of minutes until an uploaded file expires.
# To avoid expiration, set to 0.
cmx.server.attachment.temp.ttl_minutes=60

# Task Manager property
# ---------------------
# Set to true to display the Task Manager tab in applications
# that use subject areas.
cmx.dataview.taskmanager.enabled=true
```

```
# Workflow diagram properties
# ----------------------------
# Set to true to display the workflow diagram associated
# with the tasks in the Task Manager for the users with
# the ActiveVOS abAdmin role.
cmx.e360.BPMProcess.view.enabled=false
cmx.e360.BPMProcess.view.autologout.seconds=30
```

Note: If you decide to use Elasticsearch for full-text search in a Data Director application, add the Elasticsearch properties manually. For more information, see "Search Configuration Upgrade Overview" on page 129.

Properties Added in Version 10.2 and earlier

Added properties to support the Data tab, search, and the Entity 360 framework. Add the default values and then change them as needed.

```
# View properties
# ----------------
# Show or hide the views for subject areas.
cmx.dataview.enabled=true
# Show or hide the views for business entities.
cmx.e360.view.enabled=false
# Show or hide the Cross-reference view and Match view.
cmx.e360.match_xref.view.enabled=false

# Search with Solr (formerly Smart Search) properties
# ---------------------------------------------------
# Set to true to use Solr for search.
cmx.ss.enabled=false
```

Data Director Global Properties

If your Data Director environment included user preferences for shown or hidden columns, the settings are lost when you upgrade because the cryptographic hash algorithm has changed in this version. After you upgrade, clear the C_REPOS_DS_PREF_DETAIL table and recreate your user preferences.

For instructions about how to update Data Director global properties, including user preferences, see the *Multidomain MDM Data Director Implementation Guide*.

Generate the Business Entity Schema

If you use business entity services, you must generate the business entity schema after you upgrade. You can generate the business entity schema, use the Informatica Data Director Configuration Manager.

Back up and customized business entity or business entity service configuration files before you generate the business entity schema.

To generate the business entity schema from the Informatica Data Director Configuration Manager, in the Applications screen, click **Generate Business Entity Schema**.

CHAPTER 9

Search Configuration Upgrade

This chapter includes the following topics:

Search Configuration Upgrade Overview

You can use a Data Director application or a custom application to search for data within a specific business entity. Previously, you configured Solr for search operations, which is now not supported. You must configure Elasticsearch for search operations.

To upgrade the search configuration to use Elasticsearch, perform the following tasks:

1. Install and set up Elasticsearch.
2. Configure the MDM Hub properties for search.
3. Configure search by using the Provisioning tool.
4. Validate the Operational Reference Store (ORS).
5. Index the search data.

Step 1. Install and Set Up Elasticsearch

To configure search, you must install and set up Elasticsearch.

To set up Elasticsearch, perform the following tasks:

1. Complete the pre-installation tasks.
2. Install Elasticsearch.
3. Configure the Elasticsearch Java Virtual Machine (JVM).

4. Configure the Elasticsearch properties file.
5. Secure Elasticsearch.
6. Install the analysis plugins.
7. Configure stop words, synonyms, and character mappings
8. Start Elasticsearch.

Complete Pre-Installation Tasks

Before you install and set up Elasticsearch clusters, prepare the environment and determine whether you want to configure high availability.

Tasks for All Environments

Perform the following tasks to prepare the installation environment:

- Ensure that each machine satisfies the hardware requirements for the supported version of Elasticsearch. For information about hardware, see the Elasticsearch documentation.
- Ensure that each machine satisfies the software requirements for the supported version of Elasticsearch, such as supported operating systems and Java version. For information about the software requirements, see the *Elasticsearch Support Matrix*.
- Complete important system configurations, such as swapping, file descriptors, and virtual memory. For information about important system configurations, see the Elasticsearch documentation.

Tasks for UNIX Environments

In a UNIX environment, perform the following tasks:

- To avoid data loss due to insufficient number of file descriptors, set the number of file descriptors to 65536 or higher.
- To prevent memory swapping, configure the system to prevent swapping. You can configure the Java Virtual Machine (JVM) to lock the heap in memory through `mlockall`.

High Availability Requirements

If you have a large amount of data to index and search, the best practice is to implement a highly available Elasticsearch cluster. A highly available cluster has multiple nodes, and the cluster can distribute the workload among the nodes. If one node fails in a production environment, the cluster distributes the workload to the other nodes.

As a pre-installation task, decide if you want to implement a highly available Elasticsearch cluster. If so, configure the Elasticsearch cluster as usual, but ensure that you satisfy the following additional requirements:

- The Elasticsearch cluster has three or more nodes.

  Tip: You can set up a small cluster to start and scale it as necessary. Analyze the workload and make sure that you have enough capacity to handle a node failure.
- Each node is configured on a separate, dedicated machine.
- At least three of the nodes are master nodes to ensure stability and performance. Note that Elasticsearch recommends an odd number of master nodes.
  - If the cluster has only three nodes, configure all the nodes as master nodes.
  - If the cluster has more than three nodes, configure three nodes as master nodes and configure the rest of the nodes as data nodes.

- Based on the Elasticsearch cluster size, decide on the number of replicas. When you use the Provisioning tool to configure the Elasticsearch index, you can specify the number of replicas to use.
- For each node, set the following additional properties in the `elasticsearch.yml` configuration file:
  - discovery.zen.minimum_master_nodes
  - discovery.zen.ping.unicast.hosts

For more information about highly available clusters, including hardware requirements, system configurations, and property values, see the Elasticsearch documentation.

Install Elasticsearch

After you install the Hub Server and the Process Server, to configure search, install and set up Elasticsearch clusters.

Ensure that you use a supported operating system and Java version for your Elasticsearch installation. For more information, see the Elasticsearch Support Matrix.

For more information about how to install Elasticsearch and set up clusters, see the Elasticsearch documentation.

1. From the Elastic website, download the supported version of the Elasticsearch archive file.

   For information about the supported versions, see the Product Availability Matrix (PAM). You can access PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.
2. Extract the Elasticsearch archive file.

Configure the Elasticsearch Java Virtual Machine (JVM)

Configure the Elasticsearch Java Virtual Machine (JVM) to use a heap size based on the amount of RAM available on your machine. To configure the JVM, edit the `jvm.options` file.

1. Find the `jvm.options` file in the following directory:

   `<elasticsearch installation directory>/config`
2. Use a text editor to open the file, and edit the following properties:

Property	Description
-Xms	Minimum heap size. Default is `1 GB`.
-Xmx	Maximum heap size. Default is `1 GB`.
-XX:HeapDumpPath	Heap dump path. Default is `/var/lib/elasticsearch`. In a multi-cluster environment, you must set this property to an alternative path.

Note: Set the minimum heap size (Xms) and the maximum heap size (Xmx) to the same value. Use the default settings for other properties.

Configure the Elasticsearch Properties File

Informatica provides a sample Elasticsearch properties file. To configure Elasticsearch, edit the properties file.

1. Find the `elasticsearch.yml` file in the following directory:

   `<elasticsearch installation directory>/config`

2. Use a text editor to open the file, and edit the following properties:

Property	Description
bootstrap.memory_lock	Sets up memory locking. To prevent any Elasticsearch memory from being swapped out, set to `true`. Default is `true`.
cluster.name	Specify a unique name for the Elasticsearch cluster. If you have multiple clusters, ensure that the name of each cluster is unique. If a cluster has multiple nodes, ensure that on each node of the cluster, the same cluster name is specified.
discovery.zen.minimum_master_nodes	Required for a multi-node cluster to prevent data loss and maintain cluster stability. Set to the following value: `(number of master-eligible nodes / 2) + 1` For example, if a cluster has three nodes, all of which are master-eligible nodes and can contain data, then set the property to `(3 / 2) + 1`, which is rounded to `2`.
discovery.zen.ping.unicast.hosts	Required for a multi-node cluster. This property is used to specify the discovery setting, which is a list of IP addresses and transport ports of the nodes in the cluster. Use the following format to set the property: `["host1:port1","host2:port2","host3:port3"]`
http.port	Port for the HTTP requests. Default is `9200`.
network.host	The IP address of the host to use as the bind address.
node.data	Enables a node as a data node that performs data related operations, such as CRUD and search. Default is `true`.
node.ingest	Enables a node as an ingest node that transforms and enriches the data before indexing. Default is `true`.
node.master	Enables a node as a master node that controls the cluster. If a cluster has multiple nodes, enable at least one of the nodes as a master node. For high availability, set multiple nodes as master nodes. Default is `true`.
node.name	Specify a unique name for the node.
path.data	Path to the directory where you want to store the data. You can configure multiple data directories. For more information about configuring multiple data directories, see the Elasticsearch documentation.

Property	Description
path.logs	Path to the log files.
transport.tcp.port	The TCP bind port. Default is `9300`.

3. Save the properties file with the same name, `elasticsearch.yml`.

Secure Elasticsearch (Optional)

After you install Elasticsearch, secure the communication between the MDM Hub and Elasticsearch. Also, secure the Elasticsearch cluster.

For information about securing Elasticsearch, see the Elasticsearch security documentation.

Install Analysis Plugins

Install the Phonetic and Japanese (kuromji) analysis plugins, which extend Elasticsearch by adding new analyzers, tokenizers, token filters, and character filters. The Phonetic analysis plugin analyzes and converts tokens into their phonetic equivalent. The Japanese (kuromoji) analysis plugin analyzes Japanese by using the Kuromji analyzer.

1. Download the Phonetic and Japanese (kuromji) analysis plugin from the Elastic website.
2. Install the Phonetic analysis plugin on each cluster node by running the following command:

   ```
   sudo bin/elasticsearch-plugin install analysis-phonetic
   ```

3. Install the Japanese (kuromji) analysis plugin on each cluster node by running the following command:

   ```
   sudo bin/elasticsearch-plugin install analysis-kuromoji
   ```

4. Restart each cluster node.

Configure Stop Words, Synonyms, and Character Mappings

When you perform a search, MDM can ignore common words such as "and", "an", and "is". MDM can also search for synonyms of the search string. For example, when you search for "William", the search result can include the synonyms "Will" and "Willy".

To configure common words to ignore or include synonyms in search results, Informatica provides text files that contain stop words and synonyms, or you can configure your own.

To use the default Elasticsearch analyzers for languages such as Chinese, Japanese, and Korean, Informatica provides a mappings file, `mapping-FoldToASCII.txt`. The character filter of these default analyzers uses the mappings file to convert alphabetic, numeric, and symbolic characters that are not in the Basic Latin Unicode block to their ASCII equivalent.

To get the `stopwords.txt`, `synonyms.txt`, `stopwords_ja.txt`, and `mapping-FoldToASCII.txt` files, contact Informatica Global Customer Support.

To configure stop words, synonyms, and character mappings, perform the following steps:

1. Create an `analysis` directory in the following location:

   ```
   <elasticsearch installation directory>/config
   ```

2. Copy the `stopwords.txt` and `synonyms.txt` files to the `analysis` directory.

3. To configure stop words for languages such as Japanese, create a `lang` directory in the following location:

   `<elasticsearch installation directory>/config/analysis`

4. Copy the stop words files for other languages, such as `stopwords_ja.txt`, and the `mapping-FoldToASCII.txt` file to the `lang` directory.

Start Elasticsearch

After you set up Elasticsearch, start each node of the Elasticsearch cluster for the changes to take effect.

Tip: When you start Elasticsearch, if memory locking issues occur, you might need to set `soft memlock unlimited` and `hard memlock unlimited`.

1. Open a command prompt, and change to the following directory:

   `<elasticsearch installation directory>/bin`

2. Run the following command:

   On UNIX. `elasticsearch.sh`

   On Windows. `elasticsearch.bat`

Step 2. Configure the MDM Hub Properties for Search

To configure the MDM Hub properties, use the Hub Console, the Process Server properties file, and the Hub Server properties file.

1. Configure the Process Server properties.
2. Configure the Hub Server properties.

Configure the Hub Server for Search

You must configure all the Hub Server instances to enable search. Use the Hub Server tool in the Hub Console and the `cmxserver.properties` file to configure the Hub Server properties for search.

1. Use a text editor to open the `cmxserver.properties` file in the following location: `<MDM Hub Installation Directory>\hub\server\resources\cmxserver.properties`

2. Configure the following properties for search:

cmx.ss.engine

Required if you want to use the Elasticsearch engine for search. Manually add the property and set to `es`.

ex.max.conn.per.host

Sets the maximum number of Elasticsearch nodes that you want to connect to the host. Set to the number of Elasticsearch cluster nodes on the host.

ex.max.threads

Sets the maximum number of threads that you want the Apache asynchronous non-blocking receiver to use for each node in the Elasticsearch cluster. Default is `1`.
Change the value only when suggested by Informatica Global Customer Support.

es.index.refresh.interval

Sets the interval, in seconds, for Elasticsearch to commit the changes to the data after an Initially Index Smart Search Data batch job is run. The data is available for search after this time interval. Default is `30`.
This property impacts the high indexing volume encountered during initial indexing. Change the value only when suggested by Informatica Global Customer Support.

ssl.keyStore

Required if you use the HTTPS port of the application server to configure the Hub Server. Manually add the property. Absolute path and file name of the keystore file.

ssl.keyStore.password

Required if you use the HTTPS port of the application server to configure the Hub Server. Manually add the property. Plain text password for the keystore file.

ssl.trustStore

Required if you use the HTTPS port of the application server to configure the Hub Server. Manually add the property. Absolute path and file name of the truststore file.

ssl.trustStore.password

Required if you use the HTTPS port of the application server to configure the Hub Server. Manually add the property. Plain text password for the truststore file.

After you update the Hub Server properties, you must validate the Operational Reference Store (ORS), and restart the Hub Console.

Configure Process Servers for Search

When you configure search with Elasticsearch, enable search on all the Process Server instances. Use the Process Server tool in the Hub Console and the `cmxcleanse.properties` file to configure the Process Server properties for search.

1. In the Hub Console of a node, start the Process Server tool.
2. Click **Write Lock** > **Acquire Lock**.
3. In the right pane of the Process Server tool, click the **Add Process Server** button.

   The **Add/Edit Process Server** dialog box appears.
4. To enable search, select the **Enable Search Processing** property of the Process Server.
5. Click **OK**, and then click **Save**.

6. Edit the `cmxcleanse.properties` file.

   The `cmxcleanse.properties` file in the following location:`<MDM Hub Installation Directory>\hub \cleanse\resources`

   a. Configure the following properties for search:

      cmx.ss.engine

      Required if you want to use the Elasticsearch engine for search. Manually add the property and set to `es`.

      ex.max.conn.per.host

      Sets the maximum number of Elasticsearch nodes that you want to connect to the host. Set to the number of Elasticsearch cluster nodes on the host.

      ex.max.threads

      Sets the maximum number of threads that you want the Apache asynchronous non-blocking receiver to use for each node in the Elasticsearch cluster. Default is `1`.
      Change the value only when suggested by Informatica Global Customer Support.

   b. Remove the following properties that are configured for search with Solr:

      - solr.allowAdminConsole
      - zookeeper.tickTime
      - pingSolrOnStartup

7. Save the `cmxcleanse.properties` file.
8. Restart the application server.

Step 3. Configure Search by Using the Provisioning Tool

After you set up Elasticsearch and configure the MDM Hub properties, use the Provisioning tool to configure the search environment.

1. Configure the Elasticsearch cluster.
2. Configure the search result views.

Configure the Elasticsearch Cluster

Use the Provisioning tool to configure the Elasticsearch cluster for the MDM applications. The search APIs use the configuration. The Data Director application and any custom applications use the search APIs.

Note: When you configure the Elasticsearch cluster, only master nodes of the cluster must be specified.

1. Open a supported browser, and enter the following URL:

   `https://<MDM Hub Server host name>:<MDM Hub Server port number>/provisioning/`

   The **Login** page appears.
2. Enter the user name and password, and click **Log In**.
3. From the **Database** list, select the database for which you want to configure the Elasticsearch cluster.
4. Click **Configuration** > **Infrastructure Settings**.

The **Infrastructure Settings** page appears.

5. Select **Elasticsearch Cluster** from the list, and then click **ESCluster**.

   The **ESCluster** appears in the tree view panel.

6. To configure an Elasticsearch cluster node, select **esNode** in the tree view panel, and then click **Create**.
7. Specify the following properties of the configured Elasticsearch cluster:

Property	Description
Name	Name of the master node in the Elasticsearch cluster.
URL	URL of the master node in the Elasticsearch cluster. The URL format is `https://<host name>:<port>`.

8. Click **Apply**.
9. If you want to create additional master nodes, repeat steps 6 through 8.
10. Publish the changes to the MDM Hub.
    a. Click **Publish**.

       A confirmation dialog box appears that prompts you to publish or review the changes.

    b. Review the changes or publish without a review.
       - To publish without a review, click **Publish**.
       - To publish after a review, click **Review Changes** and follow the instructions that appear on the screen.

Configure the Search or Query Results Display

You can use the Provisioning tool to configure the business entity views that you want to use for search. A search result includes only the fields that are part of the business entity view that you configure for search results. You can also configure the order in which the search filters appear.

Before you configure the searchable views, create the business entity views that you want to use for the search results.

Note: To display child record fields of a business entity in the search results, use a business entity view that is transformed from a business entity. Ensure that the view includes child record fields at the root record level.

1. Log in to the Provisioning tool.
2. From the **Database** list, select the database with which your application is associated.
3. Click **Configuration** > **Application Editor**.

   The **Applications** page appears.

4. From the **Applications** list, select the application for which you want to configure search.

   If you do not have an application, create one before you can configure search.

5. In the tree view panel, select **Search Configuration**, and click **Create**.
6. In the properties panel, select a business entity and the business entity view that you want to use to display the search or query results.

   If you do not select a business entity view, the search and query results contain all the business entity fields.

7. Optionally, if you configured search, select the filters and configure the display order of the search filters.
   a. Click the **Edit** icon next to **Filter Display Order**.

      The **Edit Filter Display Order** dialog box appears. The dialog box contain filters, which are fields that are configured as filterable in the business entity model.
   b. Drag the filters from the **Available Filters** section to the **Selected Filters** section.
   c. To configure the order, drag and move the filters up or down.
   d. Click **OK**.
8. Click **Apply**.

   The search configuration is saved to the temporary workspace.
9. Publish the changes to the MDM Hub.
   a. Click **Publish**.

      A confirmation dialog box appears that prompts you to publish or review the changes.
   b. Review the changes or publish without a review.
      - To publish without a review, click **Publish**.
      - To publish after a review, click **Review Changes** and follow the instructions that appear on the screen.

Step 4. Validate the Operational Reference Store

To validate the metadata of the Operational Reference Store (ORS) that is affected by the Elasticsearch configuration, use the Repository Manager tool in the Hub Console.

1. Start the Hub Console and connect to the MDM Hub Master Database.
2. Expand the **Configuration** workbench, and click **Repository Manager**.

   The Repository Manager appears.
3. Click the **Validate** tab, and select the repository to validate.
4. Click **Validate**.

   The **Select Validation Checks** dialog box appears.
5. Select the validation checks to perform.
6. Click **OK**.

   The Repository Manager validates the repository and displays any issues in the **Issues Found** pane.
7. To repair issues, click **Repair**.

Step 5. Index the Search Data

If your environment contains data, manually run the Initially Index Smart Search Data batch job to index the data. If your environment does not contain any data, you do not need to run the Initially Index Smart Search Data job. When you run the Load batch job to load data, the Load batch job automatically runs the Initially

Index Smart Search Data batch job and indexes the data. A search request uses the indexes to search for records.

Run the Initially Index Smart Search Data batch job on all the base objects that contribute to the business entities. When you run the Initially Index Smart Search Data batch job on a base object, the Elasticsearch server indexes the data in the searchable fields. The job then adds the indexed data to all the collections that represent the business entities to which the searchable fields belong. If a collection is too large, you can split the collection into one or more shards. Shards are the logical pieces of a collection split over multiple nodes. When you perform a search, the Elasticsearch server reads the collections and returns the matching fields.

The Initially Index Smart Search Data batch job indexes the records asynchronously and reports successful completion after the job queues the indexing request for all the records. A search request can show the indexed records only after the successful completion of the index request, which might take a few minutes.

Important: If you update the searchable properties of a field after you index your data, the indexes are deleted. You must run the Initially Index Smart Search Data batch job to index the data. In addition, the indexing process is a resource-intensive process, so do not run multiple Initially Index Smart Search Data batch jobs in parallel.

Upgrading to Elasticsearch Version 7.17.0 (Optional)

If your existing MDM environment uses Elasticsearch version 6.2.3 or 6.8.6 with Search Guard, Informatica recommends that you upgrade to Elasticsearch version 7.17.0 with X-Pack community edition. Elasticsearch version 7.17.0 is backward compatible.

Prerequisites for Upgrading to Elasticsearch Version 7.17.0

Before you upgrade to Elasticsearch version 7.17.0, you must ensure that your existing Elasticsearch version 6.2.3 or 6.8.6 contains the following properties in the `elasticsearch.yml` file:

- `cluster.name:<user defined value>`
- `node.name:<user defined value>`

Note: You must migrate these properties from your existing Elasticsearch version configuration directory to the Elasticsearch 7.17.0 configuration directory. Execute the migration steps in the exact order listed.

If your existing Elasticsearch version does not contain these configuration properties, do not upgrade to Elasticsearch version 7.17.0 as your existing Elasticsearch version is not compatible.

Configuring the Elasticsearch Properties File

To configure Elasticsearch 7.17.0, edit the properties file.

1. Go to the following directory:

   `<elasticsearch 7.17.0 installation directory>/config`

2. In a text editor, open the `elasticsearch.yml` file and edit the following properties:

Property	Description
bootstrap.memory_lock	To lock the memory, set to `true`. Default is `true`.
cluster.name	A unique name for the Elasticsearch cluster. If you have multiple clusters, ensure that the name of each cluster is unique. If a cluster has multiple nodes, ensure that on each node of the cluster, the same cluster name is specified.
http.port	Port for the HTTP requests. Ensure the HTTP port for Elasticsearch 7.17.0 is enabled and the port number does not conflict with the Elasticsearch version 6.2.3 or 6.8.6 port number. Default is `9200`.
network.host	The IP address of the host to use as the bind address.
node.data	Indicates whether to use a node as a data node that performs data related operations, such as CRUD and search. set to `true`. Default is `true`.
node.ingest	Indicates whether to use a node as an ingest node that transforms and enriches the data before indexing. set to `true`. Default is `true`.
node.master	Indicates whether to use a node as a master node that controls the cluster. If a cluster has multiple nodes, enable at least one of the nodes as a master node. For high availability, set multiple nodes as master nodes. set to `true`. Default is `true`.
node.name	Specify a unique name for the node.
transport.tcp.port	The TCP bind port. Ensure the TCP bind port for Elasticsearch 7.17.0 is enabled and the port number does not conflict with the Elasticsearch version 6.2.3 or 6.8.6 port number. Default is `9300`.

3. Save the `elasticsearch.yml` file.
4. Install the analysis plugins. For more information about how to install the analysis plugins, see the "Install Analysis Plugins" on page 133 topic.

Upgrading Elasticsearch Indexes

Upgrade the existing Elasticsearch version 6.2.3 or 6.8.6 index properties to Elasticsearch version 7.17.0 index properties. To upgrade the indexes, edit the Elasticsearch configuration properties file.

1. Go to the following directory:

   `<elasticsearch<6.2.3 or 6.8.6> installation directory>/config`

2. Copy the `analysis` folder.
3. Paste the `analysis` folder in the following directory:

   `<elasticsearch 7.17.0 installation directory>/config`

4. In the Elasticsearch version 7.17.0 configuration directory, edit the following properties in the `elasticsearch.yml` file:

5. Use a text editor to open the file, and edit the following properties:

Property	Description
xpack.security.transport.ssl.enabled	Specifies whether you want to enable transport layer security (TLS) on the REST layer. Set to `true`. Default is `true`.
xpack.security.transport.ssl.verification_mode	Mode to verify the certificate. The supported mode is certificate. Enter the value `certificate`.
xpack.security.transport.ssl.keystore.path	Relative path to the keystore file. If you used the `sip_ant` script to create the keystore, use the default path. Default is `MDM_ESKEYSTORE_FILE_JKS.keystore`.
xpack.security.transport.ssl.truststore.path	Relative path to the truststore file. If you used the `sip_ant` script to create the truststore, use the default path. Default is `MDM_TRUST_JKS.keystore`.
xpack.security.transport.ssl.keystore.password	Password for keystore. Default is `infa@es`.
xpack.security.transport.ssl.truststore.password	Password for truststore. Default is `infa@es`.
xpack.security.http.ssl.enabled	Specifies whether you want to enable X-Pack. Set to `true`.
xpack.security.http.ssl.verification_mode	Mode to verify the certificate. The supported mode is certificate. Enter the value `certificate`.
xpack.security.http.ssl.client_authentication	Specifies whether you want to use the TLS client authentication mode on the REST layer. Set to `REQUIRED`.
xpack.security.http.ssl.keystore.path	Relative path to the keystore file. If you used the `sip_ant` script to create the keystore, use the default path. Default is `MDM_ESCLIENT_FILE_JKS.keystore`.
xpack.security.http.ssl.truststore.path	Relative path to the truststore file. If you used the `sip_ant` script to create the truststore, use the default path. Default is `MDM_TRUST_JKS.keystore`.
xpack.security.http.ssl.truststore.password	Password for truststore. Default is `infa@es`.
xpack.security.http.ssl.keystore.password	Password for keystore. Default is `infa@es`.
xpack.security.authc.token.enabled	Specifies whether you want to enable authentication token. Ensure that the value is always set to `false`
xpack.security.authc.api_key.enabled	Specifies whether you want to enable API key token. Ensure that the value is always set to `false`

6. In the Elasticsearch version 7.17.0 configuration directory, open the `roles.yml` file and enter the following code to define the user role:

```
user:
   cluster: [ 'monitoring_user','transport_client','kibana_user','snapshot_user' ]
   indices:
     - names: [ '*' ]
       privileges: [ 'all' ]
```

7. In the Elasticsearch version 7.17.0 configuration directory, open the `elasticsearch.yml` file and enter the following code:

```
xpack.security.authc:
              anonymous:
                   username:
                   roles:
                   authz_exception:
```

8. Start Elasticsearch. For more information about how to start Elasticsearch, see the "Start Elasticsearch" on page 134 topic.
9. Run the Elasticsearch health APIs to ensure that the Elasticsearch server uses the HTTPS protocol. For more information about how to test the Elasticsearch APIs, see the Elasticsearch documentation.
10. Stop the existing Elasticsearch version 6.2.3 or 6.8.6 environments, the MDM Hub server and Cleanse server services.
11. Go to the following directory:

    `<elasticsearch 7.17.0 installation directory>/config`
12. Configure the following properties in the `elasticsearch.yml` file:

Property	Description
path.data	Point to the existing version of Elasticsearch data directory if you want to migrate existing indexes. You can configure multiple data directories. For more information about configuring multiple data directories, see the Elasticsearch documentation. **Note:** To migrate to Elasticsearch version 7.17.0, your existing Elasticsearch version 6.2.3 or 6.8.6 configuration properties must contain the `cluster.name` and `node.name` properties. If these properties do not exist, you cannot migrate your existing indexes.
path.logs	Path to the log files.
http.port	Port for the HTTP requests. Enter the same port number that you set in the existing Elasticsearch version environment.
transport.tcp.port	The TCP bind port. Enter the same TCP bind port number that you set in the existing Elasticsearch version environment.

13. Start the Elasticsearch version 7.17.0 environment.
14. Test the Elasticsearch APIs to ensure the indexes are open and do not display any errors. For more information about how to test Elasticsearch APIs, see the Elasticsearch documentation.
15. Start the Hub Server, the Process Server and the Elasticsearch services.

    Your existing MDM environment is configured with Elasticsearch 7.17.0

CHAPTER 10

Hierarchies Upgrade

This chapter includes the following topics:

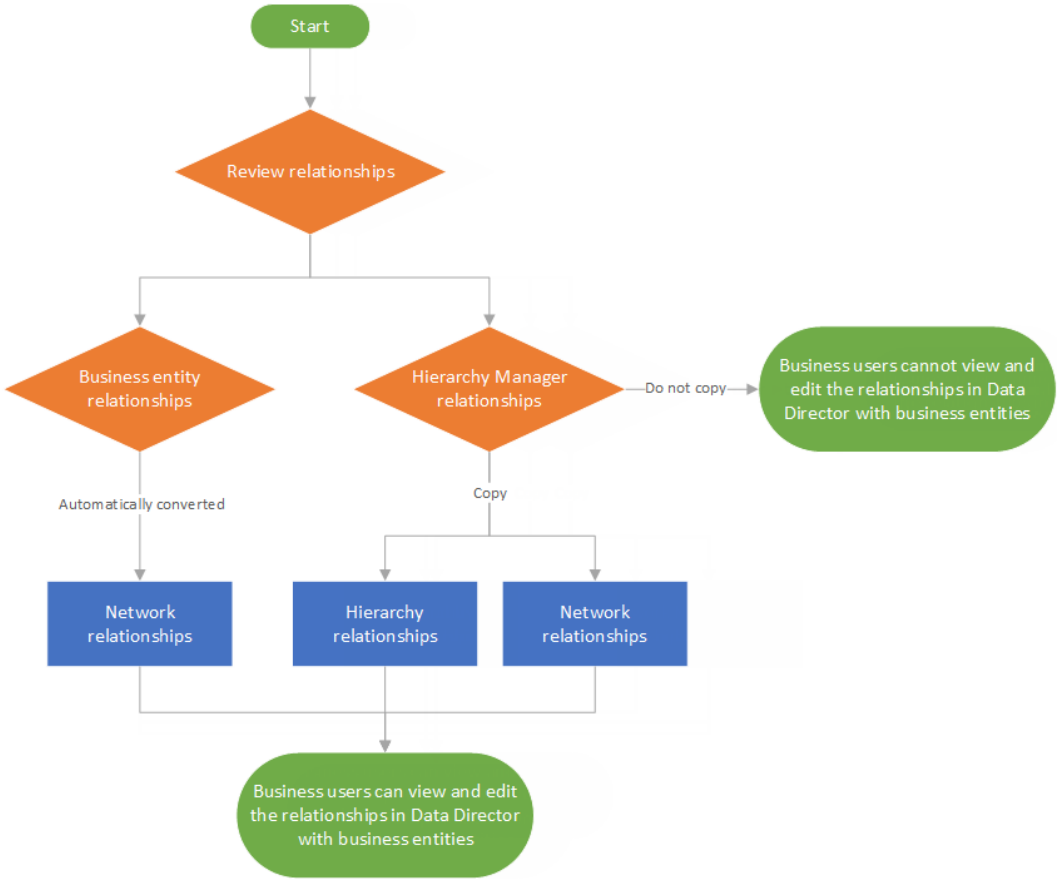
Hierarchies Upgrade Overview

Use the Provisioning tool to create hierarchy relationships and network relationships between business entities. Then in Data Director, business users can use the **Hierarchy** view and **Network** view to create and manage relationships between records.

If you are upgrading from a version prior to 10.4, you might have business entity relationships in the Provisioning tool or hierarchy relationships in Hierarchy Manager. When you upgrade to 10.4, the Provisioning tool automatically converts the business entity relationships to network relationships.

If you have relationships in Hierarchy Manager, review the relationships and determine which relationships you want to continue to use. Then based on your requirements, determine which relationships you want to copy and create as hierarchy relationships or network relationships. Use the Provisioning tool to copy the relationships in Hierarchy Manager and create them as hierarchy relationships or network relationships.

The following diagram walks you through the steps:



Understand Hierarchy Relationships and Network Relationships

Use the Provisioning tool to copy relationships in Hierarchy Manager and create them as hierarchy relationships or network relationships.

If you want to define hierarchical relationships between business entities, then copy the relationships and create them as hierarchy relationships. If you want to create a connection between related business entities, then copy the relationships and create them as network relationships.

When you copy relationships in the Hierarchy Manager, the new relationship that you create references the same relationship base object in the Hub Console. After you finish copying the relationships that you want to use, you must revert the relationship base object to a base object. Reverting a relationship base object to a base object removes the Hierarchy Manager metadata from the object and allows you to create and update hierarchy and network relationships in the Provisioning tool.

Hierarchy Relationships

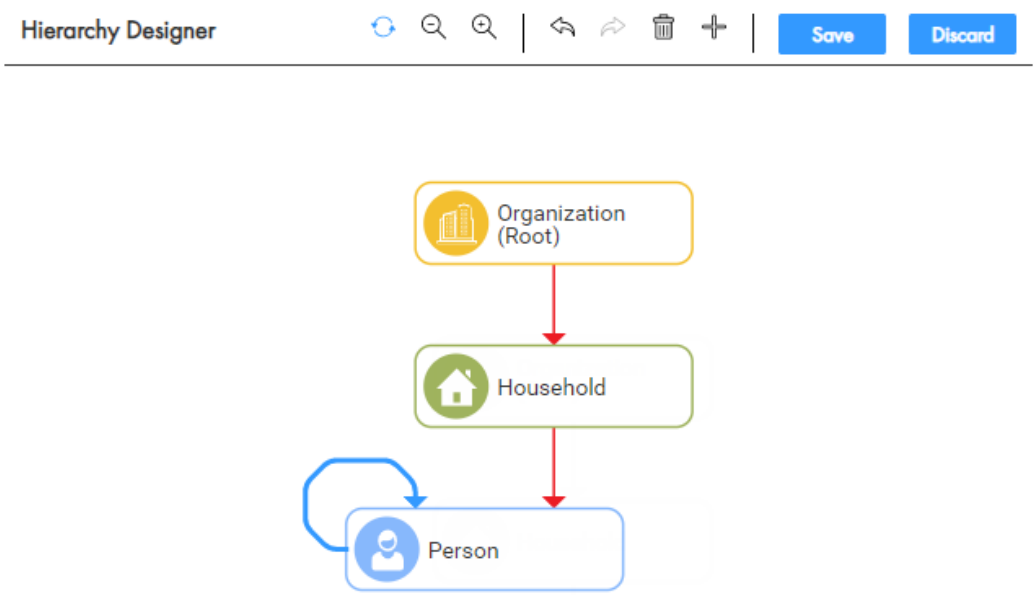
A hierarchy relationship is a parent-child relationship between business entities. A hierarchy contains a collection of hierarchy relationships between business entities. You can create multiple hierarchies to define the hierarchical relationships that are important to your organization.

Consider the following guidelines when you create hierarchy relationships:

- You must specify a business entity as the root business entity.
- You must define a direct or indirect relationship between the root business entity and each business entity in the hierarchy.
- You can create a relationship loop from one business entity to the same business entity. For example, you might create a relationship from the Person business entity to the Person business entity.

Note: The Provisioning tool supports relationship loops at the root business entity, but Data Director cannot display hierarchies for records with data loops at the root business entity level. For example, in the Provisioning tool, you create a relationship loop at the Organization root business entity. In Data Director, if the Informatica record has a relationship to the Informatica record, then Data Director cannot display the hierarchy for the Informatica record.

The following image shows a sample hierarchy in the Provisioning tool:



Network Relationships

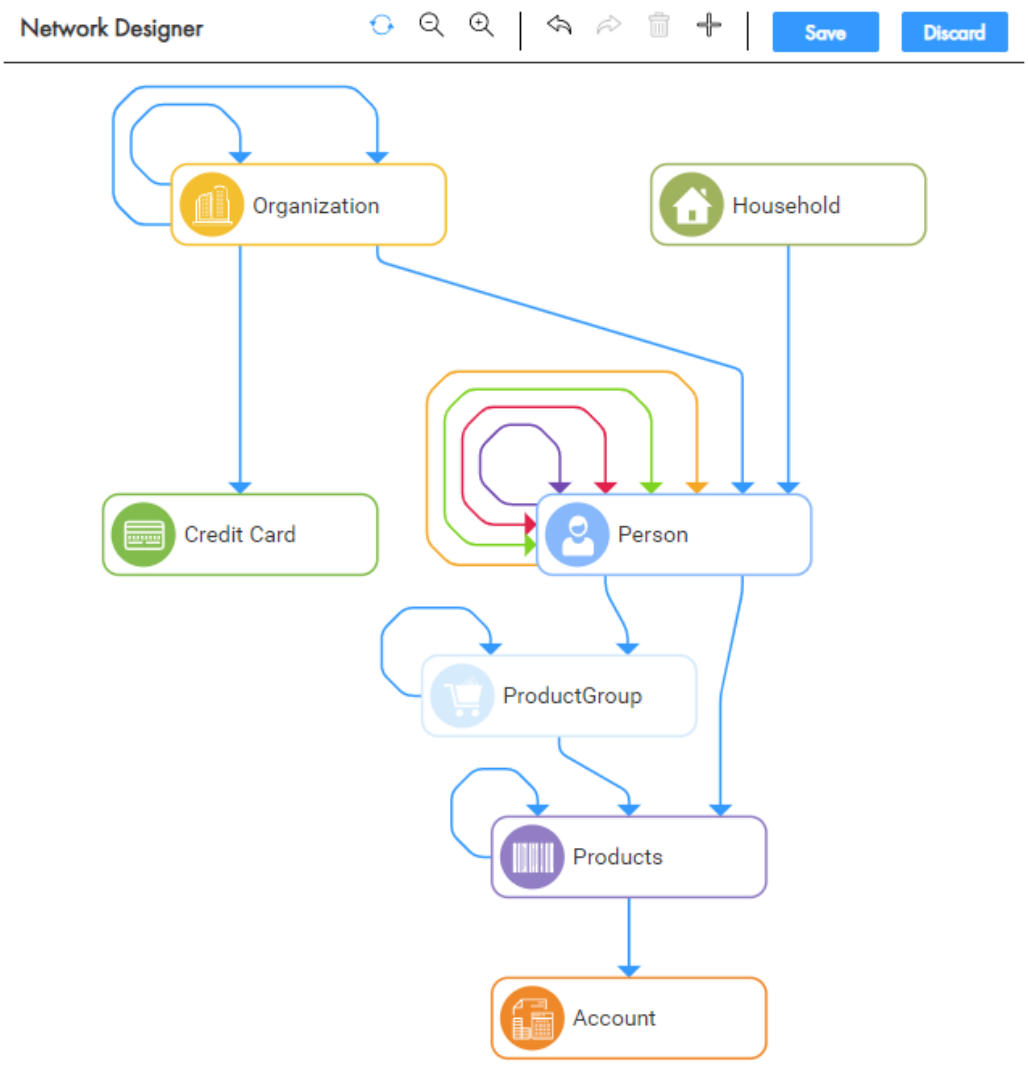
A network relationship is a connection between related business entities. Each network relationship that you create becomes part of the network.

The network is a collection of network relationships between business entities. There is only one network to which you add network relationships.

Consider the following guidelines when you create network relationships:

- You do not specify a root business entity.
- You can create a relationship loop from one business entity to the same business entity.

The following image shows a sample network in the Provisioning tool:



Copying Hierarchy Manager Relationships Creating Hierarchy Relationships

Use the Provisioning tool to copy relationships in Hierarchy Manager and create a hierarchy. You can create multiple hierarchies by copying additional relationships in Hierarchy Manager.

When you create a hierarchy by copying relationships, the Provisioning tool tries to find a corresponding business entity to configure as the root business entity node. It tries to find the root business entity based on the selected Root HM Entity. If it cannot find a corresponding business entity, it creates the business entity. Then the Provisioning tool creates the remaining hierarchy model based on the copied relationships. You must review the hierarchy and hierarchy relationships created in this process.

To copy relationships in Hierarchy Manager and then create a hierarchy, perform the following tasks:

1. Copy relationships in Hierarchy Manager and create hierarchies.
2. Configure the Hub Server to support hierarchy REST APIs.
3. Configure access to hierarchies.
4. Revert relationship base objects to base objects.

Copying Relationships in Hierarchy Manager and Creating Hierarchies

Copy relationships in Hierarchy Manager and create a hierarchy with hierarchy relationships. You can create additional hierarchies by copying additional hierarchy relationships from Hierarchy Manager.

Important: When you create a hierarchy by copying relationships, the Provisioning tool tries to find a corresponding business entity to configure as the root business entity node. It tries to find the root business entity based on the selected Root HM Entity. If it cannot find a corresponding business entity, it creates the business entity. Then the Provisioning tool creates the remaining hierarchy model based on the copied relationships. You must review the hierarchy and hierarchy relationships created by copying relationships.

1. Click **Business Entity** > **Modeling**, and then select **Hierarchies**.
2. Click **Create** > **Hierarchy Copied from Hierarchy Manager**.

   The **Create Hierarchy Copied from Hierarchy Manager** window appears.
3. In the **Select Existing Hierarchies** column, select a hierarchy that you want to copy.
4. In the **Select Root Entity** column, configure the following root entity settings:
   a. In the **Root HM Entity** field, select the HM entity you want to use as the root node.

      The Provisioning tool uses the Root HM Entity to determine the root business entity for the hierarchy.
   b. In the **New Hierarchy Name** field, enter a name for the hierarchy.
5. Click **Create New Hierarchy**.

   The Hierarchy Designer opens.
6. Review the business entities and relationships and based on the issue, resolve the issue:
   - If there is a relationship loop at the root business entity, remove the relationship loop by deleting the relationship or editing the hierarchy to specify a new business entity as the root business entity.

     Warning: The Provisioning tool supports relationship loops at the root business entity, but Data Director cannot display hierarchies for records with data loops at the root business entity level. For more information about guidelines for creating hierarchies, see "Hierarchy Relationships" on page 145.
   - If there are relationships with errors, select the relationship and review the relationship properties. Relationships with errors appear as red lines.
   - If there are business entity nodes with errors, select the node and review the business entity. Business entity nodes with errors appear with a red border.
7. Click **Save**.

Configuring the Hub Server for Hierarchies

Use the Hub Server tool in the Hub Console and the `cmxserver.properties` file to configure the Hub Server properties to support hierarchy REST APIs.

You can use hierarchy REST APIs to interact with your hierarchies. For more information about hierarchy REST APIs, see the *Multidomain MDM Business Entity Services Guide*.

1. Use a text editor to open the `cmxserver.properties` file in the following location: `<MDM Hub Installation Directory>\hub\server\resources\cmxserver.properties`.
2. Manually add and configure the following properties:

   cmx.server.hierarchy.max-search-depth

   Maximum depth searched when you use hierarchy REST APIs to find a hierarchy path. Default value is `100`.

   cmx.server.hierarchy.max-search-width

   Maximum width of the searched hierarchy to include when you use the hierarchy REST APIs to export. Default value is `1000000`.

Reverting Relationship Base Objects to Base Objects

Revert relationship base objects to base objects to remove the Hierarchy Manager metadata from the relationship object. The relationship object remains as a base object, but Hierarchy Manager does not display the base object.

If the relationship-type column that you want to revert is in the staging table for a lookup, the staging table column must be empty before you revert the relationship base object.

If you are upgrading Hierarchy Manager relationships, copy relationships in Hierarchy Manager to the Provisioning tool before you revert the relationship base object.

1. In the Hierarchies tool, acquire a write lock.
2. Right-click on the relationship base object and choose **Revert Entity/Relationship Object to BO**.
3. In the **Revert Entity/Relationship Object** dialog box, click **OK**.

   A dialog box appears when the entity is reverted.

Configuring Access to Hierarchies

Configure hierarchies as a secure resources and grant MDM roles access to hierarchies.

1. Open the Hub Console and select the database you want to administer.
2. Click **Write Lock** > **Acquire Lock**.
3. Configure the hierarchy as secure.
   a. In the **Workbenches** panel, under **Security Access Manager**, click **Secure Resources**.

      The **Secure Resources** panel appears.

   b. In the **Resources** tab, expand the **Business Entities Hierarchies** node.

   c. Select the business entity hierarchy that you want to configure as secure, and then click the **Change Resource Status to Secure** icon.

   d. Click the **Save** icon.

4. Grant MDM roles access to hierarchies.
    a. In the **Workbenches** panel, under **Security Access Manager**, click **Roles**.

       The **Roles** panel appears.
    b. Select **DataSteward**.
    c. In the **Resource Privileges** tab, expand the **Business Entity Hierarchies** node.
    d. Select the **Read**, **Create**, and **Update** check boxes for each of the hierarchies.
    e. Repeat steps **c** to **d** for the Manager and SrManager roles.
    f. Click the **Save** icon.
5. Validate your repository.
    a. In the **Workbenches** panel, under **Configuration**, click **Repository Manager**.
    b. Click **Connect to master database**.
    c. In the **Validate** tab, select the database for which you configured hierarchies access.
    d. Click the **Validate** icon.
    e. Click **OK**.

Copying Hierarchy Manager Relationships and Creating Network Relationships

Use the Provisioning tool to copy relationships in Hierarchy Manager and create them as network relationships.

To copy relationships in Hierarchy Manager and add create as network relationships, perform the following tasks:

1. Copy relationships in Hierarchy Manager and create network relationships.
2. Revert relationship base objects to base objects.

Adding Network Relationships by Copying Relationships in Hierarchy Manager

Copy relationships in Hierarchy Manager and add the relationships as network relationships. You can add additional network relationships by copying relationships in other hierarchies in Hierarchy Manager.

1. Click **Business Entity** > **Modeling**, and then select **Network**.
2. Click **Create** > **Relationships Copied from Hierarchy Manager**.

   The **Add Relationships Copied from Hierarchy Manager** window appears.
3. In the **Select Existing Hierarchies** column, select a hierarchy from which you want to copy relationships.
4. In the **Select Relationships** column, select the relationships that you want to copy.
5. Click **Add Selected to Network**.

   The Network Designer opens.
6. Review the relationships in the network. If there are relationships with errors, select the relationship and review the relationship properties.

Relationships with errors appear as red lines.

7. Click **Save**.

Reverting Relationship Base Objects to Base Objects

Revert relationship base objects to base objects to remove the Hierarchy Manager metadata from the relationship object. The relationship object remains as a base object, but Hierarchy Manager does not display the base object.

If the relationship-type column that you want to revert is in the staging table for a lookup, the staging table column must be empty before you revert the relationship base object.

If you are upgrading Hierarchy Manager relationships, copy relationships in Hierarchy Manager to the Provisioning tool before you revert the relationship base object.

1. In the Hierarchies tool, acquire a write lock.
2. Right-click on the relationship base object and choose **Revert Entity/Relationship Object to BO**.
3. In the **Revert Entity/Relationship Object** dialog box, click **OK**.

   A dialog box appears when the entity is reverted.

CHAPTER 11

ActiveVOS Post-Installation Tasks for the Application Server

This chapter includes the following topic:

ActiveVOS Post-Installation Tasks for the Application Server

Whether you perform a clean upgrade or an in-place upgrade, perform the ActiveVOS post-installation tasks for the application server to ensure your environment is properly configured.

For ActiveVOS post-installation tasks for the application server, see the ActiveVOS Post-Installation Tasks for the Application Server chapter in the *Multidomain MDM Installation Guide* that applies to your environment.

CHAPTER 12

ActiveVOS Post-Upgrade Tasks for Business Entity Adapter

This chapter includes the following topics:

ActiveVOS Post-Upgrade Tasks for the Business Entity Adapter

Whether you perform a clean upgrade or an in-place upgrade, if you use the ActiveVOS workflow adapter based on business entities, perform the ActiveVOS post-upgrade tasks for the business entity adapter to ensure your environment is properly configured.

Configuring the ActiveVOS URNs for the Business Entity Workflow Adapter

The ActiveVOS Server has two predefined uniform resource names (URNs) that it uses internally. You need to update the URL in the URN mappings to use the host name and the port number where the ActiveVOS Server runs.

1. Launch the ActiveVOS Console. In a browser, type the following URL, substituting the correct host name and port number:

   Encrypted connections.`https://[host]:[port]/activevos`

   Non-encrypted connections.`http://[host]:[port]/activevos`
2. In the ActiveVOS Console, on the Home page, click **Administration > Configure Server > URN Mappings**.
3. For the following URNs, update the paths to reflect the host name and port number of the ActiveVOS Server:

URN	URL Path
ae:internal-reporting	Encrypted connections. `https://[host]:[port]/activevos/internalreports` Non-encrypted connections. `http://[host]:[port]/activevos/internalreports`
ae:task-inbox	Encrypted connections. `https://[host]:[port]/activevos-central/avc` Non-encrypted connections. `http://[host]:[port]/activevos-central/avc`

4. Verify that **urn:mdm:service** is mapped to the host name and port number of the MDM Hub Server:

   Encrypted connections.`https://[host]:[port]/cmx/services/BEServices`

   Non-encrypted connections.`http://[host]:[port]/cmx/services/BEServices`

Set the ActiveVOS Protocol to HTTPS

To enable secure communication between ActiveVOS and the MDM Hub, set the protocol to HTTPS in the Hub Console Workflow Manager.

You must first configure the application server for HTTPS communications.

1. Start the Hub Console.
2. Acquire a write lock.
3. Click **Workflow Manager** under the Configuration workbench.
4. In the Workflow Manager, click the **Workflow Engines** tab.
5. Select the ActiveVOS workflow engine, and then click the **Edit** button.
6. In the Edit Workflow dialog box, set the protocol to HTTPS.
7. In a WebLogic environment, in the Edit Workflow dialog box, enter the user name and password of the user that belongs to the abAdmin role.

Update Customized Workflows for Business Entities

If you have customized workflows that work with the business entities adapter, review the list of changes and make updates to your customized workflows as required.

The following table lists the changes that were made to the default workflows over the releases:

Release	Change	Instructions
10 HotFix 1	Enable password encryption.	For each workflow, set `sif:encrypted` to `true`.
10 HotFix 1	Update the presentation parameters for the task inbox.	"Updating Presentation Parameters in Workflows for Business Entities" on page 154
10.3	Enable the file attachments feature.	"Enabling File Attachments in Workflows for Business Entities" on page 156

Updating Presentation Parameters in Workflows for Business Entities

If you customized a workflow based on business entities, open the .bpel file and compare the presentation parameters with the following list of parameters. Add or update the presentation parameters for each task as necessary.

While an expression can contain an absolute reference to a parameter, the best practice for customized workflows is to point to the `getInput()` method for the task. Consider replacing the absolute references to the presentation parameters with calls to the `getInput()` method. For example, replace `$ProcessTaskRequest/mdmavxsd:INFATask/` with `htd:getInput('processTaskRequest')/mdmavxsd:INFATask/`. The expressions in the following table call the input method.

Tip: The expressions use the namespace prefix `mdmavxsd`. If you encounter any issues after you update a custom workflow, verify that `mdmavxsd` is defined as a prefix for `urn:informatica.mdm.av.xsd`.

The following table lists the presentation parameters and expressions:

Parameter	Expression
hubUsername	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:hubUsername/text()
hubPassword	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:hubPassword/text()
securityPayload	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:securityPayload/text()
orsId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:orsId/text()
taskTypeName	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:name/text()
taskTypeDisplayName	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:displayName/text()
taskTypeDescription	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:description/text()

Parameter	Expression
pendingBVT	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/ mdmavxsd:pendingBVT
taskTypeDataUpdateType	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/ mdmavxsd:dataUpdateType/text()
taskTypeDisplayType	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/ mdmavxsd:displayType/text()
defaultApproval	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/ mdmavxsd:defaultApproval
taskDataTaskId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:taskId/text()
taskDataOwnerUID	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:ownerUID/text()
taskDataGroups	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:groups/mdmavxsd:groups/text()
dueDate	let $in := htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:dueDate/text() let $out :=
status	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:status/text()
taskDataPriority	length($out) > 0))), string-length($out) + (string-length($in)) * xsd:int((string-length($out) = 0)))
taskDataSubjectAreaUID	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:subjectAreaUID/text()
taskDataTitle	let $in := htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:title/text() let $out :=
taskDataComments	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:comments/text()
taskDataInteractionId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:interactionId/text()
taskDataCreator	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:creator/text()
createDate	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:createDate
taskDataUpdatedBy	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:updatedBy/text()
lastUpdateDate	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/ mdmavxsd:lastUpdateDate

Parameter	Expression
workflowVersion	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:workflowVersion/text()
beRowId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/mdmavxsd:taskRecords/mdmavxsd:INFARecordKey[1]/mdmavxsd:rowId/text()
bePkeySrcObject	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/mdmavxsd:taskRecords/mdmavxsd:INFARecordKey[1]/mdmavxsd:pkeySrcObject/text()
beSystem	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/mdmavxsd:taskRecords/mdmavxsd:INFARecordKey[1]/mdmavxsd:system/text()
beRowidXref	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/mdmavxsd:taskRecords/mdmavxsd:INFARecordKey[1]/mdmavxsd:rowidXref/text()
beTableUID	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/mdmavxsd:taskRecords/mdmavxsd:INFARecordKey[1]/mdmavxsd:tableUID/text()
taskTypeCreationType	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:creationType/text()

For more information about updating the .bpel files, see the ActiveVOS documentation.

Enabling File Attachments in Workflows for Business Entities

Effective in version 10.3, you can update your customized workflows to allow users to attach files to their tasks. To enable this feature in a customized workflow, update the process that initializes the data and update all the user processes.

Note: If you want to restrict the ability to add attachments to some tasks, you can configure the task settings in the Provisioning Tool. For more information about updating the task configuration, see the *Multidomain MDM Provisioning Tool Guide*.

1. In ActiveVOS Designer, open the .bpel file for a customized workflow.

   The workflow appears in the canvas.
2. Update the process that initializes the data.
   a. Select the **Initialize Data** process.

      The **Operations** dialog box appears.
   b. Click **New Script**.

      The **Script Builder** dialog box appears.
   c. In the **Script** box, add the following statement:

      ```
      abx:copyAllAttachments('StartRequest', '_peopleActivityAttachments')
      ```

      Replace *StartRequest* with the variable name that you use to initialize data in your workflow.
   d. Click **OK**.

      The **Script Builder** dialog box closes, and the script appears in the **Operation** list.
   e. Click **OK**.

      The **Operations** dialog box closes.

3. Update all the user processes.
    a. Select a user process.
    b. Click the **Properties** tab.
    c. Click the **All** tab.
    d. Expand **Attachment Propagation**.
    e. Set **From Process** to **All**.
    f. Set **To Process** to **newOnly**.
4. Save the .bpel file.

Configure the MDM Identity Services for ActiveVOS

If you use embedded ActiveVOS, ensure that you configure ActiveVOS to use MDM Identity Services. To configure the MDM Identity Services for ActiveVOS, use the ActiveVOS Console to set the Identity Services password to the password of the MDM Hub workflow engine user.

1. In the ActiveVOS console, select **Admin** > **Configure Services** > **Identity Services**.
2. In the Provider Configuration section, enable the **Enable** check box and select **MDM** from the **Provider Type** list.
3. In the Connection tab, enter the password of the MDM Hub user with the user name `admin`.

    Note: If you later change the password for the admin user, you must enter the new password in the ActiveVOS identity services settings.
4. Click **Update**.
5. Test that ActiveVOS can log in to the MDM Hub as the `admin` user, and that ActiveVOS can retrieve a list of roles for the user you specify as the **User for test**.
    a. Select the **Test** tab.
    b. In the **User for test** field, enter an MDM Hub user that is assigned to a role.
    c. Click **Test Settings**.

    Note: The test fails if an Operational Reference Store is not configured, the user for test does not belong to a role, or the role name contains spaces.

Custom BeMDMWorkflow Project (In-place Upgrade)

If you customized the workflows for the BeMDMWorkflow project, ensure that you update and then redeploy the project.

1. In Informatica ActiveVOS Designer, use the Project Explorer to navigate to **BeCommonMDM** > **wsdl** > **cs.wsdl**.
2. Right-click **cs.wsdl**, and then select **Open With** > **Text Editor**.
3. In the text editor, navigate to the following code:

```
<xsd:complexType name="TaskFilter">
                <xsd:sequence>
```

4. Add the following code:

```
<xsd:element minOccurs="0" name="overdueOnly" type="xsd:boolean" />
```

5. In Informatica ActiveVOS Designer, open the BeCommonMDM project to export.
6. Click **File** > **Export**.

   The **Export** dialog box opens.
7. Under **Orchestration**, select **Contribution - Business Process Archive**. Click **Next**.
8. In the **Deployment URL** field under **Server Deployment Option**, enter the URL for the ActiveVOS instance. Click **Finish**.
9. In the **Deployment Complete** dialog box, click **OK**.

Configure Unmerge and Merge Workflow Triggers (In-place Upgrade)

You must configure the unmerge and merge workflow triggers that were introduced in Multidomain MDM version 10.2. To configure the unmerge and merge workflow triggers, use the Provisioning tool.

For more information, see the *Multidomain MDM Provisioning Tool Guide*.

Add the Entity 360 Framework Task Manager

When you use the business entity ActiveVOS workflow adapter, you use the Entity 360 framework Task Manager and Entity 360 framework task inbox.

Update the Informatica Data Director configuration to replace the legacy task inbox. You can add the Entity 360 framework task inbox to the **Home** page. For more information about designing the Informatica Data Director user interface, see the *Multidomain MDM Provisioning Tool Guide*.

CHAPTER 13

ActiveVOS Post-Upgrade Tasks for Subject Areas Adapter

This chapter includes the following topics:

ActiveVOS Post-Upgrade Tasks for the Subject Area Adapter

Whether you perform a clean upgrade or an in-place upgrade, if you use the ActiveVOS workflow adapter based on subject areas, perform the ActiveVOS post-upgrade tasks for the subject area adapter to ensure your environment is properly configured.

Update the ActiveVOS URNs

To use the HTTP Secure (HTTPS) protocol for secure communication between the MDM Hub and ActiveVOS, change the URLs in the URN paths from http to https.

1. Launch the ActiveVOS Console. In a browser, type the following URL, substituting the correct host name and port number:

JBoss Version 7.2 or 7.1	URL Path
Secure connections	`https://<host>:<port>/activevos`
Non-secure connections	`http://<host>:<port>/activevos`

2. In the ActiveVOS Console, on the Home page, click **Administration > Configure Server > URN Mappings**.
3. For the following URNs, update the paths to reflect the host name and port number of the ActiveVOS Server:

URN	URL Path
ae:internal-reporting	Secure connections. `https://<host>:<port>/activevos/internalreports` Non-secure connections. `http://<host>:<port>/activevos/internalreports`
ae:task-inbox	Secure connections. `https://<host>:<port>/activevos-central/avc` Non-secure connections. `http://<host>:<port>/activevos-central/avc`

4. Verify that **MDMHost:InfaMDM** is mapped to the host name and port number of the MDM Hub Server:

   Secure connections.`https://<host>:<port>/cmx/services/SifService`

   Non-secure connections.`http://<host>:<port>/cmx/services/SifService`

Verifying the Trusted User for ActiveVOS

In the Hub Console, verify that the ActiveVOS workflow engine settings specifies the trusted user.

1. In the Hub Console, on the Configuration workbench, click **Workflow Manager**.
2. Select the **Workflow Engines** tab.
3. Acquire a write lock.
4. Select **ActiveVOS** and click the **Edit** button.
5. In the Edit Workflow dialog box, enter the user name and password of the trusted user
6. Click **OK**.

Update Informatica Data Director Task Configuration for ActiveVOS Workflows based on Subject Areas

To use the subject area-based ActiveVOS workflow adapter with the Task Manager, you must update the Informatica Data Director configuration file. If you use ActiveVOS workflows based on subject areas, you cannot migrate to ActiveVOS workflows based on business entities.

You can configure the following task parameters in the Informatica Data Director configuration file:

taskType

Describes the task type.

taskTypeID

The process name.

name

The taskType name. The name must be the same as the name of the task in the ActiveVOS workflow configuration.

Update the IDD Configuration for the Subject Area-based ActiveVOS Adapter

To use the subject area-based ActiveVOS workflow adapter with the business entity-based Task Manager, update the Data Director configuration file. If you do not update the Data Director configuration file, you cannot use the Task Manager to create tasks.

The following code sample shows how to configure subject area-based ActiveVOS tasks in the Data Director configuration file for the workflows provided with Multidomain MDM:

```
<tasks includeUnassignedTasks="true">
<!-- Task Definitions -->
<taskType taskTypeId="IDDMergeTask" name="AVOSMerge" displayName="Merge"
creationType="MERGE" displayType="MERGE">
  <description>Merge two records together.</description>
</taskType>

<taskType taskTypeId="IDDUnmergeTask" name="AVOSUnmerge" displayName="Unmerge"
creationType="UNMERGE" displayType="UNMERGE">
  <description>Unmerge an XREF record from a Base Object record.
  </description>
</taskType>

<taskType taskTypeId="IDDOneStepApprovalTask" name="AVOSFinalReview"
displayType="NORMAL" displayName="Final review" creationType="NONE" pendingBVT="true">
  <description>Update a record and require the user to go through an approval process
before completing the task.
  </description>
</taskType>

<taskType name="Notification" displayName="Notification" creationType="NONE"
displayType="NORMAL">
   <description>Notification step in the workflow</description>
</taskType>

<taskType taskTypeId="IDDTwoStepApprovalTask" name="AVOSReviewNoApprove"
 displayType="NORMAL" displayName="Review no approve" creationType="NONE"
 defaultApproval="true" pendingBVT="true">
  <description>Update a record and require the user to go through an approval process
    before completing the task.
  </description>
</taskType>
```

```
<taskType taskTypeId="IDDUpdateWithApprovalTask" name="Update" displayType="NORMAL"
displayName="Update" creationType="CREATE" pendingBVT="true">
  <description>Update a record and do not require the user to go through an approval
    process before completing the task. The approval step is optional.
  </description>
</taskType>

</tasks>
```

Configure Task Triggers For Subject Area Workflow Adapter

You must configure task triggers to use ActiveVOS workflows based on subject areas with the Task Manager. If you do not configure task triggers, the tasks do not appear in the Task Manager.

To configure triggers, use the Provisioning tool to edit the task configuration file from the Advance Configuration page. For more information, see the *Multidomain MDM Provisioning Tool Guide*.

You can configure the following `startWorkflow` attributes to configure task triggers:

process

The name of the ActiveVOS workflow process.

taskKind

Defines the type of user interface required for the process. Can be REVIEW, MERGE, or UNMERGE. The taskKind is returned by the ActiveVOS workflow engine.

taskTemplate

The name of the task template to use.

firstTask Type

The first task in the workflow. Optional. This parameter allows the task to be assigned when the task is created.

Two-step approval code sample

The following code sample shows the startWorkflow element configuration for the ActiveVOS adapter based on subject areas for the two-step approval task:

```
<trigger name="DefaultApproval">
<startWorkflow process="IDDTwoStepApprovalTask" taskKind="REVIEW"
taskTemplate="DefaultApproval" firstTaskType="AVOSReviewNoApprove"/>
<event name="CreateBE"/>
<event name="UpdateBE"/>
<role name="*"/>
</trigger>
```

One-step approval code sample

The following code sample shows the startWorkflow element configuration for the ActiveVOS adapter based on subject areas for the one-step approval task:

```
  <trigger name="DefaultApproval">
<startWorkflow process="IDDOneStepApprovalTask" taskKind="REVIEW"
taskTemplate="DefaultApproval" firstTaskType="AVOSFinalReview"/>
<event name="CreateBE"/>
<event name="UpdateBE"/>
<role name="*"/>
</trigger>
```

Update with approval code sample

The following code sample shows the startWorkflow element configuration for the ActiveVOS adapter based on subject areas for the update-with-approval task:

```
<trigger name="DefaultApproval">
<startWorkflow process="IDDUpdateWithApprovalTask" taskKind="REVIEW"
taskTemplate="DefaultApproval" firstTaskType="Update"/>
<event name="CreateBE"/>
<event name="UpdateBE"/>
<role name="*"/>
</trigger>
```

Merge code sample

The following code sample shows the startWorkflow element configuration for the ActiveVOS adapter based on subject areas for the merge task:

```
<trigger name="Matched">
<startWorkflow process="IDDMergeTask" taskKind="MERGE"
taskTemplate="MergeTaskGenerator" firstTaskType="AVOSMerge"/>
<event name="MatchedBE"/>
<role name="SYSTEM"/>
</trigger>
```

Update Customized Workflows for Subject Areas

If you have customized workflows that work with the subject area adapter, review the list of changes and make updates to your customized workflows as required.

The following table lists the changes that were made to the default workflows over the releases:

Release	Change	Instructions
10 HotFix 1	Enable password encryption.	For each workflow, set `sif:encrypted` to `true`.
10 HotFix 1	Update the presentation parameters for the task inbox.	"Updating Presentation Parameters in Workflows for Subject Areas" on page 163
10.3	Enable the file attachments feature.	"Enabling Attachments in Workflows for Subject Areas" on page 164

Updating Presentation Parameters in Workflows for Subject Areas

If you customized a workflow based on subject areas, open the .bpel file and compare the presentation parameters with the following list of parameters. Add or update the presentation parameters for each task as necessary.

While an expression can contain an absolute reference to a parameter, the best practice for customized workflows is to point to the `getInput()` method for the task. Consider replacing the absolute references to the presentation parameters with calls to the `getInput()` method. For example, replace `htd:getInput('processTaskRequest')/mdmavxsd:INFATask/` with `htd:getInput('processTaskRequest')/mdmavxsd:INFATask/`. The expressions in the following table call the input method.

Tip: The expressions use the namespace prefix `mdmavxsd`. If you encounter any issues after you update a custom workflow, verify that `mdmavxsd` is defined as a prefix for `urn:informatica.mdm.av.xsd`.

The following table lists the presentation parameters and expressions:

Parameter	Type	Expression
subjectareauid	string	$InfaTask/mdmavxsd:taskData/mdmavxsd:subjectAreaUID
title	string	$InfaTask/mdmavxsd:taskData/mdmavxsd:title
creator	string	$InfaTask/mdmavxsd:taskData/mdmavxsd:creator
mdmtasktype	string	$InfaTask/mdmavxsd:taskType/mdmavxsd:name
orsId	string	$InfaTask/mdmavxsd:orsId
duedate	string	$InfaTask/mdmavxsd:taskData/mdmavxsd:dueDate
tasktypename	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:name
taskTypeDisplayName	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:displayName
taskTypeDescription	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:description
taskTypePendingBVT	boolean	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:pendingBVT
taskTypeDataUpdateType	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:dataUpdateType
taskTypeDisplayType	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:displayType
priorityOut	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:priority
workflowVersion	string	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:workflowVersion/text()

For more information about updating .bpel files, see the ActiveVOS documentation.

Enabling Attachments in Workflows for Subject Areas

Effective in version 10.3, you can update your customized workflows to allow users to attach files to their tasks. To enable this feature in a customized workflow, update the process that initializes the data and update all the user processes.

Note: If you want to restrict the ability to add attachments to some tasks, you can configure the task settings in the Provisioning Tool. For more information about updating the task configuration, see the *Multidomain MDM Provisioning Tool Guide*.

1. In ActiveVOS Designer, open the .bpel file for a customized workflow.

   The workflow appears in the canvas.

2. Update the process that initializes the data.
    a. Select the **Initialize Data** process.

       The **Operations** dialog box appears.
    b. Click **New Script**.

       The **Script Builder** dialog box appears.
    c. In the **Script** box, add the following statement:

       ```
       abx:copyAllAttachments('StartRequest', '_peopleActivityAttachments')
       ```

       Replace *StartRequest* with the variable name that you use to initialize data in your workflow.
    d. Click **OK**.

       The **Script Builder** dialog box closes, and the script appears in the **Operation** list.
    e. Click **OK**.

       The **Operations** dialog box closes.
3. Update all the user processes.
    a. Select a user process.
    b. Click the **Properties** tab.
    c. Click the **All** tab.
    d. Expand **Attachment Propagation**.
    e. Set **From Process** to **All**.
    f. Set **To Process** to **newOnly**.
4. Save the .bpel file.

Redeploy the ActiveVOS Workflows based on Subject Areas

The ActiveVOS workflows for the workflow adapter that is based on subject areas changed in version 10.0 HotFix 2. If you use the workflow adapter based on subject areas, you must redeploy the default workflows that are provided in the Resource Kit. To deploy the Informatica ActiveVOS project that contains the task workflows to the MDM Hub Server, first export the CommonMDM project and then export the MDMWorkflow project.

1. In Informatica ActiveVOS Designer, open the BeCommonMDM project to export.
2. Click **File** > **Export**.

   The **Export** dialog box opens.
3. Under **Orchestration**, select **Contribution - Business Process Archive**. Click **Next**.
4. In the **Deployment URL** field under **Server Deployment Option**, enter the URL for the ActiveVOS instance. Click **Finish**.
5. In the **Deployment Complete** dialog box, click **OK**.
6. Repeat all steps for the BeMDMWorkflow project.

   You must export the BeCommonMDM project before you export the BeMDMWorkflow project.

Generating Business Entity and Business Entity Services Configuration Files

To generate business entity and business entity services configuration files, use the Informatica Data Director Configuration Manager.

1. In the Configuration Manager **Applications** pane, select the Informatica Data Director application whose configuration you want to generate into a business entity and business entity service configuration.
2. Click **Generate Business Entity Schema**.

   Configuration Manager generates the business entity and business entity service configuration.
3. Configuration Manager displays messages of issues encountered while generating the business entity and business entity services configuration. The messages indicate if Configuration Manager resolved the issue during the generation process and describes the changes Configuration Manager made to resolve the issue. If Configuration Manager did not fix the issue, note the issue and the suggested action you can take to resolve the issue.

APPENDIX A

Troubleshooting the Upgrade Process

If the upgrade fails or you encounter issues during the upgrade, use the following information to troubleshoot the problem.

The EAR files do not deploy within the permitted time in JBoss environments.

As you increase the number of Operational Reference Stores, the EAR file deployment time increases. If the EAR file deployment time exceeds the permitted deployment time in JBoss environments, the upgrade fails.

To resolve the issue, increase the permitted deployment time to accommodate the EAR file deployment time. The default permitted deployment time is 600 seconds.

1. Increase the value of the `deploy.wait.time` property in the `build.properties` file in the following directory: `<infamdm installation directory>/hub/server/bin`
2. Navigate to the following directory: `<JBoss installation directory>/standalone/configuration`
3. Configure the following code in the `standalone-full.xml` file to increase the timeout value:

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
    <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-
interval="5000" deployment-timeout="1200"/>
</subsystem>
```

The Hub Server upgrade fails.

To resolve the issue, redeploy the EAR file to retry the Hub Server upgrade.

Note: In JBoss environments, if you manually change the configuration of data sources in the `standalone-full.xml` file when JBoss is running, you lose the configuration changes when you run the `patchInstallSetup` script.

1. Navigate to the following directory: `<MDM Hub installation directory>/hub/server`
2. Run the following command to deploy the Hub Server application and apply changes to the application server configuration.

   Note: If you do not have embedded ActiveVOS in your environment, you do not need to include the ActiveVOS user names and passwords in the command.

On UNIX

WebLogic

```
patchInstallSetup.sh -Dweblogic.password=<WebLogic password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
```

```
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere with security enabled

```
patchInstallSetup.sh -Dwebsphere.password=<WebSphere password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere with security disabled

```
patchInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

JBoss

```
patchInstallsetup.sh -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

Note: On UNIX, if you include an exclamation mark (!) character in the password, you must include a backslash before the exclamation mark (!) character. For example, if the password is `!!cmx!!`, enter `\!\!cmx\!\!`.

On Windows

WebLogic

```
patchInstallSetup.bat -Dweblogic.password=<WebLogic password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere with security enabled

```
patchInstallSetup.bat -Dwebsphere.password=<WebSphere password> -
Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS
Console username> -Davos.password=<ActiveVOS Console password> -
Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere with security disabled

```
patchInstallSetup.bat -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

JBoss

```
patchInstallsetup.bat -Ddatabase.password=<MDM Hub Master database password> -
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console
password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

The ActiveVOS Console credentials are the same credentials as the administrative user in the application server.

The ActiveVOS database credentials are the same credentials that were used to run the create_bpm script.

The MDM Hub clean upgrade fails.

After an upgrade to version 10.4, when you perform a clean upgrade, but use the existing database, the upgrade fails. The issue occurs because each time you perform a clean upgrade, new public and private key pairs are generated. You cannot use the new key pairs to access the existing database.

To resolve the issue, perform the following steps:

1. Replace the new public and private key pairs for the Hub Server with the old public and private key pairs.

   The public and private key pairs for the Hub Server are stored in the following directory:

   ```
   <MDM Hub installation directory>/hub/server/resources/certificates
   ```

2. Run the `postInstallSetup` script to deploy the Hub Server application and apply the changes to the application server configuration.

   For more information about running the `postInstallSetup` script, see the Hub Server Post-Installation Tasks chapter in the *Multidomain MDM Installation Guide*.

3. Replace the new public and private key pairs for the Process Server with the old public and private key pairs.

   The public and private key pairs for the Process Server are stored in the following directory:

   `<MDM Hub installation directory>/hub/cleanse/resources/certificates`

4. Run the `postInstallSetup` script to deploy the Process Server application and apply the changes to the application server configuration.

   For more information about running the `postInstallSetup` script, see the Process Server Post-Installation Tasks chapter in the *Multidomain MDM Installation Guide*.

The Process Server upgrade fails in a WebLogic environment.

When you upgrade the Process Server in a WebLogic environment, the upgrade might fail with the following error:

```
Unable to start application, deployment error msg:
weblogic.management.ManagementException: [Deployer:149196]Rejecting start request for
application siperian-mrm-cleanse.ear because stop request is running for the application.
```

To resolve the issue, use the WebLogic Administrative Console to manually deploy the `siperian-mrm-cleanse.ear` file, and then restart the application server.

The Process Server upgrade fails in a WebSphere environment.

When you upgrade the Process Server in a WebSphere environment, the upgrade might fail. Open the `patchinstallSetup.log` to see the following error:

```
Failed to load webapp: com/delos/cmx/server/
Initialization.getCleanseLoggingConfiguration;loaded from file:/E:/IBM/WAS9053/AppServer/
profiles/AppSrv01/installedApps/MDMWSDB2W16001Node02Cell/siperian-mrm-
cleanse.ear.ear/lib/siperian-server
```

To resolve the issue, use the WebSphere Administration Console to manually undeploy the `siperian-mrm-cleanse.ear` file, and then run the `patchInstallSetup` script.

You can find the `patchInstallSetup` script in the following directory:

`<MDM Hub installation directory>/hub/cleanse`

The Process Server upgrade fails.

To resolve the issue, redeploy the EAR file to retry the Process Server upgrade.

Note: If you manually change the configuration of data sources in the `standalone-full.xml` file when JBoss is running, you lose the configuration changes when you run the `patchInstallSetup` script.

1. Navigate to the following directory: `<MDM Hub installation directory>/hub/cleanse`
2. Run the following command to deploy the Process Server application and apply changes to the application server configuration.

On UNIX

WebLogic

```
patchInstallSetup.sh -Dweblogic.password=<WebLogic password>
```

WebSphere

```
patchInstallSetup.sh -Dwebsphere.password= <WebSphere password>
```

JBoss

```
patchInstallsetup.sh
```

On Windows

WebLogic

```
patchInstallSetup.bat -Dweblogic.password=<WebLogic password>
```

WebSphere

```
patchInstallSetup.bat -Dwebsphere.password= <WebSphere password>
```

JBoss

```
patchInstallsetup.bat
```

Note: On UNIX, if you include an exclamation mark (!) character in the password, you must include a backslash before the exclamation mark (!) character. For example, if the password is `!!cmx!!`, enter `\!\!cmx\!\!!`.

Operational Reference Store upgrade results in ORA-00955 error.

When you upgrade the Operational Reference Store, the upgrade is successful but the following error appears in the `sip_ant` log:

```
[exec] CREATE SEQUENCE "C_REPOS_ZDT_EVENT_SEQ" MINVALUE 1 MAXVALUE 99999999999999
     INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER CYCLE
[exec] *
[exec] ERROR at line 1:
[exec] ORA-00955: name is already used by an existing object
[exec]
```

You can safely ignore the error.

Change list promotion to an empty Operational Reference Store results in ORA-00910 error.

When you promote a change list to an empty Operational Reference Store, if the total length of the match column is greater than 4000, the following error occurs:

```
ORA-00910: specified length too long for its datatype
```

To promote a change list to an empty Operational Reference Store, ensure that the match column length that the MDM Hub adds to the external match input table does not exceed 4000. The match column length is the sum of the lengths of all base object columns that are sources of the match column and the number of source columns.

Operational Reference Store upgrade in an Oracle environment results in ORA-20005 error.

If you encounter error ORA-20005 when you run `sip_ant updateorsdatabase`, perform the following steps:

1. Run the following command to grant the required permissions:

```
exec
dbms_java.grant_permission(upper('ORS_USER'),'SYS:java.net.SocketPermission','*',
'connect,resolve');
```

2. Run the following command to confirm that the Java classes are loaded in Oracle:

```
select dbms_java.longname(object_name), status from user_objects where
object_type='JAVA CLASS';
```

3. If the classes are not loaded, run the following command to reload the classes:

```
loadjava -verbose -force -resolve -oracleresolver -user &ors_name/
&ors_passwd@&tns_name siperian-cleansecaller.jar
loadjava -verbose -force -resolve -oracleresolver -user &ors_name/
&ors_passwd@&tns_name siperian-dbutil.jar
```

The Hub Store upgrade fails.

You cannot rerun the Hub Store upgrade on a partially upgraded schema. If the upgrade fails, restore the database from a full backup, and then rerun the Hub Store upgrade.

If the Hub Store upgrade fails because column names contain reserved words, contact Informatica Global Customer Support for scripts to migrate the data to renamed columns.

After upgrading from a non-English locale, some tables are in English and some are in the language of the locale.

If your Hub Store database environment is set to a non-English locale, you must change the character set to Unicode before you run the upgrade scripts to upgrade the MDM Hub Master Database and Operational Reference Stores. During the upgrade, all table metadata is translated to English with a translation key. If you did not select a Unicode character set, only some tables are translated.

Hub Console fails to launch

Verify that you are using a Java runtime environment (JRE) that is supported for the Hub Console. For system requirements, see the Product Availability Matrix for this version of Multidomain MDM on Informatica Network:
https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

The Hub Console fails to launch in a JBoss environment

In JBoss environments, if the JBoss application server does not restart, you cannot launch the Hub Console. The MDM Hub generates an error to indicate that the repository layer did not initialize.

To resolve the issue, run the following code in a batch file to restart JBoss:

```
rmdir C:\<JBoss installation directory>\standalone\tmp /s /q
mkdir C:\<JBoss installation directory>\standalone\tmp
C:\<JBoss installation directory>\bin\standalone.bat -c standalone-full.xml -b 0.0.0.0
```

Hub Console fails to launch in a Db2 environment

In an MDM Hub environment with Db2 datasources, the Hub Console fails to launch with the following errors:

```
SIP-09070: SIP-10318: Couldn't get users due to data access error.

SIP-10324: There was an unexpected exception when attempting to load data object(s).
java.lang.NullPointerException
```

This issue is caused by a mismatch in the case used for the administrative user name in the MDM Hub and in the application server. For example, the MDM Hub has the administrative user DB2ADMIN (uppercase) while the application server has db2admin (lowercase).

To resolve the issue, ensure that the user name in the application server exactly matches the user name in the MDM Hub.

Note: To avoid issues related to case-sensitivity, Informatica recommends using all uppercase letters when defining user names for Db2.

For example, if you are using WebSphere, set the user name in the WebSphere Console.

1. Open the WebSphere Console.

2. Navigate to **Resources > Data sources > siperian-cmx_system-ds > Custom properties**.
3. In the User field, type in uppercase: `DB2ADMIN`
4. In the Password field, type the password for this user.
5. Click **Apply**, and then click **Save**.
6. Restart WebSphere.
7. Launch the Hub Console and log in.

Changes made in the Provisioning tool cannot be applied in Db2 environments.

In Db2 environments, if the Operational reference Store is large and you apply changes in the Provisioning tool, the following error message appears:

```
Failed to set user workspace configuration.
```

To resolve this issue, increase the column length of the user workspace table by running the following Db2 commands on the Operational reference Store:

```
UPDATE C_REPOS_COLUMN SET DATA_LENGTH = 50000000 WHERE TABLE_NAME =
'C_REPOS_USER_WORKSPACE' AND COLUMN_NAME = 'WORKSPACE_DATA'
ALTER TABLE C_REPOS_USER_WORKSPACE ALTER COLUMN WORKSPACE_DATA SET DATA TYPE BLOB(50M)
REORG TABLE C_REPOS_USER_WORKSPACE
COMMIT
```

IDD cannot use the legacy Data View to view records that are based on subject areas.

The default page to view records in IDD is the Entity View that is based on business entities.

To use the legacy Data View, set `dataview.enabled` to `true` in the `cmxserver.properties` file.

For more information, see the following How-to article: *Migrating IDD Applications to the Business Entity Data Model*.

IDD fails with the error SIP-BV-11500.

IDD can fail with the following error: `SIP-BV-11500 Fatal Error Operational Reference Store localhost-orcl-MDM_SAMPLE does not have a workflow engine configured. Each Operational Reference Store must have a workflow engine configured for use with the IDD even if workflow will not be used.`

To resolve this issue, ensure that the primary workflow adapter is configured.

For more information, see the following KB article:
https://kb.informatica.com/solution/23/Pages/55/381456.aspx?myk=381456.

When you validate the metadata, an error states that the object exists in the metadata but not in the database.

When you use the Repository Manager to fix the issue, the following error occurs: ORA-00955 Name is already used by an existing object.

To resolve the issue, ensure that the correct privileges for the proxy role are granted for the tables that encounter the error. Refer to a table that does not encounter the error to get the list of permissions that are required.

On Windows, when match tokens are generated, an error occurs.

The Generate Match Tokens process returns an error that says that the class `ssa.ssaname3.jssan3cl` cannot be initialized.

1. Verify that the PATH environment variable includes the path to the following directory, which contains the dynamic linked library (DLL) files for SSA-NAME3: `<MDM installation directory>/hub/cleanse/lib`
2. Verify that Microsoft Visual C++ Redistributable for Visual Studio 2019 is installed on the Process Server that performs search and match for the MDM Hub.
3. If Microsoft Visual C++ Redistributable for Visual Studio 2019 is installed, use a dependency checker, such as Dependency Walker (depends.exe), to load `jssan3cl.dll` and confirm that the Visual C++ Redistributable was successfully applied.

   Tip: Visual C++ Redistributable for Visual Studio 2019 requires that Windows Server has operating system patches installed. Check the operating system requirements before installing Visual C++ Redistributable. For example, from a baseline version of Windows Server 2012, you must apply around 100 patches (totalling approximately 2 GB) to the operating system before you can successfully install Visual C++ Redistributable.

After you upgrade in a Microsoft SQL Server environment on a WebLogic application server, you cannot log in to the Hub Console.

A null pointer exception occurs when you log in to the Hub Console.

To resolve the issue, comment out the drop commands, create schema commands, and any role commands in the `xa_install.sql` script located in `<Microsoft SQL Server installation directory>\sqljdbc_4.0\enu\xa`. Run the script, and then restart the application server.

The upgrade component patchInstallSetup fails when you install the Hub Server on a WebSphere Application Server.

To resolve the issue, open the file `<WebSphere profile home>/properties/soap.client.props` and increase `com.ibm.SOAP.requestTimeout`, and then restart the WebSphere server profile. Run `patchInstallSetup.bat` again.

The entity360view.ear file fails to deploy when you upgrade the Hub Server in IBM AIX environments.

To resolve the issue, run the `patchInstallSetup.sh` script.

APPENDIX B

Frequently Asked Questions

Do we need a new license file to upgrade?

Yes. If you are upgrading from aMultidomain MDM version 10.1.x or earlier, you need a new license file.

Can we use database user exits from a pre-10.0 version?

No. Database user exits that run in the database layer are deprecated in version 10.0 and later.

Why do we need to provide the DBA username and password during the upgrade process?

The upgrade process performs actions that require DBA-level permissions, such as granting privileges and creating sequences. The DBA credentials are necessary to allow the upgrade process to perform these actions.

What happens to the existing version of ActiveVOS during the upgrade process?

If you have ActiveVOS installed in your environment and you do an in-place upgrade, the upgrade process will install the latest version of ActiveVOS if your environment does not already have it. To see which version of ActiveVOS is required, see the Product Availability Matrix on Informatica Network: https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

Is it mandatory to install ActiveVOS during the MDM upgrade process in an environment that does not have ActiveVOS installed?

No. During the upgrade process, you are prompted to choose whether you want to install ActiveVOS.

Has the recommended screen resolution for Informatica Data Director changed in this version?

No, the recommended screen resolution has not changed. The recommended screen resolution for Informatica Data Director is 1280 x 1024.

Do we need to upgrade Informatica Data Quality when we upgrade to Multidomain MDM version 10.2 or later?

Yes, if you use Informatica Data Quality (IDQ) in your environment, you must upgrade to version 10.1 of IDQ. For the system requirements, see the Product Availability Matrix on Informatica Network: https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

How do we customize MDM Hub security?

For information about customizing MDM Hub security, see the *Multidomain MDM Security Guide* and see the How-to article *Using the Security Configuration Utility in Multidomain MDM*.

Is Java 8 supported?

Yes, this version of Multidomain MDM supports Java 8. For the system requirements, see the Product Availability Matrix on Informatica Network: https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

Note: ActiveVOS requires Java 7.

APPENDIX C

Processing Existing ActiveVOS Tasks

This appendix includes the following topics:

Processing Existing ActiveVOS Tasks Overview

To work with ActiveVOS tasks that were created before Multidomain MDM version 10.1, routinely run a migration script to populate the tasks with the required presentation parameters. If you do not run the migration script, the tasks do not appear in the Task Manager. Run the migration script until you process all the tasks that were created before you upgraded to version 10.1.

The migration script requires that you set some properties. You can add the properties to a build file or you can add them in the command line.

Migration Properties

The following table describes the migration properties:

Property	Description
avos.jdbc.database.driver.jar	Path to the JAR file that contains the JDBC driver for ActiveVOS database. This parameter is populated during the Hub Server installation without the avos prefix in <infamdm installation directory>\conf\avos.install.properties.
avos.jdbc.database.driver.class	JDBC driver class for ActiveVOS database. This parameter is populated during the Hub Server installation without the avos prefix in <infamdm installation directory>\conf\avos.install.properties.

Property	Description
avos.jdbc.database.url	ActiveVOS database connection URL. This parameter is populated during the Hub Server installation without the avos prefix in <infamdm installation directory>\conf\avos.install.properties.
avos.jdbc.database.username	ActiveVOS database user name. This parameter is populated during the Hub Server installation without the avos prefix in <infamdm installation directory>\conf\avos.install.properties.
avos.jdbc.database.password	ActiveVOS database password.
avos.ws.protocol	The protocol for the ActiveVOS server connection. Can be http or https.
avos.ws.host	Host name of the application server where ActiveVOS runs.
avos.ws.port	Port number of the application server connection.
avos.ws.trusted.username	User name of the trusted user. **Note:** The trusted user is created as part of the Multidomain MDM installation and upgrade process.
avos.ws.trusted.password	Password for the trusted user. **Note:** The trusted user is created as part of the Multidomain MDM installation and upgrade process.
avos.hub.username	MDM Hub user that belongs to all task management roles.
avos.ws.pagesize	Number of tasks processed in one database transaction and batch-loaded from ActiveVOS.
avos.ws.statuses	Optional. Comma-separated list of ActiveVOS task statuses to be processed. For example, READY or IN_PROGRESS. By default all tasks are processed.

Running the Migration Script with a Properties File

Create an MDM user that belongs to the roles associated with the workflows. Add the migration properties to the Hub Server `build.properties` file, and then run the MDM Hub Master Database with the migration script command. After the migration, task owners can continue to take action on their assigned tasks.

Run the migration script on a regular schedule. After all the tasks for the subject area workflow adapter are processed, you no longer have to run the script and you can delete the MDM user that you created for this purpose.

1. Create an MDM Hub user, and assign the user all the roles that participate in workflows.
   For example, the predefined workflows use the following roles: DataSteward, Manager, and SrManager.
2. Open the following file in a text editor:

   `<MDM Hub installation directory>/hub/server/bin/build.properties`
3. Add the migration properties to the `build.properties` file. For a list of properties, see "Migration Properties" on page 176.
4. Open a command prompt.

5. Navigate to the following directory:

   ```
   <MDM Hub installation directory>/hub/server/bin
   ```

6. Run the MDM Hub Master Database upgrade script with the migration script command:
   - On UNIX. `sip_ant.sh migrate-avos-sa-tasks`
   - On Windows. `sip_ant.bat migrate-avos-sa-tasks`

Running the Migration Script with Properties on the Command Line

Create an MDM user that belongs to the roles associated with the workflows. Open a command prompt and run the script with the desired properties. After the migration, task owners can continue to take action on their assigned tasks.

Run the migration script on a regular schedule. After all the tasks for the subject area workflow adapter are processed, you no longer have to run the script and you can delete the MDM user that you created for this purpose.

1. Create an MDM Hub user, and assign the user all the roles that participate in workflows.
   For example, the predefined workflows use the following roles: DataSteward, Manager, and SrManager.
2. Open a command prompt.
3. Navigate to the following directory:

   ```
   <MDM Hub installation directory>/hub/server/bin
   ```

4. Run the MDM Hub Master Database upgrade script with the migration script command and the migration properties. For a list of properties, see "Migration Properties" on page 176.
   For example, you can run one of the following commands:
   - On UNIX.

     ```
     sip_ant.sh migrate-avos-sa-tasks -Davos.jdbc.database.password=!!cmx!!
     -Davos.ws.protocol=http -Davos.ws.host=localhost -Davos.ws.port=8080 -
     Davos.ws.pagesize=100
     -Davos.ws.trusted.username=avos -Davos.ws.trusted.password=avos -
     Davos.hub.username=admin
     ```

   - On Windows.

     ```
     sip_ant.bat migrate-avos-sa-tasks -Davos.jdbc.database.password=!!cmx!!
     -Davos.ws.protocol=http -Davos.ws.host=localhost -Davos.ws.port=8080 -
     Davos.ws.pagesize=100
     -Davos.ws.trusted.username=avos -Davos.ws.trusted.password=avos -
     Davos.hub.username=admin
     ```

APPENDIX D

Configuring Metadata Caching

This appendix includes the following topic:

Configuring Metadata Caching (Optional)

The metadata caches manage items such as data objects, repository objects, and search tokens. The MDM Hub uses Infinispan for metadata caching. Infinispan is installed with the Hub Server. For the caches that are used by the Hub Server, the Infinispan configuration file contains default attribute values.

In version 10.1 and earlier, the MDM Hub used JBoss Cache for metadata caching. After you upgrade from one of these versions, the MDM Hub Server uses the Infinispan configuration file instead of the JBoss Cache configuration file.

If the JBoss Cache configuration file was edited in the previous version of MDM Hub, you might need to edit the Infinispan configuration file. It depends on why the file was edited.

Network policy

If the JBoss Cache file was edited to work around your organization's network policy, update the Infinispan file and the jgroups* file with the same policy changes.

Performance

If the JBoss Cache file was edited to improve cache performance, first try running the MDM Hub with the default Infinispan values. If you experience performance issues, copy the changed values from the JBoss Cache configuration file to the Infinispan configuration file. If you still experience performance issues, familiarize yourself with Infinispan and adjust the values to better suit your environment.

Infinispan Attributes

The following table summarizes default Infinispan attribute values and indicates how the attributes map to the former JBoss attribute:

Infinispan Element and Attribute	Default Value	Description	JBoss Attribute
`locking acquire-timeout`	`60000`	Maximum time during which the Hub Server can try to acquire a lock.	`lockAcquisitionTimeout`
`transaction stop-timeout`	`30000`	When a cache stops, this attribute sets the maximum time that Infinispan waits while the Hub Server finishes remote and local transactions.	`sync replTimeout`
`transport cluster`	`infinispan-cluster`	Name for the underlying group communication cluster.	`clustering`
`transport stack`	`UDP`	Type of configuration: UDP or TCP. The configurations are defined in the `jgroups-udp.xml` file and the `jgroups-tcp.xml` file.	`jgroupsConfig`
`transport node-name`	`$node$`	Name of the current node. The Hub Server sets this attribute. The node-name defaults to a combination of the host name and a random number. The number differentiates multiple nodes on the same host.	--
`transport machine`	`$machine$`	ID of the machine where the node runs. The Hub Server sets this attribute.	--
`expiration lifespan`	--	Maximum lifespan of a cache entry in milliseconds. When a cache entry exceeds its lifespan, the entry expires within the cluster. If you need to optimize performance, increase the lifespan for the following caches: DISABLE_WHEN_LOCK, DATA_OBJECTS, and REPOS_OBJECTS. For example, you can increase the lifespan from one hour (3600000) to one day (86400000). Each cache has its own default value for this attribute. To find the default values, open the `inifinspanConfig.xml` file.	`eviction timeToLive`
`expiration interval`	--	Maximum interval for checking the lifespan. If you need to optimize performance, increase the interval for the following caches: DISABLE_WHEN_LOCK, DATA_OBJECTS, and REPOS_OBJECTS. For example, you can increase the interval from five seconds (5000) to five minutes (300000). Each cache has its own default value for this attribute. To find the default values, open the `inifinspanConfig.xml` file.	`eviction timeToLive`

Editing Infinispan Attributes

To configure metadata caching attributes, edit the `infinispanConfig.xml` file for the Hub Server. For help with the Infinispan configuration, see the Infinispan documentation.

Note: The Process Server also has an Infinispan configuration file. The default attribute values should be sufficient, however if you notice issues with the performance of the Process Server, you can fine-tune the attribute values.

1. Navigate to the following directory: `<MDM Hub installation directory>/hub/server/resources`
2. Make a backup copy of the following file: `infinispanConfig.xml`
3. Open the `infinispanConfig.xml` file and find the Infinispan version number, which appears in the `xsi:schemaLocation` attribute.
4. Review the documentation for the Infinispan version.

   Note: In the following URLs, substitute the version number wherever the path contains #.#.

   - To view the configuration schema, go to the URL that is contained in the `xsi:schemaLocation` attribute in the file.
   - To learn about the attributes, go to `https://docs.jboss.org/infinispan/<#.#.x>/configdocs/`
   - To learn about Infinispan, go to `http://infinispan.org/docs/<#.#.x>/` and select the "Frequently Asked Questions" link.
5. Edit the file and save it.

INDEX