

IBM InfoSphere Master Data Management
Version 11 Release 3

Upgrade Guide



IBM InfoSphere Master Data Management
Version 11 Release 3

Upgrade Guide



Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 79.

Edition Notice

This edition applies to version 11.3 of IBM InfoSphere Master Data Management and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables v

Figures vii

Upgrading from a previous version . . . 1

Upgrading from a previous version. 1

What to back up before upgrade. 1

Installing InfoSphere MDM 2

Upgrading from InfoSphere MDM version 11 . . 2

Upgrading from Initiate Master Data Service,
prior to version 11 21

Upgrading from InfoSphere MDM Server, prior
to version 11 39

Upgrading from InfoSphere MDM Server, prior to
version 11 59

Migrating customizations to work with OSGi . . 60

Deploying additions and extensions in OSGi . . 60

Upgrading the existing database for InfoSphere

MDM Server 61

Recreating ILOG rules. 77

Verifying your upgrade 78

Notices and trademarks 79

Index 85

Contacting IBM 87

Tables

1. Database upgrade worksheet.	25	3. Providing feedback to IBM	87
2. IBM resources.	87		

Figures

- | | | | | | |
|----|---|----|----|---|----|
| 1. | Upgrade overview: Initiate components in 11.3 | 22 | 3. | Upgrade overview: InfoSphere MDM Server | |
| 2. | Upgrade overview: InfoSphere MDM Server | | | components in 11.0 | 60 |
| | components in 11.0 | 40 | | | |

Upgrading from a previous version

You can upgrade to the most recent release of InfoSphere® MDM from the 11.0 version of InfoSphere MDM or from a pre-11.0 version of Initiate Master Data Service® or InfoSphere MDM Server.

About this task

If you are upgrading from an the 11.0 version of InfoSphere MDM, you can ignore the topics in this section about upgrading from InfoSphere MDM Server or IBM® Initiate®.

If you are upgrading an instance of InfoSphere MDM Server, the upgrade process requires that your current installation of InfoSphere MDM Server is version 8.5 or later.

If you are upgrading an instance of IBM Initiate, the upgrade process requires that your current installation of Initiate is version 9.0 or later.

Upgrading from a previous version

You can upgrade to the most recent release of InfoSphere MDM from the 11.0 version of InfoSphere MDM or from a pre-11.0 version of Initiate Master Data Service or InfoSphere MDM Server.

About this task

If you are upgrading from an the 11.0 version of InfoSphere MDM, you can ignore the topics in this section about upgrading from InfoSphere MDM Server or IBM Initiate.

If you are upgrading an instance of InfoSphere MDM Server, the upgrade process requires that your current installation of InfoSphere MDM Server is version 8.5 or later.

If you are upgrading an instance of IBM Initiate, the upgrade process requires that your current installation of Initiate is version 9.0 or later.

What to back up before upgrade

Before you upgrade, back up the key assets of your IBM Initiate or IBM InfoSphere MDM Server environment.

Because each installation is different, the documentation doesn't attempt to provide specific steps for backing up. Use your preferred archiving and backup tools.

- Database (including the user repository if it is stored in the database)
- LDAP repository (if users are stored outside of the installation)
- Log files
- Success and Input data files for the Initiate Broker components
- For InfoSphere MDM Server, the configuration settings within the CONFIGELEMNT database table

- For Initiate, the Workbench workspace. The workspace will include your party.imm data model file, customization artifacts, custom code, handlers, and so on.
- For Initiate, the configuration files in the conf directory

Installing InfoSphere MDM

For all upgrades, you must first install the version of InfoSphere MDM to which you want to upgrade. Use the custom installation deployment type and create a new database. Do not reuse an existing database.

About this task

Note: As part of the upgrade process, you install the latest version of the InfoSphere MDM operational server with the new database schema. However, that schema is abandoned when the operational server is re-pointed to the upgraded schema from your current installation. For example, do not install the new operational server into the schema for the previous version until you have upgraded the schema.

Procedure

1. Shut down any running instances of the prior release: servers, engine, applications, and so forth.
2. Install the latest version of InfoSphere MDM. Use the custom installation deployment type. For installation instructions, see the installation documentation.
3. During the installation, create a new database for the new version of the operational server.

Related concepts:

 Overview of OSGi changes to the MDM Workbench

 Introduction to migrating InfoSphere MDM customizations to work with OSGi

Related information:

 Installing a custom installation

Upgrading from InfoSphere MDM version 11

A set of steps describes the process of upgrading from version 11 of InfoSphere MDM.

About this task

This section provides instructions for upgrading the database and enterprise application of InfoSphere MDM version 11.

Note: Before you review any other documentation or upgrade the product, read the release notes and the installation instructions for the IBM InfoSphere Master Data Management product.

Upgrading your MDM environment often requires assistance from members of your own organization as well as IBM Services or an IBM consulting partner. Do not upgrade your implementation without first identifying the team and relying on guidance from that team.

Upgrading the existing database for InfoSphere MDM

To upgrade the existing database for IBM InfoSphere MDM, upgrade the database objects, data sources, and tablespaces.

About this task

The upgrade applies only to the original database structure and data. If you made any customizations or extensions, the changes might be deleted or changed by the upgrade.

Upgrading the virtual MDM tables:

You run madconfig scripts to upgrade the existing virtual MDM tables to the new version of InfoSphere MDM.

Procedure

1. At a command-line prompt, go to the scripts directory of the new installation: *MAD_INSTALL_HOME/mds/scripts*.
2. Run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password

3. Run the madconfig update_instance_datasource command again, but this time from the native.war directory within the WebSphere Application Server. First, navigate to *WAS_PROFILE_HOME\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\scripts* directory. Then run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password

Running the madconfig target from this second location updates the database connection that the operational server and the MDM Workbench jobs use.

4. Return to the *WAS_PROFILE_HOME\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\sql* directory.

5. Update the `mpihub.ddl` file to ensure that it contains any IDSs (implementation defined segments). You can copy any IDSs from the 11.0 version of the `mpihub.ddl` file.
6. Update the `mpihub.sto` file so that it contains the new database information. The installer creates a version of the `mpihub.sto` file, which points to the new database that you created. However, because you want to use your upgraded Initiate database rather than a new database, update the `mpihub.sto` file so that the values apply to the Initiate database.

Note: For Microsoft SQL Server, you must update the filegroup name rather than the database name.

7. Perform the same updates to copies of the `mpihub.ddl` and `mpihub.sto` files in the `MDM_INSTALL_HOME/mds/sql` directory as well. Begin by navigating to that directory.
8. Update the `mpihub.ddl` file to ensure that it contains any IDSs (implementation defined segments). As before, you can copy any IDSs from the 11.0 version of the `mpihub.ddl` file.
9. Update the `mpihub.sto` file so that it contains the new database information. As mentioned before, the installer creates a version of the `mpihub.sto` file, which points to the new database that you created. However, because you want to use your upgraded Initiate database rather than a new database, update the `mpihub.sto` file so that the values apply to the Initiate database.

Note: As before, for Microsoft SQL Server, you must update the filegroup name rather than the database name.

10. Run the `madconfig upgrade_datasource` command:
 - For Microsoft Windows: `madconfig upgrade_datasource`
 - For Linux and UNIX: `madconfig.sh upgrade_datasource`

The `madconfig` target executes the statements to update the datasource.

Results

The scripts update the following configuration files within the `WAS_PROFILE_HOME/conf` directory:

- `com.ibm.mdm.mds.jni.cfg`
- `com.ibm.mdm.mds.jdbc.cfg`

If the scripts do not update the configuration files, make the updates manually.

Creating and updating database objects:

To set up the database for the new version within the previous version of the database, run SQL scripts to create and update tables, indexes, foreign keys, and triggers.

Upgrading an existing DB2 database for UNIX or Linux:

Follow the steps to upgrade to the latest version of for DB2® on UNIX or Linux.

Before you begin

Before you begin, make sure that you read:


- The information on gold data deletes and data upgrade scripts in the topic about upgrading the existing database for IBM InfoSphere MDM Server. See the link at the end of this topic.
- The notes on table spaces in the Manual installation of the physical MDM database topic in the IBM InfoSphere Master Data Management Standard and Advanced Editions Installation Guide

About this task

You have two options for upgrading your existing DB2 database for UNIX or Linux:

- Automatic upgrade - For the automatic upgrade, you run a madconfig target that finds and runs a set of individual database upgrade scripts delivered with the current release. The madconfig target runs the version-specific scripts in sequence until the database is full upgraded to the current version. For more information, see the link at the end of this topic.
- Manual upgrade - For the manual upgrade, you run the individual version-specific scripts manually. For detailed steps, see the link at the end of this topic.

Related concepts:


 Manually installing the madconfig utility

 madconfig utility targets for InfoSphere MDM

Related tasks:

“Upgrading the existing database for InfoSphere MDM Server” on page 41

Related information:

 Manual installation of the physical MDM database

Automatically upgrading the database on DB2 for UNIX or Linux:

You can run madconfig upgrade-mdm-ae-db to perform an automatic upgrade of the MDM content on DB2 for UNIX or Linux.

For IBM InfoSphere MDM Advanced Edition, a madconfig target can perform an automatic database upgrade from any version of InfoSphere MDM Server from 8.5 onward. You must supply the version information for your current installation of InfoSphere MDM. The madconfig utility can then run the database upgrade steps required to increment from that version to the latest version of InfoSphere MDM. The madconfig utility is run from the `MDM_INSTALL_HOME/mds/scripts` directory. When you run madconfig upgrade-mdm-ae-db target, the utility runs upgrade scripts in the `MDM_INSTALL_HOME/database/Upgrade/` directory, which contains the separate directories for each previous version. The folders contain SQL scripts for performing the upgrade for that version. The madconfig target runs the required scripts in sequence for each version.

With some MDM Server installations prior to version 10, the Event Manager data could be stored in a set of tables separate from the MDM content database. If your installation has Event Manager data stored in separate tables, merge the tables into the main schema before proceeding with the upgrade.

Also before proceeding with the upgrade you must manually edit the `upgradepaths.properties` file to provide values that the madconfig target

requires, including the workspace directory, preferred language, preferred industry, database type, database credentials, and so on. Look for the properties file in the `MDM_INSTALL_HOME/mds/scripts` directory.

Editing the upgrade properties file:

Manually edit the `UpgradePaths.properties` properties file to allow the ant script to perform the database upgrade.

About this task

The `UpgradePaths.properties` properties file enables the `madconfig` target. It contains user information, connection information, and information about tablespace correspondences.

Procedure

1. Navigate to the `MDM_INSTALL_HOME/mds/scripts` directory.
2. Open the `UpgradePaths.properties` file in your preferred text editor.
3. See the instructions in the comments within the file itself for more information about each parameter. The specific parameter names are listed here:
 - **`mdm.upgrade.basedir`** - The directory you specify here will match the directory you navigated to in Step 1.
 - **`rdm.upgrade.basedir`** - This parameter is commented out by default. Do not uncomment or change the parameter. It is available to enable future functionality.
 - **`mdm.ae.locales`** - When it was installed, your current installation of InfoSphere MDM was configured to use one or more locales. Specify the same locales here that were specified with the original installation.
 - **`mdm.ae.industry`** - When it was installed, your current installation of InfoSphere MDM was configured to include one or more industry data sets. The values can be banking, financial, telco, or manufacturing. Use the same value or values that were used with the original installation.
 - **`mdm.ae.triggers`** - When it was installed, your current installation of InfoSphere MDM was configured to use either simple triggers, compound triggers, or no triggers. Refer to your installation notes for the types of triggers that are installed in MDM and enter the same value for this parameter. The type you specify here must match what is already configured for MDM.
 - **`mdm.ae.delete.triggers`** - As with **`mdm.ae.triggers`**, choose compound, simple, or none depending on how your current MDM installation is configured.
 - **`db.type`** - The current release supports DB2.
 - **`db.subtype`** - The current release supports Standard.
 - **`db.driver.class`**
 - **`db.driver.path`**
 - **`db.jdbc.url`**
 - **`db.uid`** - The user ID that is used to connect to the database and to run the upgrade scripts. Ensure that it has sufficient privileges to alter schemas and perform other necessary operations.
 - **`db.pwd`** - The password that corresponds to the user ID you provided.
 - **`db.name`** - The name of the database you to which the scripts will connect, for example, `MDMDB`.

- **starting.version.number** - Enter the version of MDM that you will be upgrading. InfoSphere MDM Version 11.3 includes the scripts that are necessary to upgrade from Version 8.5.0. The correct version numbers to use for the **starting.version.number** parameter are 8.5.0, 9.0.1, 9.0.2, 10.0.0, 10.0.1, and 11.0.0.
- **upgrade.path.8.5.0** - Do not edit this value.
- **upgrade.path.9.0.1** - Do not edit this value.
- **upgrade.path.9.0.2** - Do not edit this value.
- **upgrade.path.10.0.0** - Do not edit this value.
- **upgrade.path.10.1.0** - Do not edit this value.
- **upgrade.path.11.0.0** - Do not edit this value.
- **SCHEMA** - Some table names in the upgrade scripts are qualified with the **SCHEMA** name or the **DBUSER** name. The two terms mean the same thing. Some older scripts used the term **DBUSER** while others used the term **SCHEMA**.
- **DBUSER** - See previous note.
- **TABLE_SPACE** - Enter name of the User Data table space name. Provide the value that was used when the MDM database was first created. Refer to the database. The default pool size is 8K.
- **INDEX_SPACE** - Enter the name of Index Table space. Provide the value that was used when the MDM database was first created. Refer to the database.
- **LONG_SPACE** - Enter the name of the log data table space. Provide the value that was used when the MDM database was first created. Refer to the database.
- **TABLE_SPPMD** - Enter the name of the tablespace to contain the new Probabilistic Matching Engine tables. The upgrade process will create the tablespace for you.
- **TABLE_SPPMI** - Enter the name of the tablespace to contain the new Probabilistic Matching Engine indices. The upgrade process will create the tablespace for you.
- **DB_OS** - Enter the desired default value for /IBM/DWLCommonServices/DataBase/OS
- **DB_TYPE** - The current release supports DB2.

Note: The **DB_TYPE** element is part of the configuration of MDM at runtime. The **db.type** parameter mentioned previously identifies the folder in the upgrade scripts for the database you want to upgrade. The two parameters should always refer to the same type of database, for example "Oracle" for Oracle, and "DB2" for DB2, but the specific syntax of each parameter might differ.

- **DB_VERSION** - Enter the desired default value for /IBM/DWLCommonServices/DataBase/version. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **DEFAULT_TIMEZONE** - Enter the desired default value for /IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **TIMEZONE_ENABLED** - The current release supports a value of false.
- **NLS_SYS_LANG** - Enter the desired default value for /IBM/DWLCommonServices/NLS/system_Default_Data_Locale. See the link at the end of this topic for more information about the elements in the Configuration and Management component.

- **DSRC_CODE** - Enter desired default value for /IBM/ThirdPartyAdapters/EAS/dsrcCode. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **EXCLUSIVE_CIENT_SYSTEMS** - Enter desired default value for /IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem. See the link at the end of this topic for more information about the elements in the Configuration and Management component.

4. Save and close the file.

Related tasks:

 Elements in the Configuration and Management component

Running the madconfig target for database upgrade:

After you have completed any prerequisite steps, run the madconfig upgrade-mdm-ae-db target to perform the automatic database upgrade.

About this task

For IBM InfoSphere MDM Advanced Edition, a madconfig target can perform an automatic database upgrade from any version of InfoSphere MDM Server from 8.5 onward. The madconfig target detects your current version of InfoSphere MDM and runs the database upgrade steps required to increment from that version to the latest version of InfoSphere MDM. Although you run the madconfig target from the *MDM_INSTALL_HOME*/mds/scripts directory, the database upgrade scripts that it runs are in the *MDM_INSTALL_HOME*/database/Upgrade/ directory.


Procedure

1. At a command-prompt, navigate to the *MDM_INSTALL_HOME*/mds/scripts directory.
2. Run the madconfig target:
 - For Microsoft Windows: `madconfig upgrade-mdm-ae-db -l logfilename`
 - For Linux and UNIX: `madconfig.sh upgrade-mdm-ae-db -l logfilename`

Note: You can use the *logfilename* argument to specify a location and name for the log file. If you do not specify a *logfilename*, the log messages are written out to the console.

3. To review the process, see the log file created by the madconfig target.

Related concepts:

 Manually installing the madconfig utility

 madconfig utility targets for InfoSphere MDM

Viewing a log of the upgrade process:

As the ant script runs, it directs all log information into a single file.

About this task

If you specified a log file name and location when you ran madconfig upgrade-mdm-ae-db, look for the log file in the directory you indicated.

Procedure

1. Navigate to the location you specified for the *logfile*name.
2. Open the file.

Enabling non-case-sensitive search capability:

The automatic upgrade process does not include an option to enable non-case-sensitive search for DB2. If you want to have the ability to do non-case-sensitive search, you can manually enable the capability after you have finished the automatic upgrade.

Procedure

1. Navigate to the *MDM_INSTALL_HOME/database/Full/DB2/Standard/ddl* directory.
2. Run the following script: `db2 -tvf Insensitive_search_enabled.sql -l <log_file_name>`

Manually upgrading the database on DB2 from 11.0 to 11.3:

You can use a set of scripts to do a manual upgrade of the MDM database on DB2 from 11.0 to 11.3.

Procedure

1. Navigate to the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl* directory.
2. In the *ImpReqDataCfgMgr.sql* and *QryUpdCfgMgrData.sql* scripts, replace the `<SCHEMA>` placeholder with a schema name or owner with necessary privileges. This value must be in uppercase.
3. In the *QryUpdCfgMgrData.sql* script, replace the following tokens with values that are appropriate to your own installation:
 - **DB_OS** - Enter the desired default value for */IBM/DWLCommonServices/Database/OS*.
 - **DB_TYPE** - Use the value *DB2*.
 - **DB_VERSION** - Enter the desired default value for */IBM/DWLCommonServices/Database/version*
 - **DEFAULT_TIMEZONE** - Enter the desired default value for */IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone*
 - **TIMEZONE_ENABLED** - Use the value *false*.
 - **NLS_SYS_LANG** - Enter the desired default value for */IBM/DWLCommonServices/NLS/system_Default_Data_Locale*
 - **DSRC_CODE** - Enter desired default value for */IBM/ThirdPartyAdapters/EAS/dsrcCode*
 - **EXCLUSIVE_CIENT_SYSTEMS** - Enter desired default value for */IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem*
4. Connect to the database that you are upgrading.
5. Replace the `<SCHEMA>` tag in the scripts in the following folders with a schema name or owner, in uppercase. *CODE_LANG* in the folder name is the language that is currently installed. If you are loading English data only, set *CODE_LANG* to *en*. Otherwise, set *CODE_LANG* to the language code of the language you are loading.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_req_CODE_LANG/data*
 - *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_CODE_LANG/data-industry*
 - *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_opt_CODE_LANG/data*
6. From the command line, to load the common data, English code table data, and configure the table in the language that you selected, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_req_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The `insert.sql` script inserts new data
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` deletes obsolete table data.
 7. From the command line, to load the English industry data, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_en/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The `insert.sql` script inserts new data
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` deletes obsolete table data.
 8. If you are loading industry data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_CODE_LANG/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The `insert.sql` script inserts new data
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` deletes obsolete table data.
 9. If you are loading code table data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_opt_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The `insert.sql` script inserts new data
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` deletes obsolete table data.
 10. Repeat the previous two steps for each non-English language that is installed.
- Note:** If you have multiple deployments, before you run the following script, change the SQL clause `DEPLOYMENT_ID <> 1000` to `DEPLOYMENT_ID = DEPLOYMENT_ID` where the *DEPLOYMENT_ID* is the ID of the instance you want to upgrade.
11. From the command line, run the command: `db2 -tvf ImpReqDataCfgMgr.sql -l log_file_name`.
 12. From the command line, run the command: `db2 -tvf QryUpdCfgMgrData.sql -l log_file_name`.

Related tasks:

“Manually upgrading the database on DB2” on page 48

Upgrading an existing DB2 database for z/OS:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for DB2 on z/OS®. For information about upgrading the database for previous releases, see the links at the end of this topic.

Procedure

1. To connect to DB2 for OS/390® and z/OS server from a DB2 Linux, UNIX, or Windows server, DB2 Connect™ must be installed on the server you are connecting from. Enterprise editions contain DB2 Connect. To connect to the database, enter the following information:
 - Host name or IP address
 - DB2 port (default = 446)
 - Database name
 - Subsystem location name
 - User ID and password
2. On the client server, catalog the database by running the following commands in this order:
 - a. `db2 catalog tcpip node YOUR_NODE_NAME remote Z/OS_HOSTNAME/IP_ADDRESS server Z/OS_PORT`
 - b. `db2 catalog db YOUR_DB_NAME at node YOUR_NODE_NAME authentication dcs`
 - c. `db2 catalog dcs db YOUR_DB_NAME as Z/OS_LOCATION_NAME`
3. If you are using this user ID to connect to the DB2 subsystem for the first time, rebind the packages for the database by following either of these steps:
 - a. Rebind all the packages by running the command: `db2rbind YOUR_DB_NAME all /u USER_NAME /p PASSWORD /r any`
 - b. Rebind individual packages by running the command: `db2 bind package_name`
4. Navigate to the `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1` directory.
5. Change the placeholder tags in the all of the scripts:
 - Set <SCHEMA> to a schema name or owner in uppercase.
 - Set <USER_ACCOUNT> to an account with user authority.
6. In the folder name, `CODE_LANG` is the language that is currently installed:







Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_req_CODE_LANG/data`
- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_CODE_LANG/data-industry`
- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_opt_CODE_LANG/data`


If you are loading English data only, set `CODE_LANG` to `en`. Otherwise, set `CODE_LANG` to the language code of the language you are loading.

7. To load the common data from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_req_en/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
8. To load the English industry data from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_en/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
9. If you are loading industry data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_CODE_LANG/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
10. If you are loading code table data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_opt_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
11. Repeat the previous step for each non-English language that is installed.
12. From the command line, run the command: `db2 -tvf ImpReqDataCfgMgr_zos.sql -l log_file_name`.
13. From the command line, run the command: `db2 -tvf QryUpdCfgMgrData_zos.sql -l log_file_name`.

Related tasks:

-  Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
-  Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
-  Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
-  Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0

Related information:

-  Manual installation of the physical MDM database

Upgrading an existing Oracle database:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for Oracle. For information about upgrading the database for previous releases, see the links at the end of this topic.

Before you begin

Read the content for the link at the end of this topic about manual installation of the database.

For Oracle, the use of table spaces for data, indexes and LOBs, such as CLOBs or XML, to improve database performance is supported. Look for the following placeholders in the scripts: <TABLE_SPACE>, <INDEX_SPACE>, and <LONG_SPACE>.

Procedure

1. Replace the placeholder <SCHEMA> in the scripts in the following folders with a schema name or owner. Ensure that the name or owner is in uppercase. *CODE_LANG* in the folder name is the language that is currently installed.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_req_*CODE_LANG*/data
 - *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_*CODE_LANG*/data-*industry*
 - *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_opt_*CODE_LANG*/data
2. To load the common data, to load the English code table data, and to configure the table in the language that you selected, run the following script in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_req_*CODE_LANG*/data directory from the command line. Use the command:
sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @<script_name> >> <log_file_name>
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.

If you are loading English data only, set *CODE_LANG* to en. Otherwise, set *CODE_LANG* to the language code of the language you are loading.

3. To load the English industry data, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_en/data-*industry* directory from the command line. Use the command: sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @<scriptname> >> <log_file_name>
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
4. If you are loading data for a language other than English, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_<*CODE_LANG*>/data-*industry* directory from the command line:
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.

- c. The delete.sql script deletes obsolete table data.
5. If you are loading code table data for a language other than English, run the scripts in the `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_opt_CODE_LANG/data` directory from the command line:

```
sqlplus <SCHEMA> /<SCHEMAPASSWORD>@<DBNAME> @<scriptname> >>
<log_file_name>
```





 - The insert.sql script inserts new data.
 - The update.sql script updates existing table data.
 - The delete.sql script deletes obsolete table data.
6. Repeat the previous two steps for each non-English language that is installed.
7. Navigate to the `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/` directory.
8. Run the following command:

```
sqlplus <SCHEMA> /<SCHEMAPASSWORD>@<DBNAME>
@ImpReqDataCfgMgr_ora.script >> <log_file_name>
```
9. Run the following command:


```
sqlplus <SCHEMA> /<SCHEMAPASSWORD>@<DBNAME>
@QryUpdCfgMgrData_ora.sql >> <log_file_name>
```

 The `QryUpdCfgMgrData_ora.sql` script populates the Configuration Manager values based on previous InfoSphere MDM version 11 deployment values.

Related tasks:

-  Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
-  Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
-  Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
-  Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0

Related information:

-  Manual installation of the physical MDM database

Upgrading an existing Microsoft SQL Server database:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for Microsoft SQL Server.

Before you begin

Read the content for the link at the end of this topic about manual installation of the database.

Run the following command from the command line:

```
sqlcmd -S <server Name> -U
sa -P sapassword -i inputquery_file_name -o outputfile_name
```

Procedure

1. Replace the placeholder `<SCHEMA>` in the scripts in the following folders with a schema name or owner. Ensure that the name or owner is in uppercase. `CODE_LANG` in the folder name is the language that is currently installed.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_req_CODE_LANG/data*
 - *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_CODE_LANG/data-industry*
 - *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_opt_CODE_LANG/data*
2. To load the common data, to load the English code table data, and to configure the table in the language that you selected, run the following script in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_req_CODE_LANG/data* directory from the command line. Use the command: `sqlcmd -S <server Name> -U sa -P sapassword -i inputquery_file_name -o outputfile_name`
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.

If you are loading English data only, set *CODE_LANG* to en. Otherwise, set *CODE_LANG* to the language code of the language you are loading.
 3. To load the English industry data, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_en/data-industry* directory from the command line. Use the command: `sqlcmd -S <server Name> -U sa -P sapassword -i inputquery_file_name -o outputfile_name`
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
 4. If you are loading data for a language other than English, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_<CODE_LANG>/data-industry* directory from the command line. Use the command: `sqlcmd -S <server Name> -U sa -P sapassword -i inputquery_file_name -o outputfile_name`
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
 5. If you are loading code table data for a language other than English, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/data_opt_CODE_LANG/data* directory from the command line: `sqlcmd -S <server Name> -U sa -P sapassword -i inputquery_file_name -o outputfile_name`
 - The insert.sql script inserts new data.
 - The update.sql script updates existing table data.
 - The delete.sql script deletes obsolete table data.
 6. Repeat the previous two steps for each non-English language that is installed.
 7. Navigate to the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/SQLServer/Standard/ddl/* directory.
 8. Run the following command: `ImpReqDataCfgMgr.bat`
 9. Run the following command: `sqlcmd -S <server Name> -U sa -P sapassword -i QryUpdCfgMgrData.sql -o QryUpdCfgMgrData.log` The *QryUpdCfgMgrData.sql* script populates the Configuration Manager values based on previous deployment values.

Updating the instance datasource:

Run a madconfig target to update the instance datasource and provide connectivity information.

Procedure

1. At a command-line prompt, go to the scripts directory of the installation:
MDM_INSTALL_HOME/mds/scripts.
2. Run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig.bat update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password
- g. Database schema - You must supply a schema name if your database is DB2.

Note: In Windows, System DSN will show all the datasource names created. In Linux environments, the *MDM_INSTALL_HOME*/mds/odbc.ini file will show the datasource names.

3. When the script asks whether you want to re-create the existing data source, enter y.
4. The script prompts for the database schema. For IBM DB2 for z/OS, you can submit an alternative schema or enter nothing to use the default schema. For other database types, enter nothing.
5. Take a moment to verify that the settings you entered have been applied:
 - a. Navigate to *MDM_INSTALL_HOME*/mds/conf
 - b. Open the com.ibm.mdm.mds.jni.cfg configuration file and verify the settings.
 - c. Save and close the file.
 - d. Open the com.ibm.mdm.mds.jdbc.cfg configuration file and verify the settings.
 - e. Save and close the file.
6. Set the library path to point to *MDM_INSTALL_HOME*/mds/lib. An incorrect setting can result in a "madhub create" error when you run the bootstrap process. Use the library path variable that is appropriate for your operating system. To see the current setting, use the following commands:
 - Linux or Solaris - export LD_LIBRARY_PATH
 - AIX® - export \$libpath
 - Microsoft Windows - set PATH
7. Confirm that tablespaces are created. The process pre-pends the last three characters of your schema to "ZMDS". For example, if your schema is M5Z10M65, the database identifier will be M65ZMDS.
8. In the *MDM_INSTALL_HOME*/mds/sql directory, edit the mpihub.sto file so that the database identifier is correct for each tablespace.

9. From the *MDM_INSTALL_HOME*/mds/scripts directory, run the madconfig bootstrap_datasource command:
 - For Microsoft Windows: madconfig.bat bootstrap_datasource
 - For Linux and UNIX: madconfig.sh bootstrap_datasource
10. To make sure that everything stays in sync, run the madconfig update_instance_datasource command again, but this time from the native.war directory within the WebSphere® Application Server. First, navigate to *WAS_PROFILE_HOME*\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\scripts directory. Then run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password

Results

The first script updates the following configuration files within the *MDM_INSTALL_HOME*/mds/conf directory:

- com.ibm.mdm.mds.jni.cfg
- com.ibm.mdm.mds.jdbc.cfg

The second script creates the Initiate tables.

Updating the data sources:

As part of the upgrade process, edit the user, password, and other database information for the data sources.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to **Resources > JDBC > Data sources**.
3. Select the data source **DWLConfig**.
4. Edit the database information under **Common and required data source properties** to point to the upgraded database schema.
5. Click on **Custom properties** and edit the user and password of the upgraded database schema.
6. Also click on **JAAS - J2C authentication data** and edit all of the aliases in the list. (If you did not deploy using MQ, you do not need to edit the alias for Node01/application-name-MQ-User.) For each alias, modify the user name and password.
7. Repeat steps 4, 5, and 6 for the data source **DWLCustomer**.
8. Select the data source **MDM**.

9. Click on **Customer properties** to edit the database information of the upgraded database.

Updating the messaging engine:

Update the WebSphere Embedded Messaging engine as part of the upgrade process.

Procedure

1. Log on to the WebSphere Application Server Administration Console.
2. Navigate to **Service integration > Buses > MDM.SIB.application-name > Messaging Engine > node-name.application-name-MDM.SIB.application-name > Message Store**.
3. Edit the **Schema name** field.
4. Stop the application server.
5. Drop all of the messaging engine tables. The messaging engine table names all begin with **SIB**:
 - SIB000
 - SIB001
 - SIB002
 - SIBCLASSMAP
 - SIBKEYS
 - SIBLISTING
 - SIBOWNER
 - SIBOWNER0
 - SIBXACTS
6. Restart the application server.

Updating the install name in the appsoftware table:

Update the name of the installation within the appsoftware table.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to the Business Level Application for InfoSphere MDM: **Application > Application Types > Business-level applications**. Then look for the application with the description **MDMBLA**.
3. Open the composition unit.
4. Open the **OSGi application console**.
5. Note the bundle name: `com.ibm.mdm..hub.server-XXXXXXX`.
6. Log into the MDM database.
7. Select * from `<SCHEMA>.APPSOFTWARE` WHERE VERSION = '11.3.0'. After you run the SQL statement, note the name of the application in the NAME field. It will say **com.ibm.mdm.hub.server**.
8. Update the appsoftware table to add `-[name]` to the `com.ibm.mdm.hub.server` record. For example, if `XXXXXXX` is `E001`, then to update to InfoSphere MDM Version 11.3:
 - DB2: `update appsoftware set NAME='com.ibm.mdm.hub.server-E001', last_update_dt = current timestamp where APPLICATION_ID = 1001;`

- Oracle: update appsoftware set NAME='com.ibm.mdm.hub.server-E001', last_update_dt = sysdate where APPLICATION_ID = 1001;
- Microsoft SQL Server: insert into APPSOFTWARE values (10000,'com.ibm.mdm.hub.server-E001', '11.3.0', CURRENT_TIMESTAMP, 'unknown');

Migrating users and groups:

The process for migrating users and groups for InfoSphere MDM version 11.0 to version 11.3 depends on how you store users and groups for your current implementation of the WebSphere Application Server.

About this task

The options for storing users and groups include LDAP, local OS, federated repositories, or custom solutions. Because each implementation is different, the documentation doesn't attempt to offer complete instructions. See the documentation for your specific storage scheme.

Recreating ILOG rules:

The new OSGi architecture means that the embedded ILOG[®] engine is no longer available. Instead the ILOG engine (now called ODM) can be called remotely. Recreate existing ILOG rules as ODM rules.

About this task

For information about creating ODM rules, follow the link at the end of this topic.

Procedure

To determine whether you use ILOG rules with your previous version of InfoSphere MDM Server, run the following scripts.

1. For any ILOG-based Behavioral Extensions rules:

```
SELECT * FROM EXTENSIONSET
WHERE (JAVA_CLASS_NAME IS NULL OR JAVA_CLASS_NAME = '')
AND (RULE_SET_NAME IS NOT NULL OR RULE_SET_NAME != '')
```

2. For any ILOG-based External Rules:

```
SELECT * FROM RULEENGINEIMPL
WHERE EXT_RULE_IMPL_ID
IN (SELECT EXT_RULE_IMPL_ID FROM EXTRULEIMPLEM
WHERE EXT_RULE_TP_CODE='R')
```

Migrating configuration settings

Migrate customized values from key configuration files so that they are available within your new InfoSphere MDM installation.

About this task

Several components of IBM InfoSphere MDM Standard Edition use configuration files to control certain behaviors. The installer for the new version of InfoSphere MDM does not automatically migrate any custom configuration values from your previous installation to your new installation. The following steps describe migrating customized configuration values for the Entity Manager, Task Manager, Event Manager, Tag Manager, and for Search Synchronization.

Procedure

1. Within your InfoSphere MDM installation , navigate to the `WAS_PROFILE_HOME/conf` directory.
2. Open the `com.ibm.mdm.mds.entity.manager.cfg` file.
3. Migrate any custom configuration values to the `com.ibm.mdm.mds.entity.manager.cfg` configuration file within your new InfoSphere MDM installation: `WAS_PROFILE_HOME/conf/com.ibm.mdm.mds.entity.manager.cfg`.
4. Repeat the migration steps for the following configuration files:
 - `com.ibm.mdm.mds.event.manager.cfg`
 - `com.ibm.mdm.mds.task.manager.cfg`
 - `com.ibm.mdm.mds.tag.manager.cfg`
 - `com.ibm.mdm.mds.search.synchronization.cfg`

Note: You do not need to migrate the following files. They were updated automatically when you ran `madconfig update_instance_datasource` or they contain only path information that the installation mechanism for version 11.3 has already populated correctly:

- `com.ibm.mdm.mds.jdbc.cfg`
- `com.ibm.mdm.mds.jni.cfg` `odbc.ini`
- `com.ibm.mdm.mds.job.manager.cfg`
- `com.ibm.mdm.mds.search.manager.cfg`

Upgrading clients and other components

You must install the 11.3 clients and set up their connections to the operational server.

About this task

If you use other MDM products, such as Inspector, Web Reports, or Policy Monitoring, install the Version 11.3 editions of each client and connect to the operational server. For more detail about security, see the links at the end of this topic.

Also see the links for details about upgrading IHE and other components and features.

Finalizing and verifying your upgrade

You finish the upgrade process by finishing some final tasks and verifying that the new system is functional.

Procedure

1. Verify that the new InfoSphere MDM operational server can access the upgraded database.
2. Start the operational server.
 - a. Open IBM WebSphere Application Server Administrative Console.
 - b. Select **Applications > Application Types > Business-level applications**.
 - c. On the Business-level applications page, select your application. A red X in the **Status** field indicates the application is stopped. A green arrow indicates that the application is running.
 - d. Click **Start**.

3. Verify that the new operational server starts with no errors in the log.
4. If you have custom applications that access the engine using API calls, be sure to update the `madapi.jar` in the classpath of the server where the API calls are made from. Note that as of InfoSphere MDM version 11.0, the API includes the removal of the direct TCP port. All API calls are now routed through WebSphere HTTP ports.
5. If you use policy monitoring, migrate its configuration files and update its database and tables. For instructions, see the policy monitoring documentation.
6. If you use one or more of the various messaging brokers, convert their configuration files. For instructions, see the broker documentation.
7. If you use the IHE Accelerator project, upgrade its components. For instructions, see the IHE documentation.
8. Run InfoSphere MDM Inspector to confirm that your data is visible and available.
9. For optional components, run their user interfaces to test their functionality.

Upgrading from Initiate Master Data Service, prior to version 11

A set of steps describes the process of upgrading from Initiate Master Data Service to InfoSphere MDM, Version 11.3.

About this task

Upgrading your MDM environment often requires assistance from members of your own organization as well as IBM Services or an IBM consulting partner. Do not upgrade your implementation without first identifying the team and relying on guidance from that team.

The version 11.0 release of InfoSphere MDM represented several large-scale architecture changes to the Initiate Master Data Service product. Most significantly, the engine and its services began to run exclusively within an application server. The application server provides infrastructure (for component-to-component communication, authentication, and logging) that was previously managed by the Initiate application itself. WebLogic and Tomcat application servers are no longer supported.

In addition, because the InfoSphere MDM product combines the Initiate Master Data Service product with the InfoSphere MDM Server product, it creates a default set of table spaces and shared schema definitions designed to accommodate the legacy needs of both products. For that reason, the upgrade process allows you to update your Version 10.1 database and then point to that database from your Version 11.3 installation.

If you are prompted that you are missing "add" xeia records and asked whether you want to insert them, choose yes. The installation process assumes that these records are inserted prior to the upgrade to Version 11.3.

A third architectural change involves LDAP. The Initiate Master Data Service product included an embedded LDAP directory server. By contrast, the InfoSphere MDM product takes advantage of the LDAP directory server available within the application server. Initiate Master Data Service customers using the embedded LDAP server will have to export LDAP content and import it either into the application server LDAP directory server or into an external LDAP directory server.

Lastly, because of new security requirements, unsecured context is no longer supported. You must modify custom code to include credentials in order to create context pools or contexts. Until it is updated, existing code will not compile against the new API JAR file.

The following overview diagram conceptually shows the upgrade process for the various components, where the grey boxes show components that are taken offline after the upgrade:

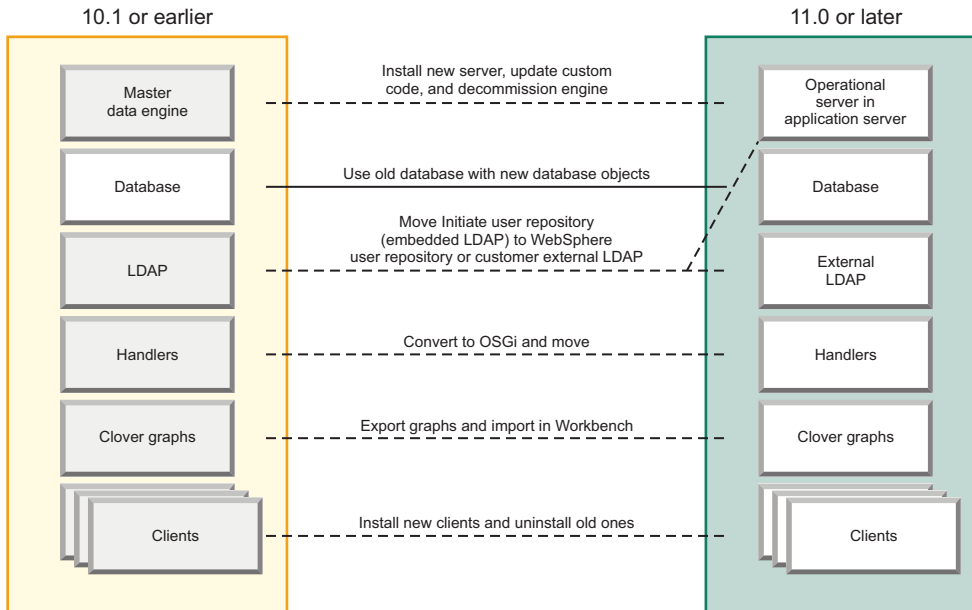


Figure 1. Upgrade overview: Initiate components in 11.3

Related concepts:

 [Architecture and concepts](#)

Updating custom code for new security requirements

You must modify custom code to include credentials to create contexts. Until it is updated, existing code does not compile against the new API JAR file.

About this task

This release of Standard and Advanced Editions makes improvements to the security for the C++ and Java™ APIs. The context constructors that previously did not require credentials must now include credentials to create contexts. Update any existing code to compile against the new API JAR file.

You must pick a new constructor for your custom code. The constructor can be similar to what you're using already, but it must add authentication credentials. For example, if you have code that creates a context and passes in the host and port, pick the constructor that adds authentication credentials with `UsrHead` or user name and password to the host and port. The credentials you specify need to be registered with the WebSphere Application Server and they must have permission to run the `virtualInteraction` transaction in the physical MDM. See the end of this topic for a link to information about configuring users and user groups for virtual MDM. When those two conditions are met, you can create Context objects and establish the initial connection from the client applications.

If you created custom code that manages user and group authentication, see the link at the end of this topic for more information about managing users.

If you have custom code that uses SSL, review the links about SSL at the end of this topic. Certain capabilities are no longer available with the current release. For example, you can no longer skip the certification validation step. Similarly, two-way SSL communication is no longer available. For these reasons, the following configuration variables are no longer relevant:

- **IBMInitiate.com.initiatesystems.sdk.seclib**
- **IBMInitiate.com.initiatesystems.sdk.sslversion**
- **IBMInitiate.com.initiatesystems.sdk.sslcertverify**





Similarly, the following VM options are no longer relevant:

- **-Djavax.net.ssl.keyStore**
- **-Djavax.net.ssl.keyStorePassword**
- **-Djavax.net.ssl.keyStoreType**

Related tasks:

“Upgrading Initiate clients and other components” on page 37

Related information:

-  [Configuring users and user groups for virtual MDM](#)
-  [Managing users](#)
-  [Configuring SSL](#)
-  [SSL security](#)

Updating the J2C authentication data:

As part of the upgrade process, configure the Java 2 Connector (J2C) authentication.

Procedure

1. Log on to the WebSphere Application Server Administration Console.
2. Navigate to **Security > Global Security**.
3. Expand **Java Authentication and Authorization Services > J2C authentication data**.
4. Edit the user ID and password of the upgraded database schema for the alias:
 - *application-name*/MDMServer-MDSDB-User-Config
 - *application-name*/DB-User
 - *application-name*/DB-User-Config

Upgrading the existing database for Initiate Master Data Service

When you install the new version of InfoSphere MDM, you must point to a new database. Do not provide the connection information for the existing database that you are upgrading.

About this task

The installation and upgrade process uses a temporary database for the new version of InfoSphere MDM, where it creates new, empty tables. Pointing to the existing database can create the risk that existing data will be overwritten.

After the installation is complete, stop the server and node in which you installed.

Conducting the pre-upgrade tasks:

Before you upgrade the Initiate database, there are a few tasks that you must complete.

About this task

Complete these tasks in the order presented.

Procedure

1. Review the 11.3 product tech notes to identify any new system requirements and data model changes.
2. Consult the individuals involved with your implementation to obtain guidelines and deployment-specific suggestions. These individuals can include members from your organization and IBM Services, or IBM consulting partners.
3. Review the MDM 11.3 database requirements. If necessary, upgrade your existing database to at least the minimum supported version required for version 11.3. For example, if your Initiate MDS version 9.0 is on DB2 9.5, then you would need to upgrade your DB2 to at least 10.1. For information about system requirements, see the link at the end of this topic.
4. Shut down your runtime instances. This includes engine, entity manager, and any inbound or outbound InfoSphere MDM Message Brokers instances. The Message Reader process can continue running and queueing up messages from the source systems until the brokers are upgraded.
5. Back up your operational data.
 - On each engine host, create a backup image of the instance home directories (*MAD_HOMEDIR*).
 - Create backup images of the data on each database server.
6. Review the operational server installation worksheets for the new version you are upgrading to and define your installation values.

Related tasks:

 Worksheets for installation and configuration

 System requirements for InfoSphere MDM

Upgrading the Initiate database:

After creating the initial runtime environment, you must upgrade the existing database. Before beginning, it is helpful to understand the upgrade process.

After reviewing this topic, use the database upgrade worksheet to define the values needed for the upgrade. You then use the **madconfig** utility to perform the upgrade.

Incremental upgrade steps. The incremental upgrade, or skip-level upgrade, processes the upgrade incrementally by version. The **madconfig upgrade_instance_datasource** and **madconfig upgrade_datasource** targets are used to perform this process. During the upgrade, a confirmation prompt is used to process through each applicable stage of the upgrade steps:

- From 9.0 to 9.2
- From 9.2 to 9.5

- From 9.5 to 9.7
- From 9.7 to 10.0
- From 10.0 to 10.1
- From 10.1 to 11.0
- From 11.0 to 11.3

Related to the upgrade steps, it is suggested that you confirm system version information and verify the database upgrade.

- Before running the **madconfig** utility, confirm that your system version information is set correctly to represent the version that the operational server runtime environment is running. Version information is found in the `mpi_syskey` table. The `keyval` column for the row when `keyname` equals “`ALIGNDEX_VERSION`” must accurately reflect the *major.minor* version of the product.
- After each upgrade step (for example, from 9.0 to 9.2, from 9.2 to 9.5, and so on), verify that the database is upgraded to the applicable version. Conduct this verification before allowing the **madconfig** utility to continue with the next upgrade step.

Checks in upgrade process. The upgrade script runs checks for these items:

- **Custom libraries.** If custom libraries are found, the script exits the process. You can bypass this check by deleting the custom library from the `mpi_libhead` table. One caution is that if the custom functionality has not been integrated into the mainline operational server in a subsequent release, the upgrade appears to run smoothly. However, the operational server might not function as expected, or it might fail to run. If the implementation requires the use of a custom library and that functionality is not part of the mainline product, perform the database upgrade manually.
- **EID script.** The script detects obsolete enterprise IDs. If the upgrade script finds obsolete EIDs, you are prompted to choose whether to discontinue the upgrade.

Upgrade limitations. Before you begin a skip-level upgrade by using `madconfig upgrade_instance_datasource` and `madconfig upgrade_datasource`, become familiar with these limitations:

- Accuracy of the upgraded data is not verified.
- Weights are not regenerated automatically.
- Data is not re-derived automatically.
- Backup and recoveries are not performed automatically.

Before starting the upgrade, complete the database upgrade worksheet.

Database upgrade worksheet:

Before beginning your upgrade, complete the database upgrade worksheet to define the required values.

Completing this worksheet in advance of the upgrade can help you quickly respond to upgrade prompts.

Table 1. Database upgrade worksheet

Configuration	Description	Your value
Instance name	Identify the name for the 11.3 operational server instance you just created.	

Table 1. Database upgrade worksheet (continued)

Configuration	Description	Your value
Rebuild the database tables in place	<p>Identify whether to rebuild the existing database tables by using the "in place" or "file system" method. The values determine how to run the upgrade process:</p> <ul style="list-style-type: none"> • y for the "in place" method. The "in place" method runs entirely within the database. It involves copying table data from existing tables; making the new release table changes; and then copying the data from the copied tables back to the updated tables. Important: If you select y, the transaction rollback log on the target database can grow large. As a result, make sure that you have sufficient disk space or disable transaction rollback during the upgrade process. <ul style="list-style-type: none"> – Pros: Work is managed on the database server rather than locally. – Cons: Requires enough free space on database server to accommodate the largest table that is being reloaded. • n for file system method. This method runs by dumping existing data to .unl files; making the new release table changes; and then reloading the .unl files to the existing database. The process requires adequate disk space to store the .unl files. <ul style="list-style-type: none"> – Pros: Offers the opportunity to use a faster method of table loading by using the native database loader. – Cons: Requires enough free space locally to accommodate the largest table that is being reloaded. 	
Reload tables with ODBC	<p>Attention: Skip defining this value unless opting to rebuild the database tables by using the file system method (as described in previous row).</p> <p>Identify whether to reload the existing database tables by using ODBC. The values are:</p> <ul style="list-style-type: none"> • y to reload the existing tables by using ODBC. This option requires that the data source is wire protocol compatible. <ul style="list-style-type: none"> – Pros: Does not require native tools. – Cons: Tends to be slow. • n means that the tables are reloaded by using the native database tools. <ul style="list-style-type: none"> – Pros: Fast. – Cons: Requires database client software to be installed and configured. 	
Confirmation	<p>Identify whether to proceed with upgrading the database from its current version to the next level.</p> <p>The madconfig utility detects which version is currently installed in the specified database. Depending on the existing version, you might need to upgrade incrementally in several steps by applying y on the confirmation prompts for each upgrade level. Continue applying y until you reach the final step to upgrade to 11.3.</p>	

Table 1. Database upgrade worksheet (continued)

Configuration	Description	Your value
Validate data dictionary	Identify whether to run the mpxdata utility to validate the data dictionary in the newly upgraded database. After each upgrade step (for example, from 9.7 to 10.0), the madconfig utility prompts you about validating the data dictionary.	
Continue	Identify whether to continue database upgrade process.	
Install database patch	Identify whether to install a patch release on the newly upgraded database. Depending on the existing version, you must upgrade incrementally in several steps by applying y on the confirmation prompts for each upgrade level until you reach the final step to upgrade to 11.3 patch version.	
Validate data dictionary	Identify whether to validate the data dictionary after the patch is installed.	

Upgrading Initiate tables:

You run madconfig scripts to upgrade the existing tables to the new version of InfoSphere MDM.

Procedure

1. At a command-line prompt, go to the scripts directory of the new installation: `MDM_INSTALL_HOME/mds/scripts`.
2. Run the madconfig update_instance_datasource command:
 - For Microsoft Windows: `madconfig update_instance_datasource`
 - For Linux and UNIX: `madconfig.sh update_instance_datasource`

The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password

3. To make sure that everything stays in sync, run the madconfig update_instance_datasource command again, but this time from the native.war directory within the WebSphere Application Server. First, navigate to `WAS_PROFILE_HOME\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\scripts` directory. Then run the madconfig update_instance_datasource command:
 - For Microsoft Windows: `madconfig update_instance_datasource`
 - For Linux and UNIX: `madconfig.sh update_instance_datasource`

The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:

- a. Database type
- b. Database host
- c. Database port

- d. Database name
 - e. Database user
 - f. Database password
4. Return to the *WAS_PROFILE_HOME*\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\sql directory.
 5. Update the *mpihub.ddl* file to ensure that it contains any IDSs (implementation defined segments). You can copy any IDSs from the 11.0 version of the *mpihub.ddl* file.
 6. Update the *mpihub.sto* file so that it contains the new database information. The installer creates a version of the *mpihub.sto* file, which points to the new database that you created. However, because you want to use your upgraded Initiate database rather than a new database, update the *mpihub.sto* file so that the values apply to the Initiate database.

Note: For Microsoft SQL Server, you must update the filegroup name rather than the database name.

7. Perform the same updates to copies of the *mpihub.ddl* and *mpihub.sto* files in the *MDM_INSTALL_HOME*/mds/sql directory as well. Begin by navigating to that directory.
8. Update the *mpihub.ddl* file to ensure that it contains any IDSs (implementation defined segments). As before, you can copy any IDSs from the 11.0 version of the *mpihub.ddl* file.
9. Update the *mpihub.sto* file so that it contains the new database information. As mentioned before, the installer creates a version of the *mpihub.sto* file, which points to the new database that you created. However, because you want to use your upgraded Initiate database rather than a new database, update the *mpihub.sto* file so that the values apply to the Initiate database.

Note: As before, for Microsoft SQL Server, you must update the filegroup name rather than the database name.

10. Run the *madconfig upgrade_datasource* command:
 - For Microsoft Windows: *madconfig upgrade_datasource*
 - For Linux and UNIX: *madconfig.sh upgrade_datasource*

The *madconfig* target executes the statements to update the datasource.

Results

1. The scripts update the following configuration files within the *WAS_PROFILE_HOME*/conf directory:
 - *com.ibm.mdm.mds.jni.cfg*
 - *com.ibm.mdm.mds.jdbc.cfg*

If the scripts do not update the configuration files, make the updates manually.

2. The scripts also update the following properties files within the *MDM_INSTALL_HOME*/properties and *WAS_PROFILE_HOME*\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\properties directories:
 - *install_native_engine_ear.properties*
 - *install_prop_file_jar.properties*
 - *install_server_config.properties*
 - *install_server_config.properties*
 - *install_server_config.properties*

- pre_check_db_version.properties
- verify_mds.properties

Again, if the scripts do not update the properties files, make the updates manually.

DB2 or Microsoft SQL Server: Adding new InfoSphere MDM database objects to the upgraded Initiate database:

As part of setting up the database, you must add some database objects from your new InfoSphere MDM installation to the existing Initiate database. Follow the steps to create the tables, indexes, foreign keys, and triggers. The steps that follow apply to systems that use DB2 or Microsoft SQL Server databases.

Before you begin

Before you follow the steps, see the links at the end of this topic to manually install the core physical MDM database tables for your database and operating system.

Procedure

1. After you have installed the physical MDM database tables, update the configuration manager table in the schema for the Initiate database with values from the schema for the new InfoSphere MDM installation. For example, if your database is DB2, run *MDM_INSTALL_HOME/database/CoreData/Full/DB2/Standard/ddl/UpdCfgMgrData_temp.sql*. The script updates values in the CONFIGELEMNT table rows with names such as **/IBM/DWLCommonServices/DataBase/OS**, **/IBM/DWLCommonServices/DataBase/type**, and so on.

Note: The UpdCfgMgrData_temp.sql file is not available for DB2 for z/OS. If you are using DB2 with z/OS, run SQL statements as a replacement for the UpdCfgMgrData_temp.sql file. The example that follows indicates an upgrade from version 10.1:

```
UPDATE CONFIGELEMNT SET VALUE_DEFAULT =
'AIX', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/DWLCommonServices/DataBase/OS';
```

```
UPDATE CONFIGELEMNT SET VALUE_DEFAULT =
'DB2', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/DWLCommonServices/DataBase/type';
```

```
UPDATE CONFIGELEMNT SET VALUE_DEFAULT =
'10.1.0.0', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/DWLCommonServices/DataBase/version';
```

```
UPDATE CONFIGELEMNT SET VALUE_DEFAULT =
'EST5EDT', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone';
```

```
UPDATE CONFIGELEMNT SET VALUE_DEFAULT =
'true', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/DWLCommonServices/MultiTimeZoneDeployment/enabled';
```

```
UPDATE CONFIGELEMNT SET VALUE_DEFAULT =
'en', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
```






```
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/DWLCommonServices/NLS/system_Default_Data_Locale';
```

```
UPDATE CONFIGELEMENT SET VALUE_DEFAULT =
'', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/ThirdPartyAdapters/EAS/dsrcCode';
```

```
UPDATE CONFIGELEMENT SET VALUE_DEFAULT =
'', LAST_UPDATE_DT = CURRENT_TIMESTAMP,
LAST_UPDATE_USER='INSTALLATION' WHERE NAME =
'/IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem';
```

2. To validate the changes, view **ELEMENT_ID** to determine whether the placeholder <DB_OS> in **default_value** was replaced with the actual database operating system.
3. Verify the schema with the new version of IBM InfoSphere MDM.

Related concepts:

-  Manual installation of the physical MDM database
-  Manual installation of the physical MDM database on DB2 for Linux or UNIX
-  Manual installation of the physical MDM database on DB2 for z/OS
-  Installing the core database manually on Microsoft SQL Server
-  Manual installation of the physical MDM database on Oracle

Oracle: Adding new InfoSphere MDM database objects to the upgraded Initiate database:

As part of setting up the database, you must add some database objects from your new InfoSphere MDM installation to the existing Initiate database. The steps that follow apply to systems that use Oracle databases.

Procedure

1. Before adding the database objects, create the tablespaces in which the physical tables will be stored:
 - a. Navigate to the *MDM_INSTALL_DIR*\temp\temp_session\database\CoreData\Full\Oracle\Standard\ddl directory.
 - b. Open the create_schema_ora.sql file.
 - c. Follow the instructions in the file and replace the placeholders for tablespace names, database name, and schema with appropriate values. Ensure that the values you provide match the values in the Installation Manager Database Configuration Panel. The tablespace names provided in the Installation Manager are also be available within the *MDM_Variables.properties* file within the *MDM_INSTALL_HOME*\properties directory. Refer to the file to ensure you provide the correct values.
 - d. Log in to the database using the connect statement and run the file using sqlplus SYS/ORACLE AS SYSDBA @create_schema_ora.sql.
 - e. Open the create_schema_ora_cfmgr.sql file in the *MDM_INSTALL_HOME*\temp\temp_session\database\CoreData\Full\Oracle\Standard\ddl directory.
 - f. Follow the instructions in the file and replace the placeholders for tablespace names, database name and schema with appropriate values.
 - g. Log in to the database using the connect statement and execute the file using sqlplus SYS/ORACLE AS SYSDBA @create_schema_ora_cfmgr.sql.

2. Within the MDM 11.3 instance, navigate to the *MDM_INSTALL_HOME*/properties directory.
3. Open the *install_core_db_ora.properties* properties file.
4. Change the values of the properties in the file in order to point to the existing database that you need to upgrade.
5. For the **user.db.password** parameter, change the asterisks to the correct password for the database.
6. For the following properties files, make the same changes to point to the existing database and supply the database password:
 - *install_domain_db_ora.properties*
 - *install_cm_domain_db_oracle.properties*
 - *update_cm_db_oracle.properties*
7. Add the following properties to the *update_cm_db_oracle.properties* configuration file:
 - **db.mtz.enabled=false**
 - **db.mtz.tz=EST5EDT**

The Installation Manager provides sample values for **db.mtz.enabled** and **db.mtz.tz**. The values are available in the *CONFIGELEMNT* table of the database where MDM is installed. Use the following statements to obtain the sample values:






- To obtain the value for **db.mtz.enabled**:

```
SELECT VALUE_DEFAULT FROM <SCHEMA>.CONFIGELEMNT WHERE  
NAME LIKE '/IBM/DWLCommonServices/MultiTimeZoneDeployment/enabled'
```
 - To obtain the value for **db.mtz.tz**:

```
SELECT VALUE_DEFAULT FROM <SCHEMA>.CONFIGELEMNT WHERE  
NAME LIKE '/IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone'
```
8. Within the MDM 11.3 instance, navigate to the *MDM_INSTALL_HOME*/mds/scripts directory.
 9. To create the Core tables and load gold data into the tables, run the following *madconfig* target:
 - For Microsoft Windows: *madconfig install_core_db_ora -propertyfile MDM_INSTALL_HOME/properties/install_core_db_ora.properties*
 - For Linux and UNIX: *./madconfig.sh install_core_db_ora -propertyfile MDM_INSTALL_HOME/properties/install_core_db_ora.properties*
 10. To create the Domain tables and load gold data into the tables, run the following *madconfig* target:
 - For Microsoft Windows: *madconfig install_domain_db_ora -propertyfile MDM_INSTALL_HOME/properties/install_domain_db_ora.properties*
 - For Linux and UNIX: *./madconfig.sh install_domain_db_ora -propertyfile MDM_INSTALL_HOME/properties/install_domain_db_ora.properties*
 11. To create the Configuration Management tables and loads gold data into the tables, run the following *madconfig* target:
 - For Microsoft Windows: *madconfig install_cm_domain_db_oracle -propertyfile MDM_INSTALL_HOME/properties/install_cm_domain_db_oracle.properties*
 - For Linux and UNIX: *./madconfig.sh install_cm_domain_db_oracle -propertyfile MDM_INSTALL_HOME/properties/install_cm_domain_db_oracle.properties*

12. In order to update certain rows in the Configuration Management tables, run the following madconfig target:
 - For Microsoft Windows: `madconfig update_cm_db_oracle -propertyfile MDM_INSTALL_HOME/properties/update_cm_db_oracle.properties`
 - For Linux and UNIX: `./madconfig.sh update_cm_db_oracle -propertyfile MDM_INSTALL_HOME/properties/update_cm_db_oracle.properties`

Related concepts:

-  Manual installation of the physical MDM database
-  Manual installation of the physical MDM database on DB2 for Linux or UNIX
-  Manual installation of the physical MDM database on DB2 for z/OS
-  Installing the core database manually on Microsoft SQL Server
-  Manual installation of the physical MDM database on Oracle

Oracle: Updating the matching type in the configelement table:

As part of the process for updating the matching type, you must obtain the ID of the EBA file. The following steps explain the process as well as the process for updating the information in the configelement table. The steps that follow apply to systems that use Oracle databases.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to the Business Level Application for InfoSphere MDM.
3. Open the composition unit.
4. Open the OSGi Administrative Console.
5. Take note of the bundle name: `com.ibm.mdm.hub.server-ebaID`.
6. To update the matching type in the configelement table:
 - a. Within the MDM 11.3 instance, navigate to the `MDM_INSTALL_HOME/mds/scripts` directory.
 - b. Run the following madconfig target. If you are using probabilistic matching, set the value of `engineType` to PME. If you are using deterministic matching, set the value of `engineType` to DME. For the `database_type`, choose db2, oracle, or mssqlu. (The value mssqlu corresponds to Microsoft SQL Server. The option is available only for Windows installations.) Provide appropriate values in the corresponding place holders.
 - For Microsoft Windows: `madconfig Update_configelement_matching -DebaID=ebaID -DInstallHome=installation_directory -Dmad.db.host=hostname -Dmad.db.port=port -Dmad.db.type=database_type -Dmad.db.dsn=DBName_ebaID -Dmad.db.service=DBName -Duser.db.schema=schema_name -Dmad.db.user=user_name -Dmad.db.password=password -DengineType=matching_engine_type`
 - For Linux and UNIX: `./madconfig.sh Update_configelement_matching -DebaID=ebaID -DInstallHome=installation_directory -Dmad.db.host=hostname -Dmad.db.port=port -Dmad.db.type=database_type -Dmad.db.dsn=DBName_ebaID -Dmad.db.service=DBName -Duser.db.schema=schema_name -Dmad.db.user=user_name -Dmad.db.password=password -DengineType=matching_engine_type`

Updating the installation name in the appsoftware table:

Whether your system uses DB2, SQL Server, or Oracle databases, you must update the name of the installation within the appsoftware table.

About this task

As part of the process for updating the installation name, you must obtain the ID of the EBA file. The following steps explain the process as well as the process for updating the name in the appsoftware table.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to the Business Level Application for InfoSphere MDM.
3. Open the composition unit.
4. Open the OSGi Administrative Console.
5. Take note of the bundle name: `com.ibm.mdm.hub.server-ebaID`.
6. To update the installation name in the appsoftware table:
 - a. Within the MDM 11.3 instance, navigate to the `MDM_INSTALL_HOME/mds/scripts` directory.
 - b. Run the following madconfig target. For the `database_type`, choose `db2`, `oracle`, or `mssql`. (The value `mssql` corresponds to Microsoft SQL Server. The option is available only for Windows installations.) Provide appropriate values in the corresponding place holders. For the `application_version`, use `11.3.0`. If a parameter value contains a space, enclose the value in double-quotation marks.
 - For Microsoft Windows: `madconfig Update_appsoftware_eba -DebaID=ebaID -DInstallHome=installation_directory -Dmad.db.host=hostname -Dmad.db.port=port -Dmad.db.type=database_type -Dmad.db.dsn=DBName_ebaID -Dmad.db.service=DBName -Duser.db.schema=schema_name -Dmad.db.user=user_name -Dmad.db.password=password -DAppVer=application_version`
 - For Linux and UNIX: `./madconfig.sh Update_appsoftware_eba -DebaID=ebaID -DInstallHome=installation_directory -Dmad.db.host=hostname -Dmad.db.port=port -Dmad.db.type=database_type -Dmad.db.dsn=DBName_ebaID -Dmad.db.service=DBName -Duser.db.schema=schema_name -Dmad.db.user=user_name -Dmad.db.password=password -DAppVer=application_version`

Pointing the new MDM installation to the upgraded database

Using the WebSphere Application Server Administration Console, set the `MDM.ear` file to point the upgraded product database.

Procedure

1. In the WebSphere Application Server administration console, navigate to the InfoSphere MDM Server instance.
2. Change the data source properties of the data source names to refer to the upgraded database.
3. For each data source name, change the JAC authentication information to enable access to the upgraded database.
4. Refer to the applicable properties in the upgraded database.

5. Update the **Common and required data source properties** section of each data source name to reflect the properties of the upgraded database.
6. From the Administration Console, test the connection to each data source.
7. Verify that the connection is properly established.


Moving the Initiate user repository

If your installation of Initiate Master Data Service uses the internal LDAP user repository, you can move the content of the repository into a WebSphere Application Server user repository or into an external LDAP repository that you choose.

About this task

Before version 11.0 of the product, Initiate MDS customers maintained their user repository in one of two ways. The default option was to store the users and groups in an internal LDAP repository. As an alternative, the product could be configured to authenticate against a customer's own LDAP repository. The procedures that are listed in the topics that follow apply to the customers who are using the internal LDAP repository. If you want to continue to use an LDAP repository that is external to the product, refer to the WebSphere Application Server documentation to implement your configuration.

Related concepts:

 [madconfig utility usage for virtual MDM implementations](#)

Moving from Initiate internal LDAP to WebSphere for Windows systems:

For Windows systems, a madconfig target can move the content of the LDAP repository within your Initiate installation into a WebSphere Application Server user repository.

About this task

Note: An error requires extra configuration steps for non-Windows systems. If you are running a non-Windows system, follow the link at the end of this topic.

Procedure

1. At a command-line prompt, navigate to the scripts directory of the new IBM InfoSphere MDM installation: *MAD_INSTALL_HOME/scripts*.
2. Run the command: `madconfig.bat migrate_users_and_groups`
3. Complete the prompts that follow with the appropriate information:
 - a. **Enter the instance's LDAP directory** - The location of the Initiate internal LDAP directory.
 - b. **Enter WAS home** - The location of the WebSphere Application Server instance.
 - c. **Enter WAS host** - The host name or IP address of the WebSphere Application Server.
 - d. **Enter WAS port** - The SOAP port address for the WebSphere Application Server.
 - e. **Enter WAS user** - The WebSphere Application Server administrative user name.
 - f. **Enter WAS password** - The WebSphere Application Server administrative user password.

- g. **Enter trust file path and name** - Fully qualified path to the WebSphere Application Server client truststore.
- h. **Enter trust file password** - The password for the WebSphere Application Server client truststore.
- i. **Enter default password for user creation (none to use the username)** - Default password for user creation.

The progress of the user and group migration is output to the console as the migration proceeds.

Related tasks:

“Moving from Initiate internal LDAP to WebSphere for non-Windows systems”

Moving from Initiate internal LDAP to WebSphere for non-Windows systems:

For non-Windows systems, make a small configuration change to enable a madconfig target to move the content of the LDAP repository within your Initiate installation into a WebSphere Application Server user repository.

Procedure

1. Within your Initiate Master Data Service installation, navigate on the command line to the *MAD_ROOTDIR/inst/inst-name/ldap/bin* directory.
2. Open the *_script-util.sh* for editing.
3. Replace *MAD_ROOTDIR=\${MAD_ROOTDIR}* with a literal path, for example, *MAD_ROOTDIR="/opt/IBM/Initiate/Engine10.1"*
4. Save and close the file.
5. Navigate to the scripts directory within your new IBM InfoSphere MDM installation: *MAD_INSTALL_HOME/scripts*.
6. Run the command: *madconfig.sh migrate_users_and_groups*
7. Complete the prompts that follow with the appropriate information:
 - a. **Enter the instance's LDAP directory** - The location of the Initiate internal LDAP directory.
 - b. **Enter WAS home** - The location of the WebSphere Application Server instance.
 - c. **Enter WAS host** - The host name or IP address of the WebSphere Application Server.
 - d. **Enter WAS port** - The SOAP port address for the WebSphere Application Server.
 - e. **Enter WAS user** - The WebSphere Application Server administrative user name.
 - f. **Enter WAS password** - The WebSphere Application Server administrative user password.
 - g. **Enter trust file path and name** - Fully qualified path to the WebSphere Application Server client truststore.
 - h. **Enter trust file password** - The password for the WebSphere Application Server client truststore.
 - i. **Enter default password for user creation (none to use the username)** - Default password for user creation.

The progress of the user and group migration is output to the console as the migration proceeds.

Related tasks:

“Moving from Initiate internal LDAP to WebSphere for Windows systems” on page 34

Moving from Initiate internal LDAP to customer-provided external LDAP:

You can move the content of the LDAP repository within your Initiate installation into an external LDAP repository that you choose.

Procedure

1. At a command-line prompt, go to the bin directory where the LDAP repository is installed for your version of Initiate: *MAD_HOMEDIR/inst/mpinet_instance_name/ldap/bin*
2. Run the applicable command.
 - For Microsoft Windows: `export-ldif.bat --backendID userRoot --ldifFile myldifexportfile`
 - For Linux and UNIX: `export-ldif.sh --backendID userRoot --ldifFile myldifexportfile`

The command creates the .ldif file with the name you specify.

3. Using the utilities included with your LDAP repository, import the .ldif file.

Migrating configuration settings from Initiate

Migrate customized values from key configuration files within Initiate so that they are available within your new InfoSphere MDM installation.

About this task

Several components of IBM InfoSphere MDM Standard Edition use configuration files to control certain behaviors. The installer for the new version of InfoSphere MDM does not automatically migrate any custom configuration values from your Initiate installation to your new installation. The following steps describe migrating customized configuration values for the Entity Manager, Task Manager, and Event Manager. For more information about the default values for the Initiate files, see the link about Master Data Engine variables at the end of this topic.

Procedure

1. Within your installation of Initiate Master Data Service, navigate to the *MAD_HOMEDIR/inst/instanceName/conf* directory.
2. Open the *com.initiate.server.entity.cfg* file.
3. Migrate any custom configuration values to the *com.ibm.mdm.mds.entity.manager.cfg* configuration file within your new InfoSphere MDM installation: *WAS_PROFILE_HOME/conf/com.ibm.mdm.mds.entity.manager.cfg*.

Note: For implementations with multiple instances of the Master Data Engine, ensure that you maintain the unique values for the *queueWrkOwner* variable between the engines in order to prevent collisions with the queue records that another engine is processing.



4. Return to the *MAD_HOMEDIR/inst/instanceName/conf* directory within your Initiate installation.
5. Open the *com.initiate.server.task.cfg* file.
6. Migrate any custom configuration values to the *com.ibm.mdm.mds.task.manager.cfg* configuration file within your InfoSphere MDM installation: *WAS_PROFILE_HOME/conf/com.ibm.mdm.mds.task.manager.cfg*.

Note: The remaining steps apply only if your installation uses event notification.

7. Return to the `MAD_HOMEDIR/inst/instanceName/conf` directory within your Initiate installation.
8. Open the `com.initiate.server.event.cfg` file.
9. Migrate any custom configuration values to the `com.ibm.mdm.mds.event.manager.cfg` configuration file within your InfoSphere MDM installation: `WAS_PROFILE_HOME/conf/com.ibm.mdm.mds.event.manager.cfg`.

Note: For implementations with multiple instances of the Master Data Engine, ensure that you maintain the unique values for the `queueWrkOwner` variable between the engines in order to prevent collisions with the queue records that another engine is processing.

Related concepts:

-  Operational server configuration files
-  Operational server environment variables

Upgrading Initiate clients and other components



You must install the 11.3 clients and set up their connections to the operational server.

About this task

If you use other MDM products, such as Inspector, Web Reports, or Policy Monitoring, install the Version 11.3 editions of each client and connect to the operational server. For more detail about security, see the links at the end of this topic.



Also see the links for details about upgrading IHE and other components and features.

Related concepts:

-  Upgrading to policy monitoring version 11.0
-  Upgrade brokers from an earlier major release version

Related tasks:

“Updating custom code for new security requirements” on page 22

-  Upgrading Virtual MDM projects in the Workbench
-  Updating an existing IHE installation on IBM WebSphere

Related information:

-  Managing users

Upgrading Flexible Search:

To enable Flexible Search with Version 11.3, you must account for a change to how node information is stored.

About this task

With Version 11.3, the format of the Flexible Search node information reflects the underlying WebSphere Application Server architecture.

As of Version 11.3, the node name convention is *cellName-nodeName-serverName*, where the *cellName* and *nodeName* indicate the name of the WebSphere cell and node where the Version 11.3 operational server is hosted. The *serverName* is the name of the server hosting the WebSphere Application Server.

The Version 10.1 node names do not follow the new naming convention. Those node names are stored and referenced in the following virtual MDM tables:

- `mpi_clustercfg`
- `mpi_idxique`

In order to address enable Flexible Search with your Version 11.3 instance, follow the steps to ensure that the tables reflect the new node name value derived from the WebSphere Application Server container configuration.

Procedure

1. In order to stop MEMPUT operations, disconnect all source systems from the previous instance.
2. Poll the `mpi_idxique` table until queue is drained. To determine whether the queue is drained, use the following SQL statement: `select count(*) from mpi_idxique where nodename='version10.1_node_name'` until zero rows are counted. For *version10.1_node_name* use the value of the **`indexSyncSvc.nodeName`** variable in the `hub_instance_path/conf/com.initiate.server.search.cfg` configuration file.

Note: Draining the queue is required in order to eliminate any references to the old node name in the queue. Draining the queue also synchronizes the Flexible Search indexes with the latest update operations for the hub before migration. The previous node name might have been present as a work item (record) in **`mpi_idxique.wrkowner`** field.

3. Stop the previous instance.
4. Update the `mpi_clustercfg` table using following SQL statement: `update mpi_clustercfg set nodename = 'version11.0_node_name' where nodename='version10.1_node_name'`. For *version11.0_node_name* use the new node name convention described previously. For *version10.1_node_name* use the value of the **`indexSyncSvc.nodeName`** variable in the `hub_instance_path/conf/com.initiate.server.search.cfg` configuration file.
5. Copy all index directory contents from the previous instance to the Version 11.3 instance. The index directory location for the previous instance is specified by the **`searchFactory.indexDir`** variable in the `hub_instance_path/conf/com.initiate.server.search.cfg` configuration file. The Version 11.3 index directory location is specified by the **`indexDir`** variable in the `WAS_PROFILE_HOME/conf/com.ibm.mdm.mds.search.index.cfg`.

Finalizing and verifying your upgrade

You finish the upgrade process by finishing some final tasks and verifying that the new system is functional.

Procedure

1. Start the operational server.
 - a. Open IBM WebSphere Application Server Administrative Console.
 - b. Select **Applications > Application Types > Business-level applications**.
 - c. On the Business-level applications page, select your application. A red X in the **Status** field indicates the application is stopped. A green arrow indicates that the application is running.

- d. Click **Start**.
2. Verify that the new operational server starts with no errors in the log.
3. If you have custom applications that access the engine using API calls, install and configure the API, update your code, and recompile the custom applications against the new API. Note that as of InfoSphere MDM version 11.0, the API includes the removal of the direct TCP port. All API calls are biw routed through WebSphere HTTP ports.
4. If you use custom callout handlers, update them to use the OSGi framework, recompile them using the new API, and redeploy them from the IBM InfoSphere MDM Workbench. For instructions, see the Workbench documentation.
5. If you use policy monitoring, migrate its configuration files and update its database and tables. For instructions, see the policy monitoring documentation.
6. If you use one or more of the various messaging brokers, convert their configuration files. For instructions, see the broker documentation.
7. If you use the IHE Accelerator project, upgrade its components. For instructions, see the IHE documentation.
8. Run InfoSphere MDM Inspector to confirm that your data is visible and available.
9. For optional components, run their user interfaces to test their functionality.

Related concepts:

 Diagnostic logging

Upgrading from InfoSphere MDM Server, prior to version 11

A set of steps describes the process of upgrading from InfoSphere MDM Server to InfoSphere MDM.

About this task

This section provides instructions for upgrading InfoSphere MDM Server database and enterprise application.

Note: Before you review any other documentation or upgrade the product, read the release notes and the installation instructions for the IBM InfoSphere Master Data Management product.

Upgrading your MDM environment often requires assistance from members of your own organization as well as IBM Services or an IBM consulting partner. Do not upgrade your implementation without first identifying the team and relying on guidance from that team.

InfoSphere MDM release 11.0 introduced several large-scale architecture changes to the InfoSphere MDM Server product. The updated architecture uses an OSGi framework (Open Services Gateway initiative) in which you can install, start, stop, and update component bundles without rebooting the system. The bundles, which are referred to as component bundle archives (CBAs) create an architecture that is service-oriented, enabling InfoSphere MDM to work within evolving enterprise service-oriented architectures.

By contrast, InfoSphere MDM Server users had to open, alter, and repackage the application EAR file to deploy the data model and any customizations. In the OSGi

framework, users deploy such customizations as independent CBAs that communicate with the application without requiring changes to the application module itself.

The new OSGi architecture also means that the embedded ILOG engine is no longer available. Instead the ILOG engine (now called ODM) can be called remotely. You can determine whether you use ILOG rules with your installation of InfoSphere MDM Server by running a script provided with the new version of InfoSphere MDM. You may want to recreate those rules in ODM. The existing rules cannot be exported.

Note that the management of configuration information changes significantly with the new version of InfoSphere MDM. In previous releases, configuration data was maintained within XML files embedded within the JAR files whose code the configuration settings controlled. The Configuration Manager read from the XML files and wrote the configuration settings into the CONFIGELEMENT table in the database. With the latest InfoSphere MDM release, configuration data is stored exclusively in the CONFIGELEMENT table. The XML files are no longer a part of configuration management.

The latest version also includes security improvements compared to InfoSphere MDM Server. Upgrade scripts included with the product configure the new installation to use the legacy security mode. To take advantage of the new security mode after upgrade, disable the TrustedClientMode setting within the CONFIGELEMENT table.

The following overview diagram conceptually shows the upgrade process for the various components, where the grey boxes show components that are taken offline after the upgrade:

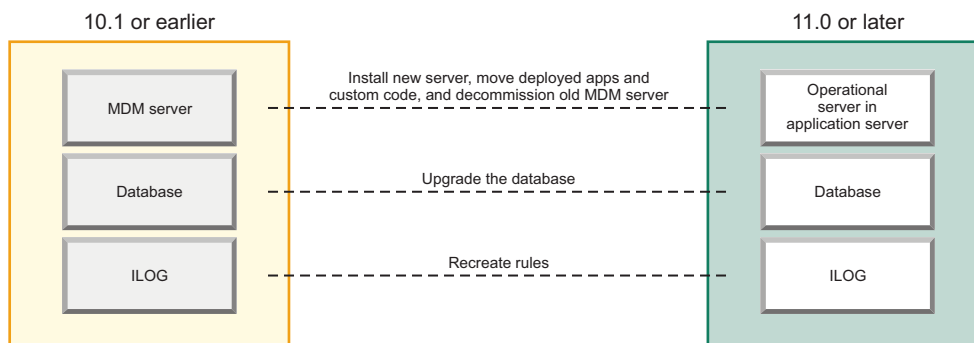


Figure 2. Upgrade overview: InfoSphere MDM Server components in 11.0

Related concepts:



Architecture and concepts

Migrating customizations to work with OSGi

Existing customizations must be migrated to use the OSGi framework.

About this task

If your customizations were created within a version of the InfoSphere MDM Server Workbench prior to 11.0, use the 11.3 version of the Workbench to configure them to work with the new version of InfoSphere MDM.

For details, see the link at the end of this topic.

By contrast, if you created your customizations without using the Workbench, migrate the customizations to OSGi manually.

For details, see the link at the end of this topic.

Related concepts:

 [Migrating an existing InfoSphere MDM development project to the OSGi framework](#)

 [Overview of OSGi migration](#)

Deploying additions and extensions in OSGi

After you have completed the migration of the customizations to work with OSGi, deploy them to the operational server instance.

About this task

For details, see the link at the end of this topic.

Related concepts:

 [Migrating an existing InfoSphere MDM development project to the OSGi framework](#)

Upgrading the existing database for InfoSphere MDM Server

You upgrade the existing database for IBM InfoSphere MDM Server instead of using a new MDM database.

Before you begin

Before you begin the upgrade, ensure that:

- You created a backup copy of your database and any additions or extensions that you created.
- You read the InfoSphere MDM Server ReadMe for the hardware and software requirements and other information.

About this task

The upgrade applies only to the original database structure and data. If you made any customizations or extensions, the changes might be deleted or changed by the upgrade.

Data upgrade scripts: Allowing for late changes and for different languages, an upgrade package might have different script sets, and one or more scripts might not exist. Skip any steps that refer to scripts that are not available. If any of the scripts are empty or not included, they are not applied to the upgrade and can be ignored.

Related tasks:

[“Upgrading an existing DB2 database for UNIX or Linux” on page 4](#)

Creating and updating database objects:

To set up the database for the new version within the previous version of the database, run SQL scripts to create and update tables, indexes, foreign keys, and triggers.

Upgrading an existing DB2 database for UNIX or Linux:

Follow the steps to upgrade to the latest version of for DB2 on UNIX or Linux.

Before you begin

Before you begin, make sure that you read:

- The information on gold data deletes and data upgrade scripts in the topic about upgrading the existing database for IBM InfoSphere MDM Server. See the link at the end of this topic.
- The notes on table spaces in the Manual installation of the physical MDM database topic in the IBM InfoSphere Master Data Management Standard and Advanced Editions Installation Guide

About this task

You have two options for upgrading your existing DB2 database for UNIX or Linux:

- Automatic upgrade - For the automatic upgrade, you run a madconfig target that finds and runs a set of individual database upgrade scripts delivered with the current release. The madconfig target runs the version-specific scripts in sequence until the database is full upgraded to the current version. For more information, see the link at the end of this topic.
- Manual upgrade - For the manual upgrade, you run the individual version-specific scripts manually. For detailed steps, see the link at the end of this topic.

Related concepts:


 Manually installing the madconfig utility

 madconfig utility targets for InfoSphere MDM

Related tasks:

“Upgrading the existing database for InfoSphere MDM Server” on page 41

Related information:

 Manual installation of the physical MDM database

Automatically upgrading the database on DB2 for UNIX or Linux:








You can run madconfig upgrade-mdm-ae-db to perform an automatic upgrade of the MDM content on DB2 for UNIX or Linux.

For IBM InfoSphere MDM Advanced Edition, a madconfig target can perform an automatic database upgrade from any version of InfoSphere MDM Server from 8.5 onward. You must supply the version information for your current installation of InfoSphere MDM. The madconfig utility can then run the database upgrade steps required to increment from that version to the latest version of InfoSphere MDM. The madconfig utility is run from the `MDM_INSTALL_HOME/mds/scripts` directory. When you run madconfig upgrade-mdm-ae-db target, the utility runs upgrade scripts in the `MDM_INSTALL_HOME/database/Upgrade/` directory, which contains the separate directories for each previous version. The folders contain SQL scripts for performing the upgrade for that version. The madconfig target runs the required scripts in sequence for each version.

With some MDM Server installations prior to version 10, the Event Manager data could be stored in a set of tables separate from the MDM content database. If your installation has Event Manager data stored in separate tables, merge the tables into the main schema before proceeding with the upgrade.

Also before proceeding with the upgrade you must manually edit the `upgradepaths.properties` file to provide values that the `madconfig` target requires, including the workspace directory, preferred language, preferred industry, database type, database credentials, and so on. Look for the properties file in the `MDM_INSTALL_HOME/mds/scripts` directory.

Related tasks:

-  Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
-  Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
-  Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
-  Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0
-  `madconfig` utility usage for virtual MDM implementations

Merging Event Manager tables:

If you are upgrading from an installation prior version 10.0, you might have Event Manager data stored in tables that are separate from the master data tables. Before upgrading, merge the tables into the primary database.

Before you begin

The following are the names of the Event Manager tables:

- ADAPTERDEF
- CDEVENTCAT
- PROENTITY
- ENTITYEVENTCAT
- EVENT
- CDEVENTDEFTP
- EVENTDEFEXTRULE
- PROCESSACTION
- PROCESSCONTROL
- CACHINGSERVICE
- ENTITYEVENTCATOPT

Determine whether those tables are stored in a separate database.

About this task

To merge the Event Manager tables run the scripts contained in the `EM_Merge` folder, located in the `/database/Upgrade` directory.

Procedure

1. Navigate to the EM_Merge directory: `MDM_INSTALL_HOME/database/Upgrade/EM_Merge`

Note: The EM_Merge directory also appears in certain subdirectories within the Upgrade directory. Regardless of where it appears in the hierarchy, the contents of the directory are always the same.

2. For the scripts in EM_Merge/DB2/Standard/ddl folder, change the placeholder tags in all of the scripts as follows:
 - a. Replace <SCHEMA> with the schema name or owner with necessary privileges. This must be in uppercase.
 - b. Replace <TABLE_SPACE> with a tablespace name that refers to the tablespace where the data is located.
 - c. Replace <INDEX_SPACE> with a tablespace name that refers to the tablespace where the indexes are located.
 - d. Replace <LONG_SPACE> with a valid tablespace name for storing long type data, either CLOB or XML data.
 - e. Replace <PATH> with the location where the data files will be stored.
3. Connect to your current Event Manager database.
4. Export the current Event Manager data, first defining the <PATH> inside the script, then from the command line run the following script:
`db2 -tvf ExportEMData.sql -l <log_file_name>`
5. Connect to your current MDM database, and set the DMD Schema Name in following scripts by running the command `db2 -tvf <script_name> -l <log_file_name>` for each of the following:
 - a. CreateEM_Tables.sql to create EM tables
 - b. CreateEM_Hist_Tables.sql to create EM history tables
 - c. CreateEM_Ix.sql to create EM Indexes
 - d. CreateEM_Fk.sql to create EM foreign Keys
6. Import the Event Manager data you got it from step 3, using the same <PATH>, then from the command line run the scripts, using the command:
`db2 -tvf ImportEMData.sql -l <log_file_name>`
7. Run the following commands to create Event Manager triggers, and make sure that you choose the same type of trigger with your MDM database:
 - a. Create simple triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_simple.sql -l <log_file_name>`
 - b. Create compound triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_compound.sql -l <log_file_name>`
 - c. **Optional:** Create simple delete triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_delete_simple.sql -l <log_file_name>`
 - d. **Optional:** Create compound delete triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_delete_compound.sql -l <log_file_name>`
8. **Optional:** After you have verified that you have successfully merged the Event Manager database into the database, you can drop the old Event Manager database and schema.

Editing the upgrade properties file:

Manually edit the UpgradePaths.properties properties file to allow the ant script to perform the database upgrade.

About this task

The UpgradePaths.properties properties file enables the madconfig target. It contains user information, connection information, and information about tablespace correspondences.

Procedure

1. Navigate to the *MDM_INSTALL_HOME*/mds/scripts directory.
2. Open the UpgradePaths.properties file in your preferred text editor.
3. See the instructions in the comments within the file itself for more information about each parameter. The specific parameter names are listed here:
 - **mdm.upgrade.basedir** - The directory you specify here will match the directory you navigated to in Step 1.
 - **rdm.upgrade.basedir** - This parameter is commented out by default. Do not uncomment or change the parameter. It is available to enable future functionality.
 - **mdm.ae.locales** - When it was installed, your current installation of InfoSphere MDM was configured to use one or more locales. Specify the same locales here that were specified with the original installation.
 - **mdm.ae.industry** - When it was installed, your current installation of InfoSphere MDM was configured to include one or more industry data sets. The values can be banking, financial, telco, or manufacturing. Use the same value or values that were used with the original installation.
 - **mdm.ae.triggers** - When it was installed, your current installation of InfoSphere MDM was configured to use either simple triggers, compound triggers, or no triggers. Refer to your installation notes for the types of triggers that are installed in MDM and enter the same value for this parameter. The type you specify here must match what is already configured for MDM.
 - **mdm.ae.delete.triggers** - As with **mdm.ae.triggers**, choose compound, simple, or none depending on how your current MDM installation is configured.
 - **db.type** - The current release supports DB2.
 - **db.subtype** - The current release supports Standard.
 - **db.driver.class**
 - **db.driver.path**
 - **db.jdbc.url**
 - **db.uid** - The user ID that is used to connect to the database and to run the upgrade scripts. Ensure that it has sufficient privileges to alter schemas and perform other necessary operations.
 - **db.pwd** - The password that corresponds to the user ID you provided.
 - **db.name** - The name of the database you to which the scripts will connect, for example, MDMDDB.
 - **starting.version.number** - Enter the version of MDM that you will be upgrading. InfoSphere MDM Version 11.3 includes the scripts that are necessary to upgrade from Version 8.5.0. The correct version numbers to use for the **starting.version.number** parameter are 8.5.0, 9.0.1, 9.0.2, 10.0.0, 10.0.1, and 11.0.0.
 - **upgrade.path.8.5.0** - Do not edit this value.
 - **upgrade.path.9.0.1** - Do not edit this value.
 - **upgrade.path.9.0.2** - Do not edit this value.

- **upgrade.path.10.0.0** - Do not edit this value.
- **upgrade.path.10.1.0** - Do not edit this value.
- **upgrade.path.11.0.0** - Do not edit this value.
- **SCHEMA** - Some table names in the upgrade scripts are qualified with the **SCHEMA** name or the **DBUSER** name. The two terms mean the same thing. Some older scripts used the term **DBUSER** while others used the term **SCHEMA**.
- **DBUSER** - See previous note.
- **TABLE_SPACE** - Enter name of the User Data table space name. Provide the value that was used when the MDM database was first created. Refer to the database. The default pool size is 8K.
- **INDEX_SPACE** - Enter the name of Index Table space. Provide the value that was used when the MDM database was first created. Refer to the database.
- **LONG_SPACE** - Enter the name of the log data table space. Provide the value that was used when the MDM database was first created. Refer to the database.
- **TABLE_SPPMD** - Enter the name of the tablespace to contain the new Probabilistic Matching Engine tables. The upgrade process will create the tablespace for you.
- **TABLE_SPPMI** - Enter the name of the tablespace to contain the new Probabilistic Matching Engine indices. The upgrade process will create the tablespace for you.
- **DB_OS** - Enter the desired default value for /IBM/DWLCommonServices/ DataBase/OS
- **DB_TYPE** - The current release supports DB2.

Note: The **DB_TYPE** element is part of the configuration of MDM at runtime. The **db.type** parameter mentioned previously identifies the folder in the upgrade scripts for the database you want to upgrade. The two parameters should always refer to the same type of database, for example "Oracle" for Oracle, and "DB2" for DB2, but the specific syntax of each parameter might differ.

- **DB_VERSION** - Enter the desired default value for /IBM/DWLCommonServices/ DataBase/version. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **DEFAULT_TIMEZONE** - Enter the desired default value for /IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **TIMEZONE_ENABLED** - The current release supports a value of false.
- **NLS_SYS_LANG** - Enter the desired default value for /IBM/DWLCommonServices/NLS/system_Default_Data_Locale. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **DSRC_CODE** - Enter desired default value for /IBM/ThirdPartyAdapters/EAS/dsrcCode. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **EXCLUSIVE_CIENT_SYSTEMS** - Enter desired default value for /IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem. See the link at the end of this topic for more information about the elements in the Configuration and Management component.

4. Save and close the file.

Related tasks:

 Elements in the Configuration and Management component

Running the madconfig target for database upgrade:

After you have completed any prerequisite steps, run the madconfig upgrade-mdm-ae-db target to perform the automatic database upgrade.

About this task

For IBM InfoSphere MDM Advanced Edition, a madconfig target can perform an automatic database upgrade from any version of InfoSphere MDM Server from 8.5 onward. The madconfig target detects your current version of InfoSphere MDM and runs the database upgrade steps required to increment from that version to the latest version of InfoSphere MDM. Although you run the madconfig target from the `MDM_INSTALL_HOME/mds/scripts` directory, the database upgrade scripts that it runs are in the `MDM_INSTALL_HOME/database/Upgrade/` directory.


Procedure

1. At a command-prompt, navigate to the `MDM_INSTALL_HOME/mds/scripts` directory.
2. Run the madconfig target:
 - For Microsoft Windows: `madconfig upgrade-mdm-ae-db -l logfilename`
 - For Linux and UNIX: `madconfig.sh upgrade-mdm-ae-db -l logfilename`

Note: You can use the *logfilename* argument to specify a location and name for the log file. If you do not specify a *logfilename*, the log messages are written out to the console.

3. To review the process, see the log file created by the madconfig target.

Related concepts:

 Manually installing the madconfig utility

 madconfig utility targets for InfoSphere MDM

Viewing a log of the upgrade process:

As the ant script runs, it directs all log information into a single file.

About this task

If you specified a log file name and location when you ran madconfig upgrade-mdm-ae-db, look for the log file in the directory you indicated.

Procedure

1. Navigate to the location you specified for the *logfilename*.
2. Open the file.

Enabling non-case-sensitive search capability:

The automatic upgrade process does not include an option to enable non-case-sensitive search for DB2. If you want to have the ability to do non-case-sensitive search, you can manually enable the capability after you have finished the automatic upgrade.

Procedure

1. Navigate to the `MDM_INSTALL_HOME/database/Full/DB2/Standard/ddl` directory.
2. Run the following script: `db2 -tvf Insensitive_search_enabled.sql -l <log_file_name>`

Manually upgrading the database on DB2:

You can use a set of scripts to do a manual upgrade of the MDM database on DB2.

About this task

The manual upgrade process involves running a series of version-specific scripts for upgrading the MDM database. For example, if you are currently running InfoSphere MDM Server version 9.0.1, you must run each of the following scripts in sequence:

1. A script to upgrade your database from version 9.0.1 to version 9.0.2.
2. A script to upgrade your database from version 9.0.2 to version 10.1.
3. A script to upgrade your database from version 10.1 to version 11.0.
4. A script to upgrade your database from version 11.0 to version 11.3.

Each script occupies a folder within a version-specific directory within `MDM_INSTALL_HOME/database/Upgrade/`. For example, the folder `MDM_INSTALL_HOME/database/Upgrade/9.0.2/Level-I` contains scripts that allow you to upgrade from version 9.0.1 to version 9.0.2. Note that the directories containing the scripts are named for the *target* version.







Certain directories contain both a Level-I directory and a Level-II directory. The scripts within Level-II directories can allow you to skip certain intermediate versions as you upgrade the database.

Be sure to read the relevant documentation for each release before you run the corresponding upgrade script. See the list of links at the end of this topic.

Use the following list to determine which scripts to run and the order to run them:

- If you are currently running version 8.5.0, run the scripts within the following directories, in sequence: `9.0.1/Level-II/DB2/Standard/ddl` then `9.0.2/Level-I/DB2/Standard/ddl` then `10.1.0/Level-II/DB2/Standard/ddl` then `11.0.0/Level-I/DB2/Standard/ddl` then `11.3.0/Level-I/DB2/Standard/ddl`.
- If you are currently running version 9.0.1, run the scripts within the following directories, in sequence: `9.0.2/Level-I/DB2/Standard/ddl` then `10.1.0/Level-II/DB2/Standard/ddl` then `11.0.0/Level-I/DB2/Standard/ddl` then `11.3.0/Level-I/DB2/Standard/ddl`.
- If you are currently running version 9.0.2, run the scripts within the following directories, in sequence: `10.1.0/Level-II/DB2/Standard/ddl` then `11.0.0/Level-I/DB2/Standard/ddl` then `11.3.0/Level-I/DB2/Standard/ddl`.
- If you are currently running version 10.0.0, run the scripts within the following directories, in sequence: `10.1.0/Level-I/DB2/Standard/ddl` then `11.0.0/Level-I/DB2/Standard/ddl` then `11.3.0/Level-I/DB2/Standard/ddl`.
- If you are currently running version 10.1.0, run the scripts within the following directories, in sequence: `11.0.0/Level-I/DB2/Standard/ddl` then `11.3.0/Level-I/DB2/Standard/ddl`.
- If you are currently running version 11.0.0, run the script within the following directory: `11.3.0/Level-I/DB2/`.

Related tasks:

-  Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
 -  Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
 -  Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
 -  Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
 -  Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
 -  Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0
- "Manually upgrading the database on DB2 from 11.0 to 11.3" on page 9

Manually upgrading the database on DB2 from 11.0 to 11.3:

You can use a set of scripts to do a manual upgrade of the MDM database on DB2 from 11.0 to 11.3.

Procedure

1. Navigate to the `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl` directory.
2. In the `ImpReqDataCfgMgr.sql` and `QryUpdCfgMgrData.sql` scripts, replace the `<SCHEMA>` placeholder with a schema name or owner with necessary privileges. This value must be in uppercase.
3. In the `QryUpdCfgMgrData.sql` script, replace the following tokens with values that are appropriate to your own installation:
 - **DB_OS** - Enter the desired default value for `/IBM/DWLCommonServices/Database/OS`.
 - **DB_TYPE** - Use the value `DB2`.
 - **DB_VERSION** - Enter the desired default value for `/IBM/DWLCommonServices/Database/version`
 - **DEFAULT_TIMEZONE** - Enter the desired default value for `/IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone`
 - **TIMEZONE_ENABLED** - Use the value `false`.
 - **NLS_SYS_LANG** - Enter the desired default value for `/IBM/DWLCommonServices/NLS/system_Default_Data_Locale`
 - **DSRC_CODE** - Enter desired default value for `/IBM/ThirdPartyAdapters/EAS/dsrcCode`
 - **EXCLUSIVE_CIENT_SYSTEMS** - Enter desired default value for `/IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem`
4. Connect to the database that you are upgrading.
5. Replace the `<SCHEMA>` tag in the scripts in the following folders with a schema name or owner, in uppercase. `CODE_LANG` in the folder name is the language that is currently installed. If you are loading English data only, set `CODE_LANG` to `en`. Otherwise, set `CODE_LANG` to the language code of the language you are loading.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_req_CODE_LANG/data`
- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_CODE_LANG/data-industry`

- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_opt_CODE_LANG/data*
6. From the command line, to load the common data, English code table data, and configure the table in the language that you selected, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_req_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The *insert.sql* script inserts new data
 - b. The *update.sql* script updates existing table data.
 - c. The *delete.sql* deletes obsolete table data.
 7. From the command line, to load the English industry data, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_en/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The *insert.sql* script inserts new data
 - b. The *update.sql* script updates existing table data.
 - c. The *delete.sql* deletes obsolete table data.
 8. If you are loading industry data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_CODE_LANG/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The *insert.sql* script inserts new data
 - b. The *update.sql* script updates existing table data.
 - c. The *delete.sql* deletes obsolete table data.
 9. If you are loading code table data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_opt_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The *insert.sql* script inserts new data
 - b. The *update.sql* script updates existing table data.
 - c. The *delete.sql* deletes obsolete table data.
 10. Repeat the previous two steps for each non-English language that is installed.

Note: If you have multiple deployments, before you run the following script, change the SQL clause `DEPLOYMENT_ID <> 1000` to `DEPLOYMENT_ID = DEPLOYMENT_ID` where the *DEPLOYMENT_ID* is the ID of the instance you want to upgrade.

11. From the command line, run the command: `db2 -tvf ImpReqDataCfgMgr.sql -l log_file_name`.
12. From the command line, run the command: `db2 -tvf QryUpdCfgMgrData.sql -l log_file_name`.

Related tasks:

“Manually upgrading the database on DB2” on page 48

Upgrading an existing DB2 database for z/OS:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for DB2 on z/OS. For information about upgrading the database for previous releases, see the links at the end of this topic.

Procedure

1. To connect to DB2 for OS/390 and z/OS server from a DB2 Linux, UNIX, or Windows server, DB2 Connect must be installed on the server you are connecting from. Enterprise editions contain DB2 Connect. To connect to the database, enter the following information:
 - Host name or IP address
 - DB2 port (default = 446)
 - Database name
 - Subsystem location name
 - User ID and password
2. On the client server, catalog the database by running the following commands in this order:
 - a. `db2 catalog tcpip node YOUR_NODE_NAME remote Z/OS_HOSTNAME/IP_ADDRESS server Z/OS_PORT`
 - b. `db2 catalog db YOUR_DB_NAME at node YOUR_NODE_NAME authentication dcs`
 - c. `db2 catalog dcs db YOUR_DB_NAME as Z/OS_LOCATION_NAME`
3. If you are using this user ID to connect to the DB2 subsystem for the first time, rebind the packages for the database by following either of these steps:
 - a. Rebind all the packages by running the command: `db2rbind YOUR_DB_NAME all /u USER_NAME /p PASSWORD /r any`
 - b. Rebind individual packages by running the command: `db2 bind package_name`
4. Navigate to the `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1` directory.
5. Change the placeholder tags in the all of the scripts:
 - Set <SCHEMA> to a schema name or owner in uppercase.
 - Set <USER_ACCOUNT> to an account with user authority.
6. In the folder name, `CODE_LANG` is the language that is currently installed:







Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_req_CODE_LANG/data`
- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_CODE_LANG/data-industry`
- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_opt_CODE_LANG/data`


If you are loading English data only, set `CODE_LANG` to `en`. Otherwise, set `CODE_LANG` to the language code of the language you are loading.

7. To load the common data from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_req_en/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
8. To load the English industry data from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_en/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
9. If you are loading industry data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_CODE_LANG/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
10. If you are loading code table data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_opt_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
11. Repeat the previous step for each non-English language that is installed.
12. From the command line, run the command: `db2 -tvf ImpReqDataCfgMgr_zos.sql -l log_file_name`.
13. From the command line, run the command: `db2 -tvf QryUpdCfgMgrData_zos.sql -l log_file_name`.

Related tasks:

-  Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
-  Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
-  Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
-  Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0

Related information:

-  Manual installation of the physical MDM database

Upgrading an existing Oracle database:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for Oracle. For information about upgrading the database for previous releases, see the links at the end of this topic.

Before you begin

Read the content for the link at the end of this topic about manual installation of the database.

For Oracle, the use of table spaces for data, indexes and LOBs, such as CLOBs or XML, to improve database performance is supported. Look for the following placeholders in the scripts: <TABLE_SPACE>, <INDEX_SPACE>, and <LONG_SPACE>.

Procedure

1. Replace the placeholder <SCHEMA> in the scripts in the following folders with a schema name or owner. Ensure that the name or owner is in uppercase. *CODE_LANG* in the folder name is the language that is currently installed.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_req_*CODE_LANG*/data
 - *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_*CODE_LANG*/data-*industry*
 - *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_opt_*CODE_LANG*/data
2. To load the common data, to load the English code table data, and to configure the table in the language that you selected, run the following script in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_req_*CODE_LANG*/data directory from the command line. Use the command:
sqlplus <SCHEMA> /<SCHEMAPASSWORD>@<DBNAME> @<script_name> >> <log_file_name>
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.

If you are loading English data only, set *CODE_LANG* to en. Otherwise, set *CODE_LANG* to the language code of the language you are loading.

3. To load the English industry data, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_en/data-*industry* directory from the command line. Use the command: sqlplus <SCHEMA> /<SCHEMAPASSWORD>@<DBNAME> @<scriptname> >> <log_file_name>
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
4. If you are loading data for a language other than English, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_<*CODE_LANG*>/data-*industry* directory from the command line:
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.

- c. The delete.sql script deletes obsolete table data.
5. If you are loading code table data for a language other than English, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_opt_CODE_LANG/data* directory from the command line:

```
sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @<scriptname> >>
<log_file_name>
```







 - The insert.sql script inserts new data.
 - The update.sql script updates existing table data.
 - The delete.sql script deletes obsolete table data.
6. Repeat the previous two steps for each non-English language that is installed.
7. Navigate to the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/* directory.
8. Run the following command:

```
sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME>
@ImpReqDataCfgMgr_ora.script >> <log_file_name>
```
9. Run the following command:


```
sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME>
@QryUpdCfgMgrData_ora.sql >> <log_file_name>
```

 The
QryUpdCfgMgrData_ora.sql script populates the Configuration Manager values based on previous InfoSphere MDM version 11 deployment values.

Related tasks:

-  Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
-  Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
-  Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
-  Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
-  Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0

Related information:

-  Manual installation of the physical MDM database

Creating new tablespaces:

After MDM has been upgraded, but before creating the Initiate database tables, you might need to run a script to create tablespaces.

About this task

The current release of IBM InfoSphere MDM supports upgrading from releases prior to 11.0. Because of changes to the MDM database, if you are upgrading from a release prior to 11.0, you must run a SQL script that creates new tablespaces. If you are upgrading from 11.0, you can skip the following steps.

Procedure

1. Navigate to *MDM_INSTALL_HOME/database/Upgrade/11.0.0/Level-I/DB2/Standard/ddl/*
2. Open the CreateTS.sql script for editing.
3. Update the following values with the values of the actual tablespaces you intend to use for MDM:
 - <TBS4K>
 - <TBS8K>

- <TBS16K>
 - <LONG_SPACE>
4. Run CreateTS.sql.

Creating the Initiate database tables:

As part of upgrading from IBM InfoSphere MDM Server to IBM InfoSphere Master Data Management Advanced Edition, create the Initiate database tables.

About this task

Important: You can skip the steps in this topic if you are upgrading to version 11.3 from version 11.0. In that case, the Initiate tables have already been installed as part of your 11.0 installation.

The latest version of InfoSphere MDM combines the IBM InfoSphere MDM Server with the Initiate Master Data Service product. The combined product relies on the database tables of both component products. Therefore, you must create the Initiate database tables as part of your upgrade process.

Procedure

1. At a command-line prompt, go to the scripts directory of the installation:
MDM_INSTALL_HOME/mds/scripts.
2. Run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig.bat update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password
- g. Database schema - You must supply a schema name if your database is DB2.

Note: In Windows, System DSN will show all the datasource names created. In Linux environments, the *MDM_INSTALL_HOME*/mds/odbc.ini file will show the datasource names.

3. When the script asks whether you want to re-create the existing data source, enter y.
4. The script prompts for the database schema. For IBM DB2 for z/OS, you can submit an alternative schema or enter nothing to use the default schema. For other database types, enter nothing.
5. Take a moment to verify that the settings you entered have been applied:
 - a. Navigate to *MDM_INSTALL_HOME*/mds/conf
 - b. Open the com.ibm.mdm.mds.jni.cfg configuration file and verify the settings.
 - c. Save and close the file.
 - d. Open the com.ibm.mdm.mds.jdbc.cfg configuration file and verify the settings.

- e. Save and close the file.
 6. Set the library path to point to *MDM_INSTALL_HOME*/mds/lib. An incorrect setting can result in a "madhub create" error when you run the bootstrap process. Use the library path variable that is appropriate for your operating system. To see the current setting, use the following commands:
 - Linux or Solaris - export LD_LIBRARY_PATH
 - AIX - export \$libpath
 - Microsoft Windows - set PATH
 7. Confirm that tablespaces are created. The process pre-pends the last three characters of your schema to "ZMDS". For example, if your schema is M5Z10M65, the database identifier will be M65ZMDS.
 8. In the *MDM_INSTALL_HOME*/mds/sql directory, edit the mpihub.sto file so that the database identifier is correct for each tablespace.
 9. From the *MDM_INSTALL_HOME*/mds/scripts directory, run the madconfig bootstrap_datasource command:
 - For Microsoft Windows: madconfig.bat bootstrap_datasource
 - For Linux and UNIX: madconfig.sh bootstrap_datasource
 10. To make sure that everything stays in sync, run the madconfig update_instance_datasource command again, but this time from the native.war directory within the WebSphere Application Server. First, navigate to *WAS_PROFILE_HOME*\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\scripts directory. Then run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource
- The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:
- a. Database type
 - b. Database host
 - c. Database port
 - d. Database name
 - e. Database user
 - f. Database password

Results

The first script updates the following configuration files within the *MDM_INSTALL_HOME*/mds/conf directory:

- com.ibm.mdm.mds.jni.cfg
- com.ibm.mdm.mds.jdbc.cfg

The second script creates the Initiate tables.

Updating the data sources:

As part of the upgrade process, edit the user, password, and other database information for the data sources.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to **Resources > JDBC > Data sources**.
3. Select the data source **DWLConfig**.
4. Edit the database information under **Common and required data source properties** to point to the upgraded database schema.
5. Click on **Custom properties** and edit the user and password of the upgraded database schema. If your DB user name and schema name are different complete these additional steps:
 - a. Within **Custom properties**, click **New**.
 - b. For the **Name** field, enter `currentSchema`.
 - c. For the **Value** field, enter the name of your schema.
6. Also click on **JAAS - J2C authentication data** and edit all of the aliases in the list. (If you did not deploy using MQ, you do not need to edit the alias for `Node01/application-name-MQ-User`.) For each alias, modify the user name and password.
7. Repeat steps 4, 5, and 6 for the data source **DWLCustomer**.
8. Select the data source **MDM**.
9. Click on **Customer properties** to edit the database information of the upgraded database.

Updating the messaging engine:

If your Version 10.1 installation is configured to use WebSphere Embedded Messaging, update the messaging engine as part of the upgrade process.

Procedure

1. Log on to the WebSphere Application Server Administration Console.
2. Navigate to **Service integration > Buses > MDM.SIB.application-name > Messaging Engine > node-name.application-name-MDM.SIB.application-name > Message Store**.
3. Edit the **Schema name** field.
4. Stop the application server.
5. Drop all of the messaging engine tables. The messaging engine table names all begin with **SIB**:
 - SIB000
 - SIB001
 - SIB002
 - SIBCLASSMAP
 - SIBKEYS
 - SIBLISTING
 - SIBOWNER
 - SIBOWNER0
 - SIBXACTS
6. Restart the application server.

Updating the install name in the appsoftware table:

Update the name of the installation within the appsoftware table.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to the Business Level Application for InfoSphere MDM: **Application > Application Types > Business-level applications**. Then look for the application with the description **MDMBLA**.
3. Open the composition unit.
4. Open the OSGi Administrative Console.
5. Note the bundle name: `com.ibm.mdm.hub.server-XXXXXXX`.
6. Log into the MDM database.
7. Select `* from <SCHEMA>.APPSOFTWARE WHERE VERSION = '11.3.0'`. After you run the SQL statement, note the name of the application in the NAME field. It will say **com.ibm.mdm.hub.server**.
8. Update the appsoftware table to add `-[name]` to the `com.ibm.mdm.hub.server` record. For example, if `XXXXXXX` is `E001`, then to update to InfoSphere MDM Version 11.3:
 - DB2: `update appsoftware set NAME='com.ibm.mdm.hub.server-E001', last_update_dt = current timestamp where APPLICATION_ID = 1001;`
 - Oracle: `update appsoftware set NAME='com.ibm.mdm.hub.server-E001', last_update_dt = sysdate where APPLICATION_ID = 1001;`
 - Microsoft SQL Server: `insert into APPSOFTWARE values (10000, 'com.ibm.mdm.hub.server-E001', '11.3.0', CURRENT_TIMESTAMP, 'unknown');`

Recreating ILOG rules

The new OSGi architecture means that the embedded ILOG engine is no longer available. Instead the ILOG engine (now called ODM) can be called remotely. Recreate existing ILOG rules as ODM rules.

About this task

For information about creating ODM rules, follow the link at the end of this topic.

Procedure

To determine whether you use ILOG rules with your previous version of InfoSphere MDM Server, run the following scripts.

1. For any ILOG-based Behavioral Extensions rules:

```
SELECT * FROM EXTENSIONSET
WHERE (JAVA_CLASS_NAME IS NULL OR JAVA_CLASS_NAME = '')
AND (RULE_SET_NAME IS NOT NULL OR RULE_SET_NAME != '')
```
2. For any ILOG-based External Rules:

```
SELECT * FROM RULEENGINEIMPL
WHERE EXT_RULE_IMPL_ID
IN (SELECT EXT_RULE_IMPL_ID FROM EXTRULEIMPLEM
WHERE EXT_RULE_TP_CODE='R')
```

Related information:

 Exporting business rules to IBM Operational Decision Manager

Verifying your upgrade

You finish the upgrade process by verifying that the new system is functional.

Procedure

1. Verify that the new InfoSphere MDM operational server can access the upgraded database.
2. Ensure that your customizations from previous releases are available in the new environment.

Upgrading from InfoSphere MDM Server, prior to version 11

A set of steps describes the process of upgrading from InfoSphere MDM Server to InfoSphere MDM.

About this task

This section provides instructions for upgrading InfoSphere MDM Server database and enterprise application.

Note: Before you review any other documentation or upgrade the product, read the release notes and the installation instructions for the IBM InfoSphere Master Data Management product.

Upgrading your MDM environment often requires assistance from members of your own organization as well as IBM Services or an IBM consulting partner. Do not upgrade your implementation without first identifying the team and relying on guidance from that team.

InfoSphere MDM release 11.0 introduced several large-scale architecture changes to the InfoSphere MDM Server product. The updated architecture uses an OSGi framework (Open Services Gateway initiative) in which you can install, start, stop, and update component bundles without rebooting the system. The bundles, which are referred to as component bundle archives (CBAs) create an architecture that is service-oriented, enabling InfoSphere MDM to work within evolving enterprise service-oriented architectures.

By contrast, InfoSphere MDM Server users had to open, alter, and repackage the application EAR file to deploy the data model and any customizations. In the OSGi framework, users deploy such customizations as independent CBAs that communicate with the application without requiring changes to the application module itself.

The new OSGi architecture also means that the embedded ILOG engine is no longer available. Instead the ILOG engine (now called ODM) can be called remotely. You can determine whether you use ILOG rules with your installation of InfoSphere MDM Server by running a script provided with the new version of InfoSphere MDM. You may want to recreate those rules in ODM. The existing rules cannot be exported.

Note that the management of configuration information changes significantly with the new version of InfoSphere MDM. In previous releases, configuration data was maintained within XML files embedded within the JAR files whose code the configuration settings controlled. The Configuration Manager read from the XML files and wrote the configuration settings into the CONFIGELEMENT table in the database. With the latest InfoSphere MDM release, configuration data is stored exclusively in the CONFIGELEMENT table. The XML files are no longer a part of configuration management.

The latest version also includes security improvements compared to InfoSphere MDM Server. Upgrade scripts included with the product configure the new installation to use the legacy security mode. To take advantage of the new security mode after upgrade, disable the `TrustedClientMode` setting within the `CONFIGELEMENT` table.

The following overview diagram conceptually shows the upgrade process for the various components, where the grey boxes show components that are taken offline after the upgrade:

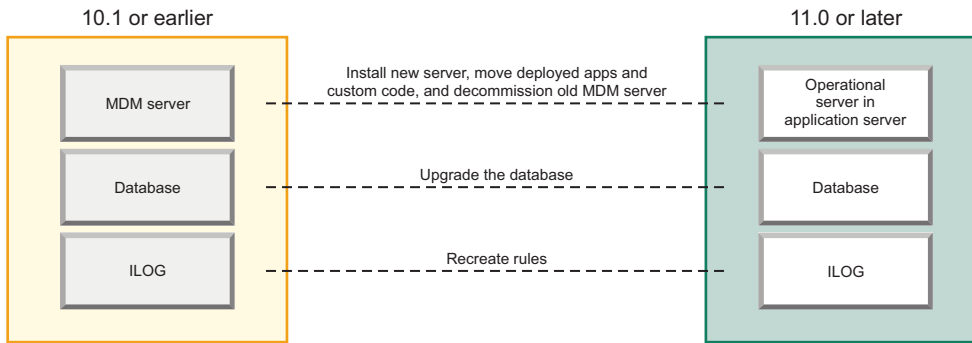


Figure 3. Upgrade overview: InfoSphere MDM Server components in 11.0

Related concepts:

[Architecture and concepts](#)

Migrating customizations to work with OSGi

Existing customizations must be migrated to use the OSGi framework.

About this task

If your customizations were created within a version of the InfoSphere MDM Server Workbench prior to 11.0, use the 11.3 version of the Workbench to configure them to work with the new version of InfoSphere MDM.

For details, see the link at the end of this topic.

By contrast, if you created your customizations without using the Workbench, migrate the customizations to OSGi manually.

For details, see the link at the end of this topic.

Related concepts:

[Migrating an existing InfoSphere MDM development project to the OSGi framework](#)

[Overview of OSGi migration](#)

Deploying additions and extensions in OSGi

After you have completed the migration of the customizations to work with OSGi, deploy them to the operational server instance.

About this task

For details, see the link at the end of this topic.

Related concepts:

 [Migrating an existing InfoSphere MDM development project to the OSGi framework](#)

Upgrading the existing database for InfoSphere MDM Server

You upgrade the existing database for IBM InfoSphere MDM Server instead of using a new MDM database.

Before you begin

Before you begin the upgrade, ensure that:

- You created a backup copy of your database and any additions or extensions that you created.
- You read the InfoSphere MDM Server ReadMe for the hardware and software requirements and other information.

About this task

The upgrade applies only to the original database structure and data. If you made any customizations or extensions, the changes might be deleted or changed by the upgrade.

Data upgrade scripts: Allowing for late changes and for different languages, an upgrade package might have different script sets, and one or more scripts might not exist. Skip any steps that refer to scripts that are not available. If any of the scripts are empty or not included, they are not applied to the upgrade and can be ignored.

Related tasks:

[“Upgrading an existing DB2 database for UNIX or Linux” on page 4](#)

Creating and updating database objects

To set up the database for the new version within the previous version of the database, run SQL scripts to create and update tables, indexes, foreign keys, and triggers.

Upgrading an existing DB2 database for UNIX or Linux:

Follow the steps to upgrade to the latest version of for DB2 on UNIX or Linux.

Before you begin

Before you begin, make sure that you read:



- The information on gold data deletes and data upgrade scripts in the topic about upgrading the existing database for IBM InfoSphere MDM Server. See the link at the end of this topic.
- The notes on table spaces in the Manual installation of the physical MDM database topic in the IBM InfoSphere Master Data Management Standard and Advanced Editions Installation Guide

About this task

You have two options for upgrading your existing DB2 database for UNIX or Linux:

- Automatic upgrade - For the automatic upgrade, you run a madconfig target that finds and runs a set of individual database upgrade scripts delivered with the current release. The madconfig target runs the version-specific scripts in sequence until the database is full upgraded to the current version. For more information, see the link at the end of this topic.
- Manual upgrade - For the manual upgrade, you run the individual version-specific scripts manually. For detailed steps, see the link at the end of this topic.


Related concepts:

-  [Manually installing the madconfig utility](#)
-  [madconfig utility targets for InfoSphere MDM](#)

Related tasks:

[“Upgrading the existing database for InfoSphere MDM Server” on page 41](#)

Related information:

-  [Manual installation of the physical MDM database](#)

Automatically upgrading the database on DB2 for UNIX or Linux:




You can run madconfig upgrade-mdm-ae-db to perform an automatic upgrade of the MDM content on DB2 for UNIX or Linux.

For IBM InfoSphere MDM Advanced Edition, a madconfig target can perform an automatic database upgrade from any version of InfoSphere MDM Server from 8.5 onward. You must supply the version information for your current installation of InfoSphere MDM. The madconfig utility can then run the database upgrade steps required to increment from that version to the latest version of InfoSphere MDM. The madconfig utility is run from the `MDM_INSTALL_HOME/mds/scripts` directory. When you run madconfig upgrade-mdm-ae-db target, the utility runs upgrade scripts in the `MDM_INSTALL_HOME/database/Upgrade/` directory, which contains the separate directories for each previous version. The folders contain SQL scripts for performing the upgrade for that version. The madconfig target runs the required scripts in sequence for each version.

With some MDM Server installations prior to version 10, the Event Manager data could be stored in a set of tables separate from the MDM content database. If your installation has Event Manager data stored in separate tables, merge the tables into the main schema before proceeding with the upgrade.

Also before proceeding with the upgrade you must manually edit the `upgradepaths.properties` file to provide values that the madconfig target requires, including the workspace directory, preferred language, preferred industry, database type, database credentials, and so on. Look for the properties file in the `MDM_INSTALL_HOME/mds/scripts` directory.

Related tasks:

-  [Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0](#)

- ➡ Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0
- ➡ Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0
- ➡ Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0
- ➡ madconfig utility usage for virtual MDM implementations

Merging Event Manager tables:

If you are upgrading from an installation prior version 10.0, you might have Event Manager data stored in tables that are separate from the master data tables. Before upgrading, merge the tables into the primary database.

Before you begin

The following are the names of the Event Manager tables:

- ADAPTERDEF
- CDEVENTCAT
- PROENTITY
- ENTITYEVENTCAT
- EVENT
- CDEVENTDEFTP
- EVENTDEFEXTRULE
- PROCESSACTION
- PROCESSCONTROL
- CACHINGSERVICE
- ENTITYEVENTCATOPT

Determine whether those tables are stored in a separate database.

About this task

To merge the Event Manager tables run the scripts contained in the EM_Merge folder, located in the /database/Upgrade directory.

Procedure

1. Navigate to the EM_Merge directory: *MDM_INSTALL_HOME*/database/Upgrade/EM_Merge

Note: The EM_Merge directory also appears in certain subdirectories within the Upgrade directory. Regardless of where it appears in the hierarchy, the contents of the directory are always the same.

2. For the scripts in EM_Merge/DB2/Standard/ddl folder, change the placeholder tags in all of the scripts as follows:
 - a. Replace <SCHEMA> with the schema name or owner with necessary privileges. This must be in uppercase.
 - b. Replace <TABLE_SPACE> with a tablespace name that refers to the tablespace where the data is located.
 - c. Replace <INDEX_SPACE> with a tablespace name that refers to the tablespace where the indexes are located.
 - d. Replace <LONG_SPACE> with a valid tablespace name for storing long type data, either CLOB or XML data.

- e. Replace <PATH> with the location where the data files will be stored.
3. Connect to your current Event Manager database.
4. Export the current Event Manager data, first defining the <PATH> inside the script, then from the command line run the following script:
`db2 -tvf ExportEMData.sql -l <log_file_name>`
5. Connect to your current MDM database, and set the DMD Schema Name in following scripts by running the command `db2 -tvf <script_name> -l <log_file_name>` for each of the following:
 - a. `CreateEM_Tables.sql` to create EM tables
 - b. `CreateEM_Hist_Tables.sql` to create EM history tables
 - c. `CreateEM_Ix.sql` to create EM Indexes
 - d. `CreateEM_Fk.sql` to create EM foreign Keys
6. Import the Event Manager data you got it from step 3, using the same <PATH>, then from the command line run the scripts, using the command:
`db2 -tvf ImportEMData.sql -l <log_file_name>`
7. Run the following commands to create Event Manager triggers, and make sure that you choose the same type of trigger with your MDM database:
 - a. Create simple triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_simple.sql -l <log_file_name>`
 - b. Create compound triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_compound.sql -l <log_file_name>`
 - c. **Optional:** Create simple delete triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_delete_simple.sql -l <log_file_name>`
 - d. **Optional:** Create compound delete triggers by running:
`db2 -v -td@ -f CreateEM_Triggers_delete_compound.sql -l <log_file_name>`
8. **Optional:** After you have verified that you have successfully merged the Event Manager database into the database, you can drop the old Event Manager database and schema.

Editing the upgrade properties file:

Manually edit the `UpgradePaths.properties` file to allow the ant script to perform the database upgrade.

About this task

The `UpgradePaths.properties` file enables the `madconfig` target. It contains user information, connection information, and information about tablespace correspondences.

Procedure

1. Navigate to the `MDM_INSTALL_HOME/mds/scripts` directory.
2. Open the `UpgradePaths.properties` file in your preferred text editor.
3. See the instructions in the comments within the file itself for more information about each parameter. The specific parameter names are listed here:
 - **mdm.upgrade.basedir** - The directory you specify here will match the directory you navigated to in Step 1.
 - **rdm.upgrade.basedir** - This parameter is commented out by default. Do not uncomment or change the parameter. It is available to enable future functionality.

- **mdm.ae.locales** - When it was installed, your current installation of InfoSphere MDM was configured to use one or more locales. Specify the same locales here that were specified with the original installation.
- **mdm.ae.industry** - When it was installed, your current installation of InfoSphere MDM was configured to include one or more industry data sets. The values can be banking, financial, telco, or manufacturing. Use the same value or values that were used with the original installation.
- **mdm.ae.triggers** - When it was installed, your current installation of InfoSphere MDM was configured to use either simple triggers, compound triggers, or no triggers. Refer to your installation notes for the types of triggers that are installed in MDM and enter the same value for this parameter. The type you specify here must match what is already configured for MDM.
- **mdm.ae.delete.triggers** - As with **mdm.ae.triggers**, choose compound, simple, or none depending on how your current MDM installation is configured.
- **db.type** - The current release supports DB2.
- **db.subtype** - The current release supports Standard.
- **db.driver.class**
- **db.driver.path**
- **db.jdbc.url**
- **db.uid** - The user ID that is used to connect to the database and to run the upgrade scripts. Ensure that it has sufficient privileges to alter schemas and perform other necessary operations.
- **db.pwd** - The password that corresponds to the user ID you provided.
- **db.name** - The name of the database you to which the scripts will connect, for example, MDMDDB.
- **starting.version.number** - Enter the version of MDM that you will be upgrading. InfoSphere MDM Version 11.3 includes the scripts that are necessary to upgrade from Version 8.5.0. The correct version numbers to use for the **starting.version.number** parameter are 8.5.0, 9.0.1, 9.0.2, 10.0.0, 10.0.1, and 11.0.0.
- **upgrade.path.8.5.0** - Do not edit this value.
- **upgrade.path.9.0.1** - Do not edit this value.
- **upgrade.path.9.0.2** - Do not edit this value.
- **upgrade.path.10.0.0** - Do not edit this value.
- **upgrade.path.10.1.0** - Do not edit this value.
- **upgrade.path.11.0.0** - Do not edit this value.
- **SCHEMA** - Some table names in the upgrade scripts are qualified with the **SCHEMA** name or the **DBUSER** name. The two terms mean the same thing. Some older scripts used the term **DBUSER** while others used the term **SCHEMA**.
- **DBUSER** - See previous note.
- **TABLE_SPACE** - Enter name of the User Data table space name. Provide the value that was used when the MDM database was first created. Refer to the database. The default pool size is 8K.
- **INDEX_SPACE** - Enter the name of Index Table space. Provide the value that was used when the MDM database was first created. Refer to the database.
- **LONG_SPACE** - Enter the name of the log data table space. Provide the value that was used when the MDM database was first created. Refer to the database.

- **TABLE_SPPMD** - Enter the name of the tablespace to contain the new Probabilistic Matching Engine tables. The upgrade process will create the tablespace for you.
- **TABLE_SPPMI** - Enter the name of the tablespace to contain the new Probabilistic Matching Engine indices. The upgrade process will create the tablespace for you.
- **DB_OS** - Enter the desired default value for /IBM/DWLCommonServices/DataBase/OS
- **DB_TYPE** - The current release supports DB2.

Note: The **DB_TYPE** element is part of the configuration of MDM at runtime. The **db.type** parameter mentioned previously identifies the folder in the upgrade scripts for the database you want to upgrade. The two parameters should always refer to the same type of database, for example "Oracle" for Oracle, and "DB2" for DB2, but the specific syntax of each parameter might differ.

- **DB_VERSION** - Enter the desired default value for /IBM/DWLCommonServices/DataBase/version. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **DEFAULT_TIMEZONE** - Enter the desired default value for /IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **TIMEZONE_ENABLED** - The current release supports a value of false.
- **NLS_SYS_LANG** - Enter the desired default value for /IBM/DWLCommonServices/NLS/system_Default_Data_Locale. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **DSRC_CODE** - Enter desired default value for /IBM/ThirdPartyAdapters/EAS/dsrcCode. See the link at the end of this topic for more information about the elements in the Configuration and Management component.
- **EXCLUSIVE_CIENT_SYSTEMS** - Enter desired default value for /IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem. See the link at the end of this topic for more information about the elements in the Configuration and Management component.

4. Save and close the file.

Related tasks:

 Elements in the Configuration and Management component

Running the madconfig target for database upgrade:

After you have completed any prerequisite steps, run the madconfig upgrade-mdm-ae-db target to perform the automatic database upgrade.

About this task

For IBM InfoSphere MDM Advanced Edition, a madconfig target can perform an automatic database upgrade from any version of InfoSphere MDM Server from 8.5 onward. The madconfig target detects your current version of InfoSphere MDM and runs the database upgrade steps required to increment from that version to the latest version of InfoSphere MDM. Although you run the madconfig target from the *MDM_INSTALL_HOME/mds/scripts* directory, the database upgrade scripts that it runs are in the *MDM_INSTALL_HOME/database/Upgrade/* directory.

Procedure

1. At a command-prompt, navigate to the `MDM_INSTALL_HOME/mds/scripts` directory.
2. Run the madconfig target:
 - For Microsoft Windows: `madconfig upgrade-mdm-ae-db -l logfilename`
 - For Linux and UNIX: `madconfig.sh upgrade-mdm-ae-db -l logfilename`

Note: You can use the *logfilename* argument to specify a location and name for the log file. If you do not specify a *logfilename*, the log messages are written out to the console.

3. To review the process, see the log file created by the madconfig target.

Related concepts:

 Manually installing the madconfig utility

 madconfig utility targets for InfoSphere MDM

Viewing a log of the upgrade process:

As the ant script runs, it directs all log information into a single file.

About this task

If you specified a log file name and location when you ran `madconfig upgrade-mdm-ae-db`, look for the log file in the directory you indicated.

Procedure

1. Navigate to the location you specified for the *logfilename*.
2. Open the file.

Enabling non-case-sensitive search capability:

The automatic upgrade process does not include an option to enable non-case-sensitive search for DB2. If you want to have the ability to do non-case-sensitive search, you can manually enable the capability after you have finished the automatic upgrade.

Procedure

1. Navigate to the `MDM_INSTALL_HOME/database/Full/DB2/Standard/ddl` directory.
2. Run the following script: `db2 -tvf Insensitive_search_enabled.sql -l <log_file_name>`

Manually upgrading the database on DB2:

You can use a set of scripts to do a manual upgrade of the MDM database on DB2.

About this task

The manual upgrade process involves running a series of version-specific scripts for upgrading the MDM database. For example, if you are currently running InfoSphere MDM Server version 9.0.1, you must run each of the following scripts in sequence:

1. A script to upgrade your database from version 9.0.1 to version 9.0.2.
2. A script to upgrade your database from version 9.0.2 to version 10.1.

3. A script to upgrade your database from version 10.1 to version 11.0.
4. A script to upgrade your database from version 11.0 to version 11.3.

Each script occupies a folder within a version-specific directory within `MDM_INSTALL_HOME/database/Upgrade/`. For example, the folder `MDM_INSTALL_HOME/database/Upgrade/9.0.2/Level-I` contains scripts that allow you to upgrade from version 9.0.1 to version 9.0.2. Note that the directories containing the scripts are named for the *target* version.







Certain directories contain both a Level-I directory and a Level-II directory. The scripts within Level-II directories can allow you to skip certain intermediate versions as you upgrade the database.

Be sure to read the relevant documentation for each release before you run the corresponding upgrade script. See the list of links at the end of this topic.

Use the following list to determine which scripts to run and the order to run them:

- If you are currently running version 8.5.0, run the scripts within the following directories, in sequence: 9.0.1/Level-II/DB2/Standard/ddl then 9.0.2/Level-I/DB2/Standard/ddl then 10.1.0/Level-II/DB2/Standard/ddl then 11.0.0/Level-I/DB2/Standard/ddl then 11.3.0/Level-I/DB2/Standard/ddl.
- If you are currently running version 9.0.1, run the scripts within the following directories, in sequence: 9.0.2/Level-I/DB2/Standard/ddl then 10.1.0/Level-II/DB2/Standard/ddl then 11.0.0/Level-I/DB2/Standard/ddl then 11.3.0/Level-I/DB2/Standard/ddl.
- If you are currently running version 9.0.2, run the scripts within the following directories, in sequence: 10.1.0/Level-II/DB2/Standard/ddl then 11.0.0/Level-I/DB2/Standard/ddl then 11.3.0/Level-I/DB2/Standard/ddl.
- If you are currently running version 10.0.0, run the scripts within the following directories, in sequence: 10.1.0/Level-I/DB2/Standard/ddl then 11.0.0/Level-I/DB2/Standard/ddl then 11.3.0/Level-I/DB2/Standard/ddl.
- If you are currently running version 10.1.0, run the scripts within the following directories, in sequence: 11.0.0/Level-I/DB2/Standard/ddl then 11.3.0/Level-I/DB2/Standard/ddl.
- If you are currently running version 11.0.0, run the script within the following directory: 11.3.0/Level-I/DB2/.

Related tasks:

-  [Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0](#)
-  [Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0](#)
-  [Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0](#)
- [“Manually upgrading the database on DB2 from 11.0 to 11.3” on page 9](#)

Manually upgrading the database on DB2 from 11.0 to 11.3:

You can use a set of scripts to do a manual upgrade of the MDM database on DB2 from 11.0 to 11.3.

Procedure

1. Navigate to the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl* directory.
2. In the *ImpReqDataCfgMgr.sql* and *QryUpdCfgMgrData.sql* scripts, replace the <SCHEMA> placeholder with a schema name or owner with necessary privileges. This value must be in uppercase.
3. In the *QryUpdCfgMgrData.sql* script, replace the following tokens with values that are appropriate to your own installation:
 - **DB_OS** - Enter the desired default value for */IBM/DWLCommonServices/Database/OS*.
 - **DB_TYPE** - Use the value *DB2*.
 - **DB_VERSION** - Enter the desired default value for */IBM/DWLCommonServices/Database/version*
 - **DEFAULT_TIMEZONE** - Enter the desired default value for */IBM/DWLCommonServices/MultiTimeZoneDeployment/defaultTimeZone*
 - **TIMEZONE_ENABLED** - Use the value *false*.
 - **NLS_SYS_LANG** - Enter the desired default value for */IBM/DWLCommonServices/NLS/system_Default_Data_Locale*
 - **DSRC_CODE** - Enter desired default value for */IBM/ThirdPartyAdapters/EAS/dsrcCode*
 - **EXCLUSIVE_CIENT_SYSTEMS** - Enter desired default value for */IBM/ThirdPartyAdapters/EAS/exclusiveSourceSystem*
4. Connect to the database that you are upgrading.
5. Replace the <SCHEMA> tag in the scripts in the following folders with a schema name or owner, in uppercase. *CODE_LANG* in the folder name is the language that is currently installed. If you are loading English data only, set *CODE_LANG* to *en*. Otherwise, set *CODE_LANG* to the language code of the language you are loading.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_req_CODE_LANG/data*
 - *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_CODE_LANG/data-industry*
 - *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_opt_CODE_LANG/data*
6. From the command line, to load the common data, English code table data, and configure the table in the language that you selected, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_req_CODE_LANG/data* directory. Use the command: *db2 -tvf script_name -l log_file_name*. For *script_name*, use the following scripts:
 - a. The *insert.sql* script inserts new data
 - b. The *update.sql* script updates existing table data.
 - c. The *delete.sql* deletes obsolete table data.
 7. From the command line, to load the English industry data, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/ddl/data_en/data-industry* directory. Use the command: *db2 -tvf script_name -l log_file_name*. For *script_name*, use the following scripts:
 - a. The *insert.sql* script inserts new data

- b. The update.sql script updates existing table data.
 - c. The delete.sql deletes obsolete table data.
8. If you are loading industry data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/dd1/data_CODE_LANG/data-industry* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The insert.sql script inserts new data
 - b. The update.sql script updates existing table data.
 - c. The delete.sql deletes obsolete table data.
9. If you are loading code table data for a language other than English, from the command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/Standard/dd1/data_opt_CODE_LANG/data* directory. Use the command: `db2 -tvf script_name -l log_file_name`. For *script_name*, use the following scripts:
 - a. The insert.sql script inserts new data
 - b. The update.sql script updates existing table data.
 - c. The delete.sql deletes obsolete table data.
10. Repeat the previous two steps for each non-English language that is installed.

Note: If you have multiple deployments, before you run the following script, change the SQL clause `DEPLOYMENT_ID <> 1000` to `DEPLOYMENT_ID = DEPLOYMENT_ID` where the *DEPLOYMENT_ID* is the ID of the instance you want to upgrade.

11. From the command line, run the command: `db2 -tvf ImpReqDataCfgMgr.sql -l log_file_name`.
12. From the command line, run the command: `db2 -tvf QryUpdCfgMgrData.sql -l log_file_name`.

Related tasks:

“Manually upgrading the database on DB2” on page 48

Upgrading an existing DB2 database for z/OS:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for DB2 on z/OS. For information about upgrading the database for previous releases, see the links at the end of this topic.

Procedure

1. To connect to DB2 for OS/390 and z/OS server from a DB2 Linux, UNIX, or Windows server, DB2 Connect must be installed on the server you are connecting from. Enterprise editions contain DB2 Connect. To connect to the database, enter the following information:
 - Host name or IP address
 - DB2 port (default = 446)
 - Database name
 - Subsystem location name
 - User ID and password
2. On the client server, catalog the database by running the following commands in this order:

- a. db2 catalog tcpip node *YOUR_NODE_NAME* remote *Z/OS_HOSTNAME/*
IP_ADDRESS server *Z/OS_PORT*
- b. db2 catalog db *YOUR_DB_NAME* at node *YOUR_NODE_NAME* authentication
dcs
- c. db2 catalog dcs db *YOUR_DB_NAME* as *Z/OS_LOCATION_NAME*
3. If you are using this user ID to connect to the DB2 subsystem for the first
time, rebind the packages for the database by following either of these steps:
 - a. Rebind all the packages by running the command: db2rbind *YOUR_DB_NAME*
all /u *USER_NAME* /p *PASSWORD* /r any
 - b. Rebind individual packages by running the command: db2 bind
package_name
4. Navigate to the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/*
ZOS/dd1 directory.
5. Change the placeholder tags in the all of the scripts:
 - Set <SCHEMA> to a schema name or owner in uppercase.
 - Set <USER_ACCOUNT> to an account with user authority.
6. In the folder name, *CODE_LANG* is the language that is currently installed:

Note: If any of the directories are empty, no data was generated and you do
not need to run the scripts.






- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/*
data_req_CODE_LANG/data
- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/*
data_CODE_LANG/data-industry
- *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/*
data_opt_CODE_LANG/data

If you are loading English data only, set *CODE_LANG* to en. Otherwise, set
CODE_LANG to the language code of the language you are loading.


7. To load the common data from the command line, run the scripts in the
MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/
data_req_en/data directory. Use the command: db2 -tvf *script_name* -l
log_file_name:
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
8. To load the English industry data from the command line, run the scripts in
the *MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/*
data_en/data-industry directory. Use the command: db2 -tvf *script_name* -l
log_file_name:
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
9. If you are loading industry data for a language other than English, from the
command line, run the scripts in the *MDM_INSTALL_HOME/database/Upgrade/*
11.3.0/Level-I/DB2/ZOS/dd1/data_CODE_LANG/data-industry directory. Use
the command: db2 -tvf *script_name* -l *log_file_name*
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.

10. If you are loading code table data for a language other than English, from the command line, run the scripts in the `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/DB2/ZOS/dd1/data_opt_CODE_LANG/data` directory. Use the command: `db2 -tvf script_name -l log_file_name`:
 - a. The `insert.sql` script inserts new data.
 - b. The `update.sql` script updates existing table data.
 - c. The `delete.sql` script deletes obsolete table data.
11. Repeat the previous step for each non-English language that is installed.
12. From the command line, run the command: `db2 -tvf ImpReqDataCfgMgr_zos.sql -l log_file_name`.
13. From the command line, run the command: `db2 -tvf QryUpdCfgMgrData_zos.sql -l log_file_name`.

Related tasks:

-  [Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0](#)
-  [Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0](#)
-  [Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0](#)
-  [Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0](#)

Related information:

-  [Manual installation of the physical MDM database](#)

Upgrading an existing Oracle database:

Follow the steps to upgrade the Version 11.0 InfoSphere MDM Server database to the Version 11.3 InfoSphere MDM database for Oracle. For information about upgrading the database for previous releases, see the links at the end of this topic.

Before you begin

Read the content for the link at the end of this topic about manual installation of the database.

For Oracle, the use of table spaces for data, indexes and LOBs, such as CLOBs or XML, to improve database performance is supported. Look for the following placeholders in the scripts: `<TABLE_SPACE>`, `<INDEX_SPACE>`, and `<LONG_SPACE>`.

Procedure

1. Replace the placeholder `<SCHEMA>` in the scripts in the following folders with a schema name or owner. Ensure that the name or owner is in uppercase. `CODE_LANG` in the folder name is the language that is currently installed.

Note: If any of the directories are empty, no data was generated and you do not need to run the scripts.

- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/Oracle/Standard/dd1/data_req_CODE_LANG/data`
- `MDM_INSTALL_HOME/database/Upgrade/11.3.0/Level-I/Oracle/Standard/dd1/data_CODE_LANG/data-industry`

- *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_opt_CODE_LANG/data
2. To load the common data, to load the English code table data, and to configure the table in the language that you selected, run the following script in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_req_CODE_LANG/data directory from the command line. Use the command: `sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @<script_name> >> <log_file_name>`
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.

If you are loading English data only, set *CODE_LANG* to en. Otherwise, set *CODE_LANG* to the language code of the language you are loading.
 3. To load the English industry data, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_en/data-industry directory from the command line. Use the command: `sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @<scriptname> >> <log_file_name>`
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
 4. If you are loading data for a language other than English, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_<CODE_LANG>/data-industry directory from the command line:
 - a. The insert.sql script inserts new data.
 - b. The update.sql script updates existing table data.
 - c. The delete.sql script deletes obsolete table data.
 5. If you are loading code table data for a language other than English, run the scripts in the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/data_opt_CODE_LANG/data directory from the command line: `sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @<scriptname> >> <log_file_name>`
 - The insert.sql script inserts new data.
 - The update.sql script updates existing table data.
 - The delete.sql script deletes obsolete table data.
 6. Repeat the previous two steps for each non-English language that is installed.
 7. Navigate to the *MDM_INSTALL_HOME*/database/Upgrade/11.3.0/Level-I/Oracle/Standard/ddl/ directory.
 8. Run the following command: `sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @ImpReqDataCfgMgr_ora.script >> <log_file_name>`
 9. Run the following command: `sqlplus <SCHEMA>/<SCHEMAPASSWORD>@<DBNAME> @QryUpdCfgMgrData_ora.sql >> <log_file_name>` The QryUpdCfgMgrData_ora.sql script populates the Configuration Manager values based on previous InfoSphere MDM version 11 deployment values.

Related tasks:

- ➡ Upgrading from InfoSphere MDM Server version 8.5 to version 9.0.1
- ➡ Upgrading from InfoSphere MDM Server version 9.0.1 to version 9.0.2
- ➡ Upgrading from InfoSphere MDM Server version 9.0.2 to version 10.0.0
- ➡ Upgrading from InfoSphere MDM Server version 9.0.2 to 10.1.0

➡ Upgrading from InfoSphere MDM Server version 10.0.0 to version 10.1.0

➡ Upgrading from InfoSphere MDM Server version 10.1.0 to version 11.0.0

Related information:

➡ Manual installation of the physical MDM database

Creating new tablespaces

After MDM has been upgraded, but before creating the Initiate database tables, you might need to run a script to create tablespaces.

About this task

The current release of IBM InfoSphere MDM supports upgrading from releases prior to 11.0. Because of changes to the MDM database, if you are upgrading from a release prior to 11.0, you must run a SQL script that creates new tablespaces. If you are upgrading from 11.0, you can skip the following steps.

Procedure

1. Navigate to *MDM_INSTALL_HOME*/database/Upgrade/11.0.0/Level-I/DB2/Standard/ddl/
2. Open the CreateTS.sql script for editing.
3. Update the following values with the values of the actual tablespaces you intend to use for MDM:
 - <TBS4K>
 - <TBS8K>
 - <TBS16K>
 - <LONG_SPACE>
4. Run CreateTS.sql.

Creating the Initiate database tables

As part of upgrading from IBM InfoSphere MDM Server to IBM InfoSphere Master Data Management Advanced Edition, create the Initiate database tables.

About this task

Important: You can skip the steps in this topic if you are upgrading to version 11.3 from version 11.0. In that case, the Initiate tables have already been installed as part of your 11.0 installation.

The latest version of InfoSphere MDM combines the IBM InfoSphere MDM Server with the Initiate Master Data Service product. The combined product relies on the database tables of both component products. Therefore, you must create the Initiate database tables as part of your upgrade process.

Procedure

1. At a command-line prompt, go to the scripts directory of the installation:
MDM_INSTALL_HOME/mds/scripts.
2. Run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig.bat update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information:

- a. Database type

- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password
- g. Database schema - You must supply a schema name if your database is DB2.

Note: In Windows, System DSN will show all the datasources created. In Linux environments, the *MDM_INSTALL_HOME*/mds/odbc.ini file will show the datasources.

3. When the script asks whether you want to re-create the existing data source, enter y.
4. The script prompts for the database schema. For IBM DB2 for z/OS, you can submit an alternative schema or enter nothing to use the default schema. For other database types, enter nothing.
5. Take a moment to verify that the settings you entered have been applied:
 - a. Navigate to *MDM_INSTALL_HOME*/mds/conf
 - b. Open the com.ibm.mdm.mds.jni.cfg configuration file and verify the settings.
 - c. Save and close the file.
 - d. Open the com.ibm.mdm.mds.jdbc.cfg configuration file and verify the settings.
 - e. Save and close the file.
6. Set the library path to point to *MDM_INSTALL_HOME*/mds/lib. An incorrect setting can result in a "madhub create" error when you run the bootstrap process. Use the library path variable that is appropriate for your operating system. To see the current setting, use the following commands:
 - Linux or Solaris - export LD_LIBRARY_PATH
 - AIX - export \$libpath
 - Microsoft Windows - set PATH
7. Confirm that tablespaces are created. The process pre-pends the last three characters of your schema to "ZMDS". For example, if your schema is M5Z10M65, the database identifier will be M65ZMDS.
8. In the *MDM_INSTALL_HOME*/mds/sql directory, edit the mpihub.sto file so that the database identifier is correct for each tablespace.
9. From the *MDM_INSTALL_HOME*/mds/scripts directory, run the madconfig bootstrap_datasource command:
 - For Microsoft Windows: madconfig.bat bootstrap_datasource
 - For Linux and UNIX: madconfig.sh bootstrap_datasource
10. To make sure that everything stays in sync, run the madconfig update_instance_datasource command again, but this time from the native.war directory within the WebSphere Application Server. First, navigate to *WAS_PROFILE_HOME*\installedApps\YOUR_CELL_NAME\MDM-native-IDENTIFIER.ear\native.war\scripts directory. Then run the madconfig update_instance_datasource command:
 - For Microsoft Windows: madconfig update_instance_datasource
 - For Linux and UNIX: madconfig.sh update_instance_datasource

The command prompts for the following information. Provide the information for the Initiate database that you want to upgrade:

- a. Database type
- b. Database host
- c. Database port
- d. Database name
- e. Database user
- f. Database password

Results

The first script updates the following configuration files within the *MDM_INSTALL_HOME/mds/conf* directory:

- *com.ibm.mdm.mds.jni.cfg*
- *com.ibm.mdm.mds.jdbc.cfg*

The second script creates the Initiate tables.

Updating the data sources

As part of the upgrade process, edit the user, password, and other database information for the data sources.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to **Resources > JDBC > Data sources**.
3. Select the data source **DWLConfig**.
4. Edit the database information under **Common and required data source properties** to point to the upgraded database schema.
5. Click on **Custom properties** and edit the user and password of the upgraded database schema. If your DB user name and schema name are different complete these additional steps:
 - a. Within **Custom properties**, click **New**.
 - b. For the **Name** field, enter *currentSchema*.
 - c. For the **Value** field, enter the name of your schema.
6. Also click on **JAAS - J2C authentication data** and edit all of the aliases in the list. (If you did not deploy using MQ, you do not need to edit the alias for *Node01/application-name-MQ-User*.) For each alias, modify the user name and password.
7. Repeat steps 4, 5, and 6 for the data source **DWLCustomer**.
8. Select the data source **MDM**.
9. Click on **Customer properties** to edit the database information of the upgraded database.

Updating the messaging engine

If your Version 10.1 installation is configured to use WebSphere Embedded Messaging, update the messaging engine as part of the upgrade process.

Procedure

1. Log on to the WebSphere Application Server Administration Console.

2. Navigate to **Service integration > Buses > MDM.SIB.application-name > Messaging Engine > node-name.application-name-MDM.SIB.application-name > Message Store**.
3. Edit the **Schema name** field.
4. Stop the application server.
5. Drop all of the messaging engine tables. The messaging engine table names all begin with **SIB**:
 - SIB000
 - SIB001
 - SIB002
 - SIBCLASSMAP
 - SIBKEYS
 - SIBLISTING
 - SIBOWNER
 - SIBOWNER0
 - SIBXACTS
6. Restart the application server.

Updating the install name in the appsoftware table

Update the name of the installation within the appsoftware table.

Procedure

1. Log in to the WebSphere Application Server Administration Console.
2. Navigate to the Business Level Application for InfoSphere MDM: **Application > Application Types > Business-level applications**. Then look for the application with the description **MDMBLA**.
3. Open the composition unit.
4. Open the OSGi Administrative Console.
5. Note the bundle name: `com.ibm.mdm.hub.server-XXXXXXX`.
6. Log into the MDM database.
7. Select * from <SCHEMA>.APPSOFTWARE WHERE VERSION = '11.3.0'. After you run the SQL statement, note the name of the application in the NAME field. It will say **com.ibm.mdm.hub.server**.
8. Update the appsoftware table to add `-[name]` to the `com.ibm.mdm.hub.server` record. For example, if `XXXXXXX` is `E001`, then to update to InfoSphere MDM Version 11.3:
 - DB2: `update appsoftware set NAME='com.ibm.mdm.hub.server-E001', last_update_dt = current timestamp where APPLICATION_ID = 1001;`
 - Oracle: `update appsoftware set NAME='com.ibm.mdm.hub.server-E001', last_update_dt = sysdate where APPLICATION_ID = 1001;`
 - Microsoft SQL Server: `insert into APPSOFTWARE values (10000, 'com.ibm.mdm.hub.server-E001', '11.3.0', CURRENT_TIMESTAMP, 'unknown');`

Recreating ILOG rules

The new OSGi architecture means that the embedded ILOG engine is no longer available. Instead the ILOG engine (now called ODM) can be called remotely. Recreate existing ILOG rules as ODM rules.

About this task

For information about creating ODM rules, follow the link at the end of this topic.

Procedure

To determine whether you use ILOG rules with your previous version of InfoSphere MDM Server, run the following scripts.

1. For any ILOG-based Behavioral Extensions rules:

```
SELECT * FROM EXTENSIONSET
WHERE (JAVA_CLASS_NAME IS NULL OR JAVA_CLASS_NAME = '')
AND (RULE_SET_NAME IS NOT NULL OR RULE_SET_NAME != '')
```

2. For any ILOG-based External Rules:

```
SELECT * FROM RULEENGINEIMPL
WHERE EXT_RULE_IMPL_ID
IN (SELECT EXT_RULE_IMPL_ID FROM EXTRULEIMPLEM
WHERE EXT_RULE_TP_CODE='R')
```

Related information:

 Exporting business rules to IBM Operational Decision Manager

Verifying your upgrade

You finish the upgrade process by verifying that the new system is functional.

Procedure

1. Verify that the new InfoSphere MDM operational server can access the upgraded database.
2. Ensure that your customizations from previous releases are available in the new environment.

Notices and trademarks

This information was developed for products and services offered in the U.S.A.

Notices

This information was developed for products and services offered in the U.S.A. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Privacy Policy Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's name, user name, password, profile name, or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration, or web page identification that the user tried to load prior to login. These cookies can be disabled, but disabling them will also likely eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at www.ibm.com/privacy and IBM's Online Privacy Statement at www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at www.ibm.com/software/info/product-privacy.

General statement regarding product security

IBM systems and products are designed to be implemented as part of a comprehensive security approach that might require the use of other systems, products, or services to be most effective. A comprehensive security approach must be reviewed whenever systems and products are added to your environment.

No IT system or product can be made completely secure, and no single product or security measure can be completely effective in preventing improper access.

IT system security involves protecting systems and information through prevention, detection, and response to improper access from within and outside your enterprise. Improper access can result in information that is altered, destroyed, or misappropriated, or can result in misuse of your systems to attack others.

IBM does not warrant that systems and products are immune from the malicious or illegal conduct of any party.

IBM does not believe that any single process can be completely effective in helping identify and address security vulnerabilities. IBM has a multilayered approach:

- An ongoing, internal initiative promotes consistent adoption of security practices in development of products and services, with the goal of continually improving the quality and security characteristics of all IBM products and services. This initiative is described in the IBM Redguide *Security in Development: The IBM Secure Engineering Framework*, which contains public information about software development practices from IBM.
- Tests and scans of IBM products use various IBM technologies to proactively identify and remediate defects and vulnerabilities, including high or greater criticality vulnerabilities. Remediation takes place within IBM-defined response target timeframes for analysis, impact assessment, and fix delivery.
- The IBM Product Security Incident Response Team (PSIRT) manages the receipt, investigation, and internal coordination of security vulnerability information that is related to IBM offerings. The IBM PSIRT team acts as a focal point that security researchers, industry groups, government organizations, vendors, and customers can contact through the IBM PSIRT portal to report potential IBM product security vulnerabilities. This team coordinates with IBM product and solutions teams to investigate and identify the appropriate response plan.
- A global supply-chain integrity program and framework provide buyers of IT products with a choice of accredited technology partners and vendors in the Open Group Trusted Technology Forum (OTTF).

Because security of computer systems and computer software is a very complex issue, IBM does not provide information about development practices for individual products other than what is found in standard product documentation or as published through IBM's public activities.

Public information about software development practices recommended by IBM is documented in the IBM Secure Engineering Framework. This information is a compilation of practices from across IBM business units and development teams.

In most cases, published vulnerabilities are documented at timely intervals through IBM Security Bulletins that include the associated Common Vulnerability Scoring System (CVSS) base score. In some cases, IBM might contact customers directly and discreetly regarding specific vulnerabilities.

Customers who want to further validate the vulnerability of IBM Software beyond the assessments that are performed internally by IBM are welcome to conduct their own scans against licensed software. They may use the tool of their choice within the existing software licensing terms. For example, scanning is acceptable, but reverse compiling or reverse engineering IBM Software is not authorized except as expressly permitted by law without the possibility of contractual waiver.

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Index

C

customer support
 contacting 87

D

databases
 new 2
 reusing 2
 upgrading 2

I

Initiate 21
installing 2

L

legal notices 79

M

Master Data Engine
 pre-upgrade tasks 24
Master Data Service 21

S

software services
 contacting 87
support
 customer 87

T

trademarks
 list of 79

U

upgrade 21, 24
 Master Data Engine database 24
 pre-upgrade tasks 24
upgrades
 database worksheet 25

W

worksheets
 database upgrade 25

Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

Table 2. IBM resources

Resource	Description and location
Product documentation for InfoSphere MDM	You can search and browse across all the InfoSphere MDM documents at http://www.ibm.com/support/knowledgecenter/SSWSR9_11.3.0 .
Product documentation for InfoSphere MDM Custom Domain Hub, including InfoSphere MDM Reference Data Management	You can search and browse across all the InfoSphere MDM Custom Domain Hub documents at http://www.ibm.com/support/knowledgecenter/SSLSQH_11.3.0 .
IBM Support Portal	You can customize support information by choosing the products and the topics that interest you at www.ibm.com/support/ .
Software services	You can find information about software, IT, and business consulting services, on the solutions site at www.ibm.com/businesssolutions/ .
My IBM	You can manage links to IBM web sites and information that meet your specific technical support needs by creating an account on the My IBM site at www.ibm.com/account/ .
Training and certification	You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at www.ibm.com/software/sw-training/ .
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/ .

Providing feedback

The following table describes how to provide feedback to IBM about products and product documentation.

Table 3. Providing feedback to IBM

Type of feedback	Action
Product feedback	You can provide general product feedback through the Consumability Survey at https://www.ibm.com/survey/oid/wsb.dll/studies/consumabilitywebform.htm .

Table 3. Providing feedback to IBM (continued)

Type of feedback	Action
Documentation feedback	<p>To comment on the product documentation:</p> <ul style="list-style-type: none"> • Click the Feedback link on the bottom of any topic in IBM Knowledge Center • Online reader comment form: www.ibm.com/software/data/rcf/ • E-mail: comments@us.ibm.com



Printed in USA

GI13-2670-01

