



Informatica<sup>®</sup> MDM Registry Edition  
10.2

# Installation and Configuration Guide

Informatica MDM Registry Edition Installation and Configuration Guide

10.2

December 2020

© Copyright Informatica LLC 2010, 2020

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2020-12-11

# Table of Contents

<b>Preface</b> .....	<b>6</b>
Informatica Resources. ....	6
Informatica Network. ....	6
Informatica Knowledge Base. ....	6
Informatica Documentation. ....	6
Informatica Product Availability Matrices. ....	7
Informatica Velocity. ....	7
Informatica Marketplace. ....	7
Informatica Global Customer Support. ....	7
<b>Chapter 1: Installation Overview</b> .....	<b>8</b>
MDM Registry Edition Installation. ....	8
Components of MDM Registry Edition. ....	8
Installation Tasks. ....	9
<b>Chapter 2: Installing MDM Registry Edition</b> .....	<b>10</b>
Pre-Installation Tasks. ....	10
Verify the Minimum System Requirements. ....	10
Install Java. ....	11
Set the Environment Variables. ....	11
Install the .NET Framework. ....	11
Install the ODBC Drivers. ....	11
Extract the Installer Files on Windows. ....	11
Install the Microsoft Visual C++ Redistributable Package for Visual Studio 2015 (x64). ....	11
Extract the Installer Files on UNIX. ....	12
Set Up the X Window Server on UNIX. ....	12
Installing MDM Registry Edition in Graphical Mode. ....	12
Step 1. Generate the License Information File. ....	12
Step 2. Install the MDM Registry Edition Components. ....	13
Installing MDM Registry Edition in Console Mode. ....	14
Step 1. Generate the License Information File. ....	14
Step 2. Configure a Parameter File. ....	15
Step 3. Install the MDM Registry Edition Components. ....	15
List of Parameters. ....	15
<b>Chapter 3: Post-Installation Tasks</b> .....	<b>19</b>
Post-Installation Tasks Overview. ....	19
Database Objects. ....	20
Configuring Oracle. ....	20
Step 1. Configure the Optimizer Mode. ....	20

Step 2. Create Users and Tables. . . . .	21
Step 3. Define an ODBC Data Source. . . . .	22
Step 4. Configure the Oracle Instance Parameters. . . . .	24
Configuring Microsoft SQL Server. . . . .	24
Step 1. Create a User. . . . .	24
Step 2. Define an ODBC Data Source. . . . .	25
Step 3. Create Tables. . . . .	26
Step 4. Create Microsoft SQL Server API Components. . . . .	27
Configuring IBM DB2 UDB. . . . .	27
Step 1. Set the Database Configuration Parameters. . . . .	28
Step 2. Create a User. . . . .	28
Step 3. Create Tables. . . . .	28
Step 4. Define an ODBC Data Source. . . . .	29
Step 5. Install the User-Defined Functions. . . . .	31
Step 6. Install IBM DB2 Driver for ODBC and CLI. . . . .	31
Step 7. Set the Environment Variables. . . . .	31
Configuring IBM DB2 on z/OS. . . . .	32
Testing the Database Configuration. . . . .	32
Testing the MDM Registry Edition Installation. . . . .	33
Test the MDM Registry Edition Installation in Graphical Mode. . . . .	33
Test the MDM Registry Edition Installation in Console Mode. . . . .	34
Test the ssapid Utility on UNIX. . . . .	34
Increasing the Number of Database Connections. . . . .	35
Configuring Security. . . . .	35
Configuring MDM Registry Edition Security Server. . . . .	35
Starting the MDM Registry Edition Components. . . . .	37
Configure the Environment Variables. . . . .	37
Start MDM Registry Edition. . . . .	37
Stopping the MDM Registry Edition Server. . . . .	38
<b>Chapter 4: Troubleshooting. . . . .</b>	<b>39</b>
Troubleshooting MDM Registry Edition Installation. . . . .	39
Troubleshooting IBM DB2 UDB. . . . .	39
<b>Chapter 5: Upgrading MDM Registry Edition. . . . .</b>	<b>42</b>
Upgrade Overview. . . . .	42
Upgrading the MDM Registry Edition Components in Console Mode. . . . .	42
List of Parameters. . . . .	43
Disabling Authorization. . . . .	46
<b>Chapter 6: Uninstalling MDM Registry Edition. . . . .</b>	<b>47</b>
Uninstallation Overview. . . . .	47
Uninstalling MDM Registry Edition in Graphical Mode. . . . .	47

Uninstalling MDM Registry Edition in Console Mode. . . . . 48

**Index. . . . . 49**

# Preface

The *Informatica MDM Registry Edition Installation and Upgrade Guide* is written for database administrators, system administrators, and implementers who are responsible for installing and setting up Informatica MDM Registry Edition. This guide assumes that you have knowledge of operating systems and database environments.

## Informatica Resources

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

### Informatica Network

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

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

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

### Informatica Knowledge Base

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

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at [KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com).

### Informatica Documentation

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

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

## Informatica Product Availability Matrices

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

## Informatica Velocity

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

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

## Informatica Marketplace

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

## Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

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

# CHAPTER 1

## Installation Overview

This chapter includes the following topics:

- [MDM Registry Edition Installation, 8](#)
- [Components of MDM Registry Edition, 8](#)
- [Installation Tasks, 9](#)

## MDM Registry Edition Installation

MDM Registry Edition searches identity data to identify matches accurately regardless of language, format, structure, location, duplication, omissions, or errors. MDM Registry Edition contains server components, client components, and standard populations. Before you install MDM Registry Edition, review the system requirements, and perform the pre-installation tasks.

You can install the server components, client components, and standard populations on the same computer or on separate computers. If you plan to install the server components and client components on separate computers, you must first install the server components. You can also install the server components and client components on the computer on which you installed the database.

After you install the components of MDM Registry Edition, perform the post-installation tasks that are specific to your source and target database.

## Components of MDM Registry Edition

MDM Registry Edition consists of the following components:

### **Server Components**

The server components include the Search Server, Console Server, Rulebase Server, Connection Server, core modules, and optional servers. The core modules include System Loader, Table Loader, and Update Synchronizer.

### **Client Components**

The client components include the Console Client, Search Client, SDF Wizard, Custom Search Client, and optional components. The optional components include examples of client programs for different programming languages.



## Data Director Components

The Informatica Data Director package contains all the Informatica Data Director components. For information about configuring Informatica Data Director, see *Informatica MDM Registry Edition Data Director Installation Guide*.

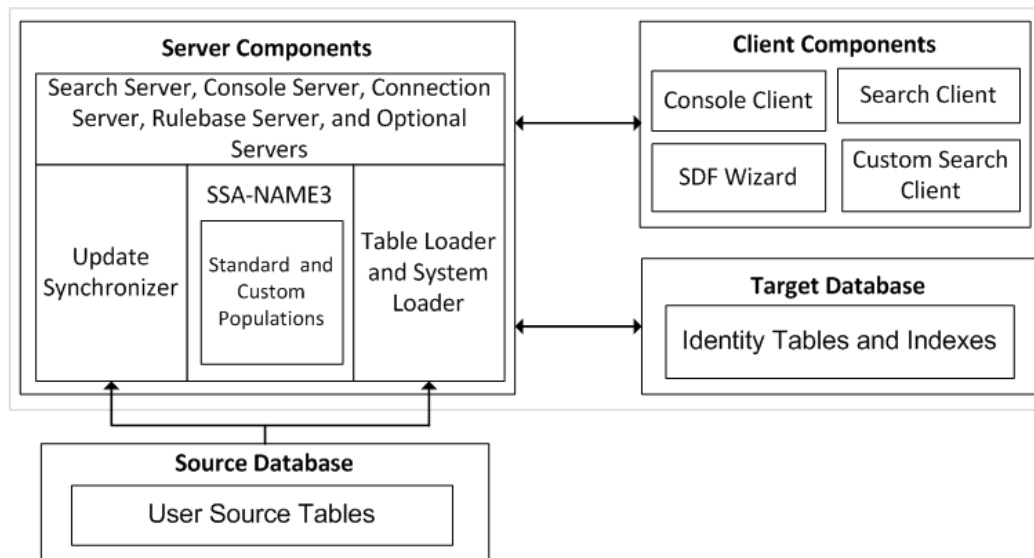
## Standard Population Files

A population file contains SSA-NAME3 key-building algorithms, search strategies, and matching schemes. MDM Registry Edition includes more than 60 standard populations that cover different countries, languages, and regions. You can select the standard population based on the data that you plan to search and match.

## Source and Target Database

A source database contains the user data in the User Source Tables (UST). A target database contains identity tables (IDT) and identity indexes (IDX) that MDM Registry Edition creates and updates.

The following figure shows the architecture of the MDM Registry Edition components:



# Installation Tasks

To install MDM Registry Edition, perform the following tasks:

1. Complete the pre-installation tasks to ensure that you can successfully run the MDM Registry Edition installer.
2. Generate the license information file by using the MDM Registry Edition installer.
3. Send the license information file to the Informatica Global Customer Support. Based on the license information file, a representative from Global Customer Support creates a license key file and sends it to you.
4. Install the components of MDM Registry Edition.
5. Perform the post-installation tasks.

## CHAPTER 2

# Installing MDM Registry Edition

This chapter includes the following topics:

- [Pre-Installation Tasks, 10](#)
- [Installing MDM Registry Edition in Graphical Mode, 12](#)
- [Installing MDM Registry Edition in Console Mode, 14](#)

## Pre-Installation Tasks

Before you install MDM Registry Edition, set up the machine to meet the requirements to install and run MDM Registry Edition. If the machine does not meet the pre-installation requirements, the installation might fail.

### Verify the Minimum System Requirements

Verify that the machine meets the hardware and software requirements to install MDM Registry Edition.

The following table describes the minimum system requirements to install and run the MDM Registry Edition server components:

Component	Minimum requirement
CPU Cores	4
RAM	8 GB

The following table describes the minimum system requirements to install and run the MDM Registry Edition client components:

Component	Minimum requirement
CPU Cores	1
RAM	2 GB

The hardware requirements for an MDM Registry Edition implementation depend on the search strategies and the number of identity tables, identity indexes, and concurrent users that you plan to use.

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:

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

## Install Java

Install Java Development Kit (JDK) or Java Runtime Environment (JRE) version 1.8.

## Set the Environment Variables

The following table describes the environment variables that you must configure if the variables are not already configured:

Variable	Description
PATH	Include the location of the <code>bin</code> directory of the JDK or JRE installation.
JAVA_OPTS	Optional. Additional parameters that you want to pass to the JRE.

## Install the .NET Framework

On Windows, install the .NET Framework version 2 or later.

## Install the ODBC Drivers

MDM Registry Edition uses the ODBC API to access the database.

Unless you use Oracle, install and configure the ODBC drivers for your database. MDM Registry Edition does not use the Oracle ODBC driver and uses custom ODBC drivers for Oracle, `ssaoci9.dll` on Windows and `libssaoci9.so` on UNIX. The custom drivers do not require any further configuration.

**Note:** You do not have to install the ODBC drivers if you plan to install only the MDM Registry Edition client components.

## Extract the Installer Files on Windows

On Windows, the installer files are compressed and distributed as a zip file.

Use a zip utility to extract the installer files to a directory on your machine.

You can extract the installer files in the following ways:

- Installation DVD. Copy the Informatica zip file from the installation DVD to a directory on your machine and then extract the installer files, or extract the installer files directly from the DVD to a directory on your machine.
- FTP download. Download the Informatica installation zip file from the Informatica Electronic Software Download (END) site to a directory on your machine and extract the installer files.

## Install the Microsoft Visual C++ Redistributable Package for Visual Studio 2015 (x64)

On Windows, download and install the Microsoft Visual C++ Redistributable package for Visual Studio 2015 (x64) from the Microsoft website.

## Extract the Installer Files on UNIX

On UNIX, the installer files are distributed as a tar file.

Use a native tar or GNU tar utility to extract the installer files to a directory on your machine.

You can extract the installer files in the following ways:

- Installation DVD. Copy the Informatica tar file from the installation DVD to a directory on your machine and extract the installer files, or extract the installer files directly from the DVD to a directory on your machine.
- FTP download. Download the Informatica installation tar file from the Informatica Electronic Software Download (ESD) site to a directory on your machine and extract the installer files.

## Set Up the X Window Server on UNIX

On UNIX, when you run the installer in graphical mode, use a graphics display server. The graphics display server is typically an X Window server. If you do not have the X Window server installed on the machine where you want to install the product, you can use an X Window server installed on another machine or run the installer in the console mode.

# Installing MDM Registry Edition in Graphical Mode

You can install MDM Registry Edition in graphical mode on Windows and UNIX. On UNIX, the installer requires an X Window server. On Windows, log in to the computer as an administrator before you start the installation.

1. Generate the license information file and send it to Informatica Global Customer Support. Based on the license information file, a representative from Global Customer Support creates a license key file and sends it to you.
2. Install the components of MDM Registry Edition.

## Step 1. Generate the License Information File

Before you install MDM Registry Edition, you must generate the license information file. The license information file contains the properties of the computer on which you install the product.

Send the license information file to Informatica Global Customer Support. A Global Customer Support representative creates a license key file and sends it to you.

The license key file is specific to the computer on which you generated the license information file. You cannot use the same key file on a different computer.

**Note:** If you have a date-limited trial license key, you do not have to generate the license information file. You can use the date-limited trial license key on any computer.

1. Run the installer.
2. Select **Generate License Information File**, and click **Next**.
3. Enter the absolute path for the license information file, and click **Next**.

The default path is as follows:

- On Windows: <User Home Directory>\idslcns.txt

- On UNIX: <User Home Directory>/idslcnst.txt
4. Click **Next**.
  5. Click **Create**.
  6. Click **Finish** to go back to the main page of the installer, or click **Exit** to exit the installer.
  7. Send the `idslcnst.txt` file to Informatica Global Customer Support.  
A Global Customer Support representative creates a license key file and sends it to you.
  8. Copy the license key file to the root directory of the installer.

## Step 2. Install the MDM Registry Edition Components

You can install the components of MDM Registry Edition after you copy the license key file to the root directory of the installer.

1. Run the installer.
2. Select **Install Informatica IR Products**, and click **Next**.
3. Enter the absolute path for the installation directory, and click **Next**.
4. Select the components and the populations that you want to install, and click **Next**.  
The installer enables only the components and the populations for which you have a license.
5. On the pre-installation summary page, if you want to change any options, click **Back** to navigate to the previous pages. Otherwise, click **Next**.
6. Verify the host name for the following servers and the port on which the servers listen, and update them if needed:
  - SSA-NAME3 Server
  - Search Server
  - Connection Server
  - Rulebase Server
  - Console Server
  - Secure Search Server. Required if you plan to use Informatica Data Director. The server starts only if you configure the host name and the port number.
  - Security Server. Required if you plan to use Informatica Data Director. The server starts only if you configure the host name and the port number.
  - HTTP Search Server. Optional. The server starts only if you configure the host name and the port number.
  - XML Search Server. Optional. The server starts only if you configure the host name and the port number.
  - XML Synchronization Server. Optional. The server starts only if you configure the host name and the port number.
  - XML Console Server. Optional. The server starts only if you configure the host name and the port number.
7. If you select to install MDM Registry Edition server components, enter the following ODBC configuration details, and click **Next**.
  - Service name that MDM Registry Edition uses to refer to the database service.

- Data Source Name (DSN). If you do not specify any value, it defaults to the service name. All database drivers except Oracle require a valid DSN.
- Name of the ODBC driver that communicates with the database.
- Name of the database service that you configure in the database.

**Note:** You can change these values later by updating the `odbc.ini` file that you can find in the following directory:

- On Windows: `<MDM Registry Edition Installation Directory>\bin`
  - On UNIX: `<MDM Registry Edition Installation Directory>/bin`
8. If you select to install only the MDM Registry Edition client components, enter the absolute path of the server installation directory, and click **Next**.  
Ensure that you provide the same directory that you provided when you installed the server components.  
For example, if you have installed the server components in the `/applications/InformaticaIR` directory on UNIX, use the same directory even though you install the client components on Windows.
  9. Click **Install**.
  10. Click **Finish** to go to the main page of the installer, or click **Exit** to exit the installer.

## Installing MDM Registry Edition in Console Mode

You can install MDM Registry Edition in console mode on Windows and UNIX.

**Note:** On Windows, log in to the computer as an administrator before you start the installation.

1. Generate the license information file and send it to Informatica Global Customer Support.  
Based on the license information file, a Global Customer Support representative creates a license key file and sends it to you.
2. Optionally, if you plan to use a parameter file, create a parameter file and add the required parameters to the file.
3. Install the components of MDM Registry Edition.

### Step 1. Generate the License Information File

Before you install MDM Registry Edition, you must generate the licence information file. The license information file contains the properties of the computer on which you install the product..

1. Go to the following directory:
  - On Windows. `<Root Directory of Installer>\bin`
  - On UNIX. `<Root Directory of Installer>/bin`
2. Run the following command:

```
ssalinfo [-o <Absolute Path and File Name>]
```

`Absolute Path and File Name` indicates the absolute path and the file name for the license information file. If you do not specify `-o <Absolute Path and File Name.txt>`, the utility creates the license information file in the current directory. The default file name is `idslcons.txt`.

For example, the following command generates the license information file in the current directory:

```
ssalinfo -o licenseinfo.txt
```

3. Send the license information file to Informatica Global Customer Support.  
A Global Customer Support representative creates a license key file and sends it to you.
4. Copy the license key file to the root directory of the installer.

## Step 2. Configure a Parameter File

Optionally, create a parameter file to install the components of MDM Registry Edition in silent mode on Windows or UNIX. The installer reads the parameter file to determine the installation options.

Consider the following guidelines when you specify parameters in a parameter file:

- Use a space or a new line to separate the parameters.
  - Specify the parameters in any order.
  - If you use a parameter both in the parameter file and in the console, the latest occurrence of the parameter takes precedence. The installer reads the parameter file first and then the parameters specified in the console. The values of the parameters specified in the console override the values of the parameters specified in the file.
1. Create a parameter file and name it as `instparms.txt`.
  2. Add the required parameters to the `instparms.txt` file, and save it.

### RELATED TOPICS:

- [“List of Parameters” on page 15](#)

## Step 3. Install the MDM Registry Edition Components

After you add the license key file to the root directory of the installer, you can install all the components or only the client components of MDM Registry Edition.

To install all the component or the client components of MDM Registry Edition, run the following command:

```
install [<Parameter File>] <Parameter List>
```

Configure the following parameters:

#### Parameter File

Optional. Absolute path of the parameter file, `instparms.txt`. For example:

```
install c:\store\instparms.txt
```

#### Parameter List

List of parameters that you can add to the installer. You can provide multiple parameters, separated by spaces. For example:

```
install --install-type all --server-co localhost:1660 --install-dir C:\InformaticaIR
```

## List of Parameters

You can add the following parameters to the MDM Registry Edition installer:

#### -1<Log File Name>

Optional. Absolute path of the log file. If you specify this parameter, the installer captures all the log entries and the debugging messages in the specified file.

If you do not specify this parameter, the installer creates a log file in the temporary directory with the `iirin` prefix, a three digit number, and the `.log` extension. The installer sequentially increments the three digit number.

For example, if the first log file name is `iirin001.log`, the subsequent log file name is `iirin002.log`. The installer returns an error if the log files with numbers up to 999 already exist.

Consider the following guidelines when you specify this parameter:

- You cannot use this parameter within a parameter file.
- On Windows, ensure that you define either the `TMP` or `TEMP` environment variable. If you do not define the environment variable, the installer returns an error.
- On UNIX, the temporary directory can be `$HOME/tmp`, `/usr/tmp`, `/tmp`, or `/var/tmp`.

#### **-2<Error File Name>**

Optional. Absolute path of the error file. If you specify this parameter, the installer captures all the error messages in the specified error file.

If you do not specify this parameter, the installer creates an error file in the temporary directory with the `iirin` prefix, a three digit number, and the `.err` extension. The installer sequentially increments the three digit number.

For example, if the first error file name is `iirin001.err`, the subsequent error file name is `iirin002.err`. The installer returns an error if the error files with numbers up to 999 already exist.

Consider the following guidelines when you specify this parameter:

- You cannot use this parameter within a parameter file.
- On Windows, ensure that you define either the `TMP` or `TEMP` environment variable. If you do not define the environment variable, the installer returns an error.
- On UNIX, the temporary directory can be `$HOME/tmp`, `/usr/tmp`, `/tmp`, or `/var/tmp`.

#### **--debug-level <Debug Level>**

Optional. Level of debug information that you want to capture in the log file. The debug level value must be an integer that is greater than 0.

The default value is 0. If the debug level value is 0, the installer does not log any debug information.

#### **--install-dir <Installation Directory>**

Required. Absolute path for the installation directory.

#### **--install-dir-remote <Installation Directory of Remote Server>**

Required if you install only the client components. Absolute path of the installation directory if you have installed the server components on a separate computer.

You must provide the same directory that you provided when you installed the server components.

For example, if you install the server components in the `/user/InformaticaIR` directory on UNIX, use the same directory even though you install the client components on Windows.

#### **--install-type all | cli | lic | upg | force-upgrade | uni**

Required. Type of installation. Enter one of the following values:

- `all`. Installs both the server and the client components.
- `cli`. Installs the client components.



- upg. Performs an upgrade of an existing installation.
- force-upgrade. Performs a forced upgrade of an existing installation.
- uni. Uninstalls an existing installation.

**--jre <JRE Executable>**

Required if you do not add the absolute path of the `bin` directory of the JDK or JRE installation to the `PATH` environment variable. Absolute path of the Java executable. You can optionally set the `JAVA_OPTS` environment variable to pass additional options to the JRE.

The Java executable must not contain spaces, and you cannot use this parameter within a parameter file.

**--odbc-dbtype <Database Type>**

Required if you configure the ODBC settings. Type of database. Enter one of the following values:

- msq. Indicates Microsoft SQL Server.
- ora. Indicates Oracle.
- udb. Indicates IBM DB2 UDB.

**--odbc-driver <ODBC Driver Name>**

Required if you configure the ODBC settings. Name of the ODBC driver that communicates with the database.

**--odbc-dsn <ODBC Data Source Name>**

Optional. Data Source Name (DSN). If you do not specify any value, it defaults to the service name. All database drivers except Oracle Database require a valid DSN.

**--odbc-native-service <Native Database Service>**

Required if you configure the ODBC settings. Name of the database service that you configure in the database.

**--odbc-service-name <Service Name>**

Required if you configure the ODBC settings. Service name that MDM Registry Edition uses to refer to the database service.

**--progress-window**

Optional. Displays a window with the progress messages during the installation or uninstallation. On UNIX, you can see the messages if you have a graphics display server.

**--server-co <Host Name>:<Port Number>**

Host name or IP address of the Connection Server and the port number on which the server listens.

**--server-cs <Host Name>:<Port Number>**

Host name or IP address of the Console Server and the port number on which the server listens.

**--server-cx <Host Name>:<Port Number>**

Optional. Host name or IP address of the XML Console Server and the port number on which the server listens.

**--server-ht <Host Name>:<Port Number>**

Optional. Host name or IP address of the HTTP Search Server and the port number on which the server listens.

The server starts only if you specify this parameter.

**--server-n3 <Host Name>:<Port Number>**

Optional. Host name or IP address of the SSA-NAME3 Server and the port number on which the server listens.

**--server-rb <Host Name>:<Port Number>**

Host name or IP address of the Rulebase Server and the port number on which the server listens.

**--server-se <Host Name>:<Port Number>**

Host name or IP address of the Search Server and the port number on which the server listens.

**--server-xm <Host Name>:<Port Number>**

Optional. Host name or IP address of the XML Search Server and the port number on which the server listens.

The server starts only if you specify this parameter.

**--server-xs <Host Name>:<Port Number>**

Optional. Host name or IP address of the XML Synchronization Server and the port number on which the server listens.

The server starts only if you specify this parameter.

## RELATED TOPICS:

- [“Step 2. Configure a Parameter File” on page 15](#)

# CHAPTER 3

## Post-Installation Tasks

This chapter includes the following topics:

- [Post-Installation Tasks Overview, 19](#)
- [Database Objects, 20](#)
- [Configuring Oracle, 20](#)
- [Configuring Microsoft SQL Server, 24](#)
- [Configuring IBM DB2 UDB, 27](#)
- [Configuring IBM DB2 on z/OS, 32](#)
- [Testing the Database Configuration, 32](#)
- [Testing the MDM Registry Edition Installation, 33](#)
- [Increasing the Number of Database Connections, 35](#)
- [Configuring Security, 35](#)
- [Starting the MDM Registry Edition Components, 37](#)
- [Stopping the MDM Registry Edition Server, 38](#)

## Post-Installation Tasks Overview

After you install the components of MDM Registry Edition, perform the following post-installation tasks:

1. Configure the database and ODBC data source.
2. Test the database configuration.
3. Test the MDM Registry Edition Server installation.
4. Optionally, increase the number of database connections.
5. Configure security.
6. Install Informatica Data Director components. For information about installing and configuring Informatica Data Director, see the *Informatica MDM Registry Edition Data Director Installation Guide*.
7. Optionally, install the client components on different computers.
8. Configure security.
9. Start the MDM Registry Edition components.

# Database Objects

MDM Registry Edition categorizes the database objects into the following logical groups:

## Source Database

A source database contains the user data in the User Source Tables (UST). MDM Registry Edition can access the USTs by using a user ID that has select privilege on the USTs.

MDM Registry Edition creates fuzzy indexes on the USTs. Use triggers to synchronize the indexes with the UST updates. The triggers store information about the UST rows that you updated, deleted, or added in a transaction table. The transaction table is part of the source database schema. The user ID must have the privilege to create triggers on the USTs.

## Target Database

A target database contains Identity Tables (IDT) and Identity Indexes (IDX). MDM Registry Edition creates and updates the identity tables and identity indexes by using a user ID that has create privileges.

If you want the IDTs and IDXs to remain in synchronization with the USTs, MDM Registry Edition also creates a two-phase-commit table, `IDS_2PC`, in the target database schema.

# Configuring Oracle

After you install the components of MDM Registry Edition, you must configure Oracle so that MDM Registry Edition can communicate with it.

**Note:** Informatica recommends that you must be an experienced DBA with appropriate authority to configure Oracle.

To configure Oracle for MDM Registry Edition, perform the following tasks:

1. Configure the optimizer mode.
2. Create users, tables, and database objects by running the database scripts.
3. Configure database access on the Search Server.
4. Optionally, configure the Oracle instance parameters to improve the database performance.

## Step 1. Configure the Optimizer Mode

The Update Synchronizer and some of the Oracle features that MDM Registry Edition uses work only with the cost-based optimizer (CBO). If you have optimized the existing applications to use the rule-based optimizer (RBO), you can enable CBO only for MDM Registry Edition.

To enable CBO for MDM Registry Edition, set the `OPTIMIZER_MODE` initialization parameter to `RULE` and collect statistics for the USTs. The existing applications continue to use RBO, and when MDM Registry Edition accesses the USTs, MDM Registry Edition specifies the `CHOOSE` hint. The `CHOOSE` hint causes the optimizer to choose between RBO and CBO based on the presence of statistics. If statistics exist for any of the USTs, MDM Registry Edition uses the CBO.

## Step 2. Create Users and Tables

Informatica provides SQL scripts that you can run to create users and tables. You can find the SQL scripts in the following directory:

- On Windows: <MDM Registry Edition Installation Directory>\dbscript\ora
- On UNIX: <MDM Registry Edition Installation Directory>/dbscript/ora

Before you run the SQL scripts, customize the scripts to update the user ID, password, and default table space.

**Note:** Copy the scripts to another directory before you customize the scripts. When you perform an upgrade, the installer can overwrite the customized scripts.

For a complete list of the database objects that the scripts create, see the database scripts.

1. On each source database, run the `idsuseru.sql` script.

The script creates a user with the required privileges.

The following table describes the privileges that the script grants to the user when you run the script:

Database Type	Privilege
Source database without synchronization	CREATE SESSION
Source database with synchronization	CREATE SESSION CREATE TABLE UNLIMITED TABLESPACE CREATE SEQUENCE CREATE PROCEDURE CREATE ANY TRIGGER

2. Grant the SELECT privilege on all the user source tables to the newly created user.
3. On each target database, run the `idsuseri.sql` script.

The `idsuseri.sql` script creates a user with the required privileges.

The following table describes the privileges that the script grants to the user when you run the script:

Database Type	Privilege
Target database without synchronization	CREATE SESSION CREATE TABLE UNLIMITED TABLESPACE
Target database with synchronization	CREATE SESSION CREATE TABLE UNLIMITED TABLESPACE CREATE SEQUENCE CREATE PROCEDURE

4. On each source database that requires synchronization and on each target database, run the `idstbora.sql` script.

The `idstbora.sql` script creates the following objects on the database:

- `IDS_FDT_META`
- `IDS_FDT_RECID`
- `IDS_RB_GROUPS`

5. On each source database that requires synchronization, run the `updsyncu.sql` script.

The `updsyncu.sql` script creates the `IDS_UPD_SYNC_TXN` object on the source database.

**Note:** Never rerun the `updsyncu.sql` script on a source database that you have already synchronized. Rerunning the script might result in a loss of synchronization.

6. If you want to create public synonyms for the database objects so that the objects can be shared, perform the following tasks:

- a. Open the `updsyncg.sql` script, and ensure that you specify the same user ID that you use to run the `updsyncu.sql` script.
- b. Run the `updsyncg.sql` script by using a user ID that has the `CREATE PUBLIC SYNONYM` privilege.

7. On each target database that requires synchronization, run the `updsynci.sql` script.

The `updsynci.sql` script creates the following synchronizer objects on the target database:

- `IDS_2PC`
- `IDS_UPD_SYNC_NSA`
- `IDS_UPD_SYNC_PID`
- `IDS_UPD_SYNC_REJ`

8. On each target database, run the `idsconv9.sql` script.

The `idsconv9.sql` script creates the `IDS_CONV` utility on the target database.

9. Optionally, on each target database, run the `ssasep.sql`, `ssaseb.sql`, and `ssasec.sql` scripts that you can find in the following directory:

- On Windows: `<MDM Registry Edition Installation Directory>\bin`
- On UNIX: `<MDM Registry Edition Installation Directory>/bin`

These scripts create PL/SQL API components on each target database. For more information about the scripts, see *Informatica MDM Registry Edition Developer Guide*.

## Step 3. Define an ODBC Data Source

You must define an ODBC data source on the Search Server to connect to Oracle. The MDM Registry Edition components access the database through ODBC except the MDM Registry Edition Table Loader, which directly loads data into the database. You must create ODBC data sources for each database that you plan to access.

To define an ODBC data source, open the `odbc.ini` file that you can find in the `<MDM Registry Edition Installation Directory>\bin` directory, and add or update the following database entries:

```
[<Service Name>]
driver = <ODBC Driver>
server = <Native Database Service>
ServerName = <Oracle Database Instance>
[connect_attributes = <Keyword1=Value1;Keyword2=Value2;...>]
[connect_attr_only = <Text1>]
```

Configure the following parameters:

### Service Name

Service name that MDM Registry Edition uses to refer to the database service.

### ODBC Driver

Name of the ODBC driver that communicates with the database. The driver name does not include the directory path.

Use one of the following values:

- On Windows. `sqora32.dll`
- On UNIX. `libsqora.so`

### Native Database Service

Name of the database service that you configure in the database.

### Oracle Database Instance

Name of the database instance. The MDM Registry Edition installer uses the same value as that of the database service.

### Keywords

List of connection string keywords that MDM Registry Edition uses to generate a connection string that establishes a connection with the database server.

The format uses the following values:

- `Keywords`. Optional. Connection string keywords that your ODBC driver supports. If you enable the `connect_attr_only` parameter, the specified values override the database source definitions and creates the connection string based only on the specified values. If you do not include the `connect_attr_only` parameter, MDM Registry Edition uses the specified values and the database source definitions to create the connection string.
- `Text1`. Optional. Indicates whether to use only the connection string keywords or the connection string keywords along with other data source definitions to create the connection string. Enter any text, such as `Yes`, to use only the keywords specified in the `connection_attributes` parameter to create the connection string. If you want to use the specified keywords with other database source definitions, do not add the `connect_attr_only` parameter to the `odbc.ini` file.

The following sample configuration indicates to use only the specified connection string keywords to create the connection string:

```
[o12clx6]
ServerName = o12clx6.cbr
driver = sqora32.dll
server = o12clx6.cbr
connect_attributes = UID=James;DSN=Orcl;PWD=secret;
connect_attr_only = yes
```

**Note:** You can use the following environment variables instead of configuring the connection string keywords in the `odbc.ini` file:

- `SSA_USE_SQLDRIVERCONNECT_ATTRIBUTES`
- `SSA_USE_SQLDRIVERCONNECT_ATTRIBUTES_ONLY`

For more information about these environment variables, see the *Informatica MDM Registry Edition Operations Guide*.

You can find a sample configuration file, `odbc.ini.ori`, in the following directory:

- On Windows. `<MDM Registry Edition Installation Directory>\bin`

- On UNIX. <MDM Registry Edition Installation Directory>/bin

**Note:** If you use multiple databases, add the database entries for each database. For example:

```
[oraclsrc]
driver = ssaoci9.dll
server = oraclsrc

[oracltgt]
driver = ssaoci9.dll
server = oracltgt
```

## Step 4. Configure the Oracle Instance Parameters

Optionally, you can configure the following Oracle instance parameters to improve the performance of the database:

- OPTIMIZER\_DYNAMIC\_SAMPLING
  - MAX\_COMMIT\_PROPAGATION\_DELAY
1. Configure `OPTIMIZER_DYNAMIC_SAMPLING=0` to prevent the optimizer from collecting dynamic statistics, which improves the query execution time.
  2. Configure `MAX_COMMIT_PROPAGATION_DELAY=0` to avoid any delay in data synchronization.

# Configuring Microsoft SQL Server

After you install the components of MDM Registry Edition, you must configure Microsoft SQL Server so that MDM Registry Edition can communicate with it.

**Note:** Informatica recommends that you must be an experienced DBA with appropriate authority to configure Microsoft SQL Server.

To configure Microsoft SQL Server for MDM Registry Edition, perform the following tasks:

1. Create a user and grant the required permissions.
2. Define an ODBC data source on the Search Server.
3. Create tables and database objects by running the database scripts.
4. Optionally, create Microsoft SQL API components.

## Step 1. Create a User

Create a user that you can use to create and access the MDM Registry Edition tables and database objects. The user must not be a database owner (dbo). A database owner is a member of the `sysadmin` fixed server role. If a member of the `sysadmin` fixed server role creates an object, the object belongs to dbo, not to the member. In this case, MDM Registry Edition cannot identify the database objects that the user creates. You must add the user to the `db_ddladmin` fixed database role to create triggers on the USTs.



The following table describes the fixed database roles to which you must add the user that you create:

Database Type	Fixed Database Role
Source database without synchronization	db_datareader
Target database without synchronization	db_datareader db_datawriter db_ddladmin
Source database with synchronization	db_datareader db_datawriter db_ddladmin
Target database with synchronization	db_datareader db_datawriter db_ddladmin

## Step 2. Define an ODBC Data Source

You must define an ODBC data source on the Search Server to connect to Microsoft SQL Server. The MDM Registry Edition components access the database through ODBC except the MDM Registry Edition Table Loader, which directly loads data into the database. You must create ODBC data sources for each database that you plan to access.

1. Open the `odbc.ini` file that you can find in the `<MDM Registry Edition Installation Directory>\bin` directory, and add or update the following database entries:

```
[<Service Name>]
DataSourceName = <ODBC DSN>
driver = <ODBC Driver>
server = <Native Database Service>
```

Configure the following parameters:

### Service Name

Service name that MDM Registry Edition uses to refer to the database service.

### ODBC DSN

Data Source Name. If you do not specify any value, it defaults to the service name.

### ODBC Driver

Name of the ODBC driver that communicates with the database. The driver name does not include the directory path. Use `sqlncli11` as the ODBC driver name. You can also use the Microsoft SQL Server Native Client as the ODBC driver.

### Native Database Service

Name of the Microsoft SQL Server. For example, `localhost`.

A sample configuration file, `odbc.ini.ori`, is located in the `<MDM Registry Edition Installation Directory>\bin` directory.

**Note:** If you use multiple databases, add the database entries for each database. For example:

```
[srcDBcli10]
driver = sqlncli11
server = (local)
```

```

DataSourceName = srcDBcli10

[tgtDBcli11]
driver = sqlncli11
server = (local)
DataSourceName = tgtDBcli11

```

2. Open the ODBC Data Source Administrator by performing the following tasks:
  - a. From the Windows Control Panel, click **Administrative Tools**.
  - b. In the **Administrative Tools** dialog box, double-click **Data Sources (ODBC)**.
3. On the **User DSN** tab or **Service DSN** tab, click **Add**.
4. Select **SQL Server** or **SQL Server Native Client**, and click **Finish**.
5. Follow the instructions in the Wizard, and ensure that you enter the following values:
  - Specify a data source name. Ensure that the data source name matches the data source name that you specify in the `odbc.ini` file.
  - Select the Microsoft SQL Server authentication and provide the user credentials that you created.
  - Specify a default database.
  - Ensure that you do not select the **Use regional settings when outputting currency, numbers, dates and times** check box.

**Note:** If you want to use the Windows authentication to connect to Microsoft SQL Server, set the `SSA_DB_WINDOWS_AUTHENTICATION` environment variable to Yes.

## Step 3. Create Tables

Informatica provides scripts that you can run to create tables in Microsoft SQL Server. You can find the scripts in the following directory: `<MDM Registry Edition Installation Directory>\dbscript\msq`

1. On each source database that requires synchronization and on each target database, run the following command to run the `idstbmsq.sql` script:

```
sqlcmd -i idstbmsq.sql -U <User Name> -P <Password> -S <Server Name>
```

This command uses the following parameters:

**User Name**

Name of the user that can create tables and database objects.

**Password**

Password for the user.

**Server Name**

Name of the Microsoft SQL Server. For example, localhost.

You can ignore the following warning:

```
Warning: The table 'IDS\FDT\META' has been created but its maximum row size (9171)
exceeds the maximum number of bytes per row (8060). INSERT or UPDATE of a row in
this table will fail if the resulting row length exceeds 8060 bytes.
```

The `idstbmsq.sql` script creates the following objects on the database:

- `IDS_FDT_META`
- `IDS_FDT_RECID`
- `IDS_RB_GROUPS`

2. On each source database that requires synchronization, run the `msqsyncu.sql` script.

The `msqsyncu.sql` script creates the `IDS_UPD_SYNC_TXN` object on the source database.

**Note:** Never rerun the `msqsyncu.sql` script on a source database that you have already synchronized. Rerunning the script might result in a loss of synchronization.

3. Open the `msqsyncp.sql` script and change the database name in the INSERT statement to the database name that contains the tables created by the `msqsyncu.sql` script.
4. On each source database that requires synchronization, run the `msqsyncp.sql` script.  
This script creates the `IDS_UPD_SYNC` object on the source database.
5. On each target database that requires synchronization, run the `msqsynci.sql` script.

The `msqsynci.sql` script creates the following synchronizer objects on the target database:

- `IDS_UPD_SYNC_NSA`
- `IDS_2PC`

## Step 4. Create Microsoft SQL Server API Components

Optionally, you can create Microsoft SQL Server API components on each target database. For more information about Microsoft SQL Server API components, see the *Informatica MDM Registry Edition Developer Guide*.

1. Copy the `<MDM Registry Edition Installation Directory>\mssql\ssasexp.dll` file to the `<Microsoft SQL Server Installation Directory>\Binn` directory.
2. Copy the following files from the `<MDM Registry Edition Installation Directory>\bin` directory to the `<Microsoft SQL Server Installation Directory>\Binn` directory:
  - `ssaik.dll`
  - `ssasec.dll`
3. Run the following command to run the `<MDM Registry Edition Installation Directory>\mssql\ssasexp.sql` script:

```
sqlcmd -U <User Name> -P <Password> -S <Server Name> -i ssasexp.sql
```

Configure the following parameters:

**User Name**

Name of the user that can create tables and database objects.

**Password**

Password for the user.

**Server Name**

Name of the Microsoft SQL Server. For example, localhost.

## Configuring IBM DB2 UDB

After you install the components of MDM Registry Edition, you must configure IBM DB2 UDB so that MDM Registry Edition can communicate with it.

**Note:** Informatica recommends that you must be an experienced DBA with appropriate authority to configure IBM DB2 UDB.

To configure IBM DB2 UDB for MDM Registry Edition, perform the following tasks:

1. Set the database configuration parameters.
2. Create a user and grant the required authorities.
3. Create tables by running the database scripts.
4. Define an ODBC Data Source on the Search Server.
5. Install user-defined functions on all the source and target databases.
6. Install IBM DB2 Driver for ODBC and CLI.
7. Optionally, set the environment variables to improve load performance.

## Step 1. Set the Database Configuration Parameters

You must set the following database configuration parameters:

- APPLHEAPSZ. Set it to a minimum value of 1024.
- MAXAPPLS. Set it to a minimum value of 256.

## Step 2. Create a User

Create a user that you can use to create and access the MDM Registry Edition tables and database objects. IBM DB2 UDB uses the external authentication mechanism, so you can use the operating system to create the user.

By default, MDM Registry Edition uses the load utility of IBM DB2 UDB to move large quantities of data into newly created tables. You must have the SYSADM or DBADM authority to run the load utility.

The following table describes the authorities that you must grant the user:

Database Type	Authority
Source database without synchronization	CONNECT
Target database without synchronization	CONNECT CREATETAB LOAD
Source database with synchronization	CONNECT CREATETAB LOAD
Target database with synchronization	CONNECT CREATETAB LOAD ALTER

## Step 3. Create Tables

Informatica provides scripts that you can run to create tables in IBM DB2 UDB. You can find the scripts in the following directory:

- On Windows: <MDM Registry Edition Installation Directory>\dbscript\udb

- On UNIX: <MDM Registry Edition Installation Directory>/dbscript/udb

For a complete list of the database objects that the scripts create, see the database scripts.

1. On each database, run the following command to run the `idstbuddb.sql` script:

```
db2 -v -t -f idstbuddb.sql -z <Log File Name> +o
```

The `idstbuddb.sql` script creates the following objects on the database:

- IDS\_FDT\_META
- IDS\_FDT\_RECID
- IDS\_RB\_GROUPS

2. On each source database that requires synchronization, run the `udbsyncu.sql` script.

The `udbsyncu.sql` script creates the `IDS_UPD_SYNC_TXN` object on the source database.

**Note:** Never rerun the `udbsyncu.sql` script on a source database that you have already synchronized. Rerunning the script might result in a loss of synchronization.

3. On each target database that requires synchronization, run the `udbsynci.sql` script.

The `udbsynci.sql` script creates the following synchronizer objects on the target database:

- IDS\_UPD\_SYNC\_NSA
- IDS\_2PC

## Step 4. Define an ODBC Data Source

Define an ODBC data source on the Search Server to connect to IBM DB2 UDB. The MDM Registry Edition components access the database through ODBC. You must create ODBC data sources for each database that you plan to access.

1. Open the `odbc.ini` file that you can find in the <MDM Registry Edition Installation Directory>\bin directory on Windows or <MDM Registry Edition Installation Directory>/bin on UNIX, and add or update the following database entries:

```
[<Service Name>]
DataSourceName = <ODBC DSN>
driver = <ODBC Driver>
server = <Native Database Service>
[connect_attributes = <Keyword1=Value1;Keyword2=Value2;...>]
[connect_string_udb = CONNECT TO <Database Name>]
[connect_attr_only = <Text1>]
```

Configure the following parameters:

### Service Name

Service name that MDM Registry Edition uses to refer to the database service.

### ODBC DSN

Data Source Name. If you do not specify any value, it defaults to the service name.

### ODBC Driver

Name of the ODBC driver. The driver name does not include the directory path. Enter one of the following values:

- `db2cli` on Windows
- `db2` on UNIX
- `libdb2.a(shr_64.o)` on 64-bit AIX environments

## Native Database Service

Name of the database alias.

### Keywords

List of connection string keywords that MDM Registry Edition uses to generate a connection string that establishes a connection with IBM DB2 UDB.

The format uses the following values:

- **Keywords.** Optional. Connection string keywords that your ODBC driver supports. If you enable the `connect_attr_only` parameter, the specified values override the database source definitions and creates the connection string based only on the specified values. If you do not include the `connect_attr_only` parameter, MDM Registry Edition uses the specified values and the database source definitions to create the connection string.
- **Database Name.** Optional. The database name which authenticates the connection to IBM DB2 UDB.
- **Text1.** Optional. Indicates whether to use only the connection string keywords or the connection string keywords along with other data source definitions to create the connection string. Enter any text, such as `Yes`, to use only the keywords specified in the `connection_attributes` parameter to create the connection string. If you want to use the specified keywords with other database source definitions, do not add the `connect_attr_only` parameter to the `odbc.ini` file.

The following sample configuration indicates to use only the specified connection string keywords to create the connection string:

```
[KRBTEST]
driver = /opt/ibm/db2/V11.1/lib64/libdb2o.so
server = KRBTEST
connect_attributes = DSN=DBKRB;
connect_string_udb = CONNECT TO DBKRB;
connect_attr_only = Yes
```

For the preceding sample configuration, you get the following connection string which enables connection to IBM DB2 UDB using Kerberos authentication:

```
DSN=DBKRB
```

**Note:** You need not specify User ID or Password in the `connect_attributes` parameter as MDM Registry Edition uses the predefined Kerberos authorization.

A sample configuration file, `odbc.ini.ori`, is located in the following directory:

- On Windows: `<MDM Registry Edition Installation Directory>\bin`
- On UNIX: `<MDM Registry Edition Installation Directory>/bin`

**Note:** If you use multiple databases, add the database entries for each database. For example:

```
[srcDBcli]
driver = db2cli
server = (local)
DataSourceName = srcDBcli

[tgtDBcli]
driver = db2cli
server = (local)
DataSourceName = tgtDBcli
```

2. Open the ODBC Data Source Administrator by performing the following tasks:
  - a. From the Windows Control Panel, click **Administrative Tools**.
  - b. In the **Administrative Tools** dialog box, double-click **Data Sources (ODBC)**.
3. On the **User DNS** tab or **Service DNS** tab, click **Add**.
4. Select **IBM DB2 ODBC DRIVER**, and click **Finish**.

5. Follow the instructions in the Wizard, and ensure that you enter the following values:
  - Specify a data source name. Ensure that the data source name matches the data source name that you specify in the `odbc.ini` file.
  - Specify a default database.

## Step 5. Install the User-Defined Functions

You must install the user-defined functions on all the source and target databases. The user-defined functions extend the function of IBM DB2 UDB.

To install the user-defined functions, run the following command:

- **On Windows:** `copy <MDM Registry Edition Installation Directory>\lib\ssaudf16.dll <IBM DB2 UDB Installation Directory>\sqllib\function`
- **On UNIX:** `cp <MDM Registry Edition Installation Directory>/lib/ssaudf16 <IBM DB2 UDB Installation Directory>/sqllib/function`

## Step 6. Install IBM DB2 Driver for ODBC and CLI

You must install IBM DB2 Driver for ODBC and CLI to provide run-time support for the DB2 CLI API and the ODBC API. For more information about installing IBM DB2 Driver for ODBC and CLI, see the database documentation.

You must update the IBM DB2 ODBC and CLI initialization file, `db2cli.ini`, with the database source name and the database alias name.

1. Open the `db2cli.ini` file that you can find in the following directory:
  - **On Windows:** `<Database Instance Home Directory>\sqllib`
  - **On UNIX:** `<Database Instance Home Directory>/sqllib`
2. Add the following entries to the `db2cli.ini` file:

```
[<data-source>]
DBALIAS=<alias>
```

The `db2cli.ini` file uses the following parameters:

**data-source**

Name of the data source.

**alias**

Name of the database alias.

## Step 7. Set the Environment Variables

Optionally, you can set the following environment variables to improve the load performance:

- `SSAUSBPARALLELLOAD=<Integer>`
- `SSAUSBSEQUENCES=Yes`

IBM DB2 UDB prevents parallel loading when you load a table. To allow parallel loading, configure the `SSAUSBPARALLELLOAD` environment variable to a higher value that indicates the number of allowed parallel loads. A value of 999999 indicates unlimited parallel loads.

By default, IBM DB2 UDB does not use sequences. Set the `SSAUSBSEQUENCES` environment variable to Yes to use sequences.

# Configuring IBM DB2 on z/OS

You can use IBM DB2 on z/OS as a source database for MDM Registry Edition.

To configure IBM DB2 on z/OS, perform the following tasks:

1. Request the `ssa.xmit.ids910.ssaudf` data set file from Informatica Global Customer Support.

**Note:** This file is not part of the installation files, so you must request it from the Global Customer Support.

2. By using a FTP client in binary mode, transfer the data set file to your mainframe system.

For example:

```
ftp> quote site lrecl=80 recfm=fb blksize=3120 vol=ssawrk cylinders
ftp> bin
ftp> put ssa.xmit.ids910.ssaudf 'ssa.xmit.ids910.ssaudf'
```

**Note:** If you use the z/OS FTP client, use the `LOCSite` command instead of the `QUOte` command.

3. Unpack the `ssa.xmit.ids910.ssaudf` file by using one of the following methods:

- Run the following command from the shell:

```
receive indsn('ssa.xmit.ids910.ssaudf') da('ssaudf.load')
```

- Run the following command from JCL by invoking the `IKJEFT1A` TSO interpreter:

```
//SSAINS JOB (1),IEBCOPY,NOTIFY=&SYSUID
//STEP1 EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.IDS910.SSAUDF') DA('SSAUDF.LOADD') VOLUME(SSA001)
/*
/*
```

4. Copy the `SSAUDF` and `II4TOUL` data set members to the `DB2HLQ.RUNLIB.LOAD` load library.

## Testing the Database Configuration

After you configure the database, verify the database configuration to ensure that MDM Registry Edition communicates with the database.

You can use the `dblist` utility to verify your ODBC configuration. You can find the `dblist` utility in the following directory:

- On Windows: `<MDM Registry Edition Installation Directory>\bin`
- On UNIX: `<MDM Registry Edition Installation Directory>/bin`

The `dblist` utility uses the following syntax:

```
dblist -c -dodb:99:<User Name>/<Password>@<Service Name> [-i]
```

The `dblist` utility uses the following parameters:

**User Name**

Name of the user that you created.

**Password**

Password for the user.



## Service Name

Service name that MDM Registry Edition uses to refer to the database service.

**-i**

Optional. Includes debug information in the response.

The following text shows a sample response of the dblist utility:

```
dblist -c -dodb:99:ssa/SSA@ora920 -i
Maximum connections per module: 1024
Linked databases: sdb: odb:
Driver Manager: 'Informatica ODBC Driver Manager 860275'
ODBC Driver: 'ssaoci9 SSADB8 2.7.0.00GCCLIBC2Aug 20 2013 16:05:00'
DBMS Name: 'Oracle DBMS (11.2.0.1.0)'
Native DB type: 'ora'
Schema: 'DG6BYAGA'
dbtype = ora
driver = libssaoci9.so
server = o11r21x6.cbr

===== ODBC Driver = > testing =====
SQLROWSETSIZE = 8 bytes
SQLLEN = 8 bytes
SQL_C_WCHAR = 2 bytes
*** Connection successful ***
```

# Testing the MDM Registry Edition Installation

After you configure the database and verify the database configuration, you can test the MDM Registry Edition installation by loading a predefined system and verifying its search and synchronization results. If the test is successful, MDM Registry Edition is ready for the user-defined systems.

You can test the MDM Registry Edition installation either in the graphical mode or in the console mode.

## Test the MDM Registry Edition Installation in Graphical Mode

You can test the MDM Registry Edition installation in graphical mode on Windows and UNIX.

1. Start the MDM Registry Edition Server (Configure Mode) by performing one of the following tasks:

- On Windows: On a command line, run the following command from the <MDM Registry Edition Installation Directory>\bin directory:  
`idsup.bat -itests-mdmre.dat -t<MDM Registry Edition Installation Directory>\ids\tests`
- On Windows: From the Windows Start menu, click **Programs > Informatica > Identity Resolution [Version] > Informatica MDM - Registry Edition > Informatica MDM - Registry Server - start (Configure Mode)**.
- On UNIX: On a shell command line, run the following command from the <MDM Registry Edition Installation Directory>/bin directory:  
`idsup -itests-mdmre.dat -t<MDM Registry Edition Installation Directory>/ids/tests`

2. Start the MDM Registry Edition Console Client (Admin Mode) by performing any of the following tasks:

- On Windows and UNIX: Run the following command:  
`idsconc -a`
- On Windows: From the Windows Start menu, click **Programs > Informatica > Identity Resolution [Version] > Informatica MDM - Registry Edition > Informatica MDM - Registry Console Client (Admin Mode)**.

3. Enter the user name that you created in the database, password, and the database service name.
4. Select either all the tests or the specific tests to run, and click **OK**.

## Test the MDM Registry Edition Installation in Console Mode

If your environment does not support graphical user interface, you can test the MDM Registry Edition installation in console mode.

1. Create an input file with `idsbatch.in` as its name.
2. Add the following entries to the file, and save the file.

```
work-directory=#SSAWORKDIR#
#a
ction=rulebase-create
rulebase-name=odb:0:<username>/<password>@<dbserver>
#a
ction=database-create
database-name=odb:1:<username>/<password>@<dbserver>
#a
ction=system-import-without-load
system-name=ssa000
input-file=#SSATOP#/ids/systems/ssa000.sys
rulebase-name=odb:0:<username>/<password>@<dbserver>
database-name=odb:1:<username>/<password>@<dbserver>
work-directory=#SSAWORKDIR#
#a
ction=job-run
job-name=Job01
system-name=ssa000
rulebase-name=odb:0:<username>/<password>@<dbserver>
work-directory=#SSAWORKDIR#
```

3. From the <MDM Registry Edition Installation Directory>/bin directory, run the following command:

```
idsup -itests.dat -t<MDM Registry Edition Installation Directory>/ids/tests
```

4. Run the `idsbatch` utility by using the following command:

```
idsbatch -h#SSA_CSHOST# -i<Absolute Path>/idsbatch.in -l<Absolute Path>/idsbatch.log
-2<Absolute Path>/idsbatch.err
```

If the test is successful, the `idsbatch.err` file must be blank.

## Test the `ssapid` Utility on UNIX

On UNIX, the `ssapid` utility extracts process identification number of a process. Some MDM Registry Edition processes use the `ssapid` utility to extract the process identification numbers.

Ensure that the `ssapid` utility works before starting the MDM Registry Edition Server.

To verify that the `ssapid` utility works, run the `ssapid $$` command from the <MDM Registry Edition Installation Directory>/bin directory.

The output of the command returns all the process identification numbers with system-related information. For example:

```
735024109 -bash
```

If the command does not return any process identification numbers, the utility failed to run. The utility fails if your computer does not have a `ps` utility that supports the `-p` option. You must contact Informatica Global Customer Support to change the `ssapid` script to suit the `ps` utility that is available in your computer.

# Increasing the Number of Database Connections

MDM Registry Edition uses a modular architecture to manage the number of database connections. The `libssadbm.so` database module manages the database connections, and it can manage a fixed number of connections, which is 1024. You can find the database module in the following directory:

- On Windows: `<MDM Registry Edition Installation Directory>\bin`
- On UNIX: `<MDM Registry Edition Installation Directory>/bin`

To know how many connections the database module supports, run the following command:

```
dblist -c
```

The following text shows a sample response:

```
Maximum connections per module: 1024
Linked databases: odb: sdb:
```

If you want more connections, create copies of the `libssadbm.so` file, and use the following syntax to name the database modules:

`libssadbm<nnn>.so`, where `nnn` can be an integer from 001 through 999.

For each copy of the database module, you can have an additional 1024 connections.

## Configuring Security

You can configure the security framework of MDM Registry Edition to authenticate the Informatica Data Director client. The security framework exposes a simple API interface that implements the security feature with minimal efforts.

For more information about the security framework, see the *Informatica MDM Registry Edition Security Framework Guide*.

## Configuring MDM Registry Edition Security Server

Configure the Security Server that exposes the security framework authentication APIs over SSL to authenticate the Informatica Data Director client. MDM Registry Edition uses a secure channel to communicate with the Security Server to run the security framework.

By default, the Security Server uses LDAP security provider. To use the database security provider, update the `<MDM Registry Edition Installation Directory>\bin\secup` file on Windows or `<MDM Registry Edition Installation Directory>/bin/secup` file on UNIX with the `-tDB` option as follows:

```
start /b %SSAJAVA% -Djavax.net.ssl.keyStore=%SSA_SEC_KEYSTORE% ^
-Djavax.net.ssl.keyStorePassword=%SSA_SEC_KEYPASSWORD% ^
-Djava.util.logging.config.file="d:\\logging.properties" ^
-classpath "%SSA_SEC_CLASSPATH%" com.informatica.rbss.server.impl.SecurityServer ^
-w%SSASERVERWORKDIR% -h%SSA_SSHOST% %SSA_SEC_ARGS% -tDB
```

Before you start MDM Registry Edition, verify the following environment variables for the Security Server and update them if required:

### **SSANOSECAUTHORIZATION**

Indicates whether you want to disable the authorization process. If you use a database or LDAP as the security provider, you must disable the authorization process. If you disable the authorization process,

the security framework does not verify whether a user has sufficient privileges to access the requested resource.

To disable the authorization process, set the `SSANOSECAUTHORIZATION` environment variable to Yes. By default, the authorization process is enabled.

### **SSASERVERLOGSDIR**

Directory that stores the server logs. The installer sets this environment variable, and it creates the `sserver.trc` debug file in this directory.

### **SSA\_SEC\_LOGGING\_PROPERTIES**

Configure this environment variable only if you require custom logging. The following text shows a sample custom logging.properties file:

```
handlers = java.util.logging.ConsoleHandler, java.util.logging.FileHandler
# Set the default logging level for the root logger
.level = OFF
# Set the default logging level for new ConsoleHandler instances
java.util.logging.ConsoleHandler.level = OFF
# Set the default logging level for new FileHandler instances
java.util.logging.FileHandler.level = INFO
# Set the default formatter for new ConsoleHandler instances
java.util.logging.ConsoleHandler.formatter =
java.util.logging.SimpleFormatter
# Set the default logging level for the logger
# use level FINE to enable debug level messages
com.informatica.rbs.level = INFO
com.informatica.rbss.level = INFO
```

### **SSA\_SEC\_LIB**

Directory that contains dependent libraries for the Security Server. The default directory is as follows:

- On Windows: <MDM Registry Edition Installation Directory>\security\lib
- On UNIX: <MDM Registry Edition Installation Directory>/security/lib

### **SSA\_SEC\_KEYSTORE**

The `server.keystore` file that contains the key for the server to run as a SSL server. The default value for this variable is as follows:

- On Windows: <MDM Registry Edition Installation Directory>\security\cert\server.keystore
- On UNIX: <MDM Registry Edition Installation Directory>/security/cert/server.keystore

### **SSA\_SEC\_CONFIG**

Framework configuration file for the Security Server. The default value for this variable is as follows:

- On Windows: <MDM Registry Edition Installation Directory>\security\SecConfig.xml
- On UNIX: <MDM Registry Edition Installation Directory>/security/SecConfig.xml

### **SSA\_SEC\_DICT**

Framework `.dict` file for the Security Server. If you set this variable, MDM Registry Edition does not consider the `SSA_SEC_CONFIG` variable.

# Starting the MDM Registry Edition Components

After you complete all the post-installation tasks, you can start the MDM Registry Edition components.

To start the MDM Registry Edition components, perform the following tasks:

1. Run the `mdmres` script to configure the environment variables for the MDM Registry Edition Server.
2. Start the MDM Registry Edition Server.
3. Start the MDM Registry Edition Console Client.

## Configure the Environment Variables

Before starting the MDM Registry Edition Server, you must configure environment variables for the MDM Registry Edition Server.

To configure the environment variables, run the `mdmres` command from the following directory:

- On Windows: `<MDM Registry Edition Installation Directory>\env directory`
- On UNIX: `<MDM Registry Edition Installation Directory>/env directory`

**Note:** On POSIX-compliant shells, the script might unexpectedly terminate if the script refers to any undefined environment variable. If the script terminates unexpectedly, add the `set +u` command to the second line of the `mdmres` script, and run the script again.

## Start MDM Registry Edition

After you configure the environment variables, you can start the MDM Registry Edition Server.

1. Start the MDM Registry Edition servers by performing one of the following tasks:
  - On Windows: On a command line, run the `idsup.bat` file from the `<MDM Registry Edition Installation Directory>\bin directory`.
  - On Windows: From the Windows Start menu, click **Programs > Informatica > Identity Resolution [Version] > Informatica MDM - Registry Edition > Informatica MDM - Registry Server - start**.
  - On UNIX: On a shell command line, run the `idsup` command from the `<MDM Registry Edition Installation Directory>/bin directory`.
2. Start the MDM Registry Edition Console Client (Admin Mode) by performing any of the following tasks:
  - On Windows and UNIX: Run the `idsconc -a` command.
  - On Windows: From the Windows Start menu, click **Programs > Informatica > Identity Resolution [Version] > Informatica MDM - Registry Edition > Informatica MDM - Registry Console Client (Admin Mode)**.
3. Enter the user name that you created in the database, password for the user, and the database service name.

A successful authentication of the user credentials launches the MDM Registry Edition Console Client.

# Stopping the MDM Registry Edition Server

To stop the MDM Registry Edition Server, perform one of the following tasks:

- On Windows: On a command line, run the `idsdown.bat` file from the `<MDM Registry Edition Installation Directory>\bin` directory.
- On Windows: From the Windows Start menu, click **Programs > Informatica > Identity Resolution [Version] > Informatica MDM - Registry Edition > Informatica MDM - Registry Server - stop**.
- On UNIX: On a shell command line, run the `idsdown` command from the `<MDM Registry Edition Installation Directory>/bin` directory.

# CHAPTER 4

## Troubleshooting

This chapter includes the following topics:

- [Troubleshooting MDM Registry Edition Installation, 39](#)
- [Troubleshooting IBM DB2 UDB, 39](#)

### Troubleshooting MDM Registry Edition Installation

**When I installed MDM Registry Edition on Windows, the installer failed to run the specified program.**

If you did not install the Microsoft Visual C++ 2008 SP1 Redistributable Package (x86) library, you get the following message:

```
The system cannot execute the specified program.
```

Download and install the Microsoft Visual C++ 2008 SP1 Redistributable Package (x86) library from the Microsoft web site.

**When I installed MDM Registry Edition, the installer could not find the Java installation.**

If the PATH environment variable does not point to the `bin` directory of the JDK or JRE installation, you get the following message:

```
Java not found in the current PATH.
```

If you receive this message, add the `bin` directory of the JDK or JRE installation to the PATH environment variable.

**When I installed MDM Registry Edition in an existing empty directory, the installation failed.**

Delete the directory, and run the installer again.

### Troubleshooting IBM DB2 UDB

The first time when you use IBM DB2 UDB after you install MDM Registry Edition, you might encounter the following issues:

## When I ran the Update Synchronizer, I received an SQL07323N error message.

If you do not install the user-defined functions, the Update Synchronizer displays the following message:

```
SQL07323N An error occurred in a triggered SQL statement in trigger "SSA.IDS0000U"  
Information returned from the error includes SQLCODE "-444",  
SQLSTATE "42724" and message tokens  
"*_VARCHAR|SQL050704134911400|...qlib/function". SQLSTATE=09000
```

Run the following command to get the reason code, and install the user-defined functions on all the source and target databases.

```
db2 values ids_comp_varchar  
(SMALLINT(3),'fred','tony','henry',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ');:
```

## When I ran a database process, I received an SQL1224N error message.

If you do not start the database manager or the database agent, the following error might appear:

```
SQL1224N A database agent could not be started to service a request, or was terminated  
as a result of a database system shutdown or a force command.
```

Perform the following tasks:

1. Ensure that the database manager runs on the database server.
2. Ensure that the database agent runs.
3. Ensure that the number of agents allocated to the database manager does not exceed the maximum limit.
4. Connect your local machine as a remote client by performing the following tasks:
  - a. Run the following command to catalog the node:

```
db2 catalog tcpip node <Node Name> remote <Host Name> server <Server Name>
```
  - b. Run the following command to verify that the catalog command worked correctly:

```
db2 list node directory
```
  - c. Run the following command to catalog the database:

```
db2 catalog db <Database Name> as <Database Alias>  
db2 uncatalog db <Database Name>  
db2 catalog db <Database Alias> as <Database Name> at node <Node Name>
```

**Note:** This command prevents the SQL1334N error that can occur when the database name is same as the database alias.
  - d. Stop and start IBM DB2 UDB.

## On Windows, from the Start menu, when I selected Informatica MDM - Registry Server - start, the idsup.bat file launched a Windows command prompt instead of the IBM DB2 UDB command prompt.

If you configure IBM DB2 UDB as your database after installing MDM Registry Edition, the **Informatica MDM - Registry Server - start** menu might launch the Windows command prompt, `cmd.exe`.

You get the Windows prompt if the **Informatica MDM - Registry Server - start** shortcut refers to the Windows command prompt. For example, `C:\SQLLIB\bin\cmd.exe /C C:\InformaticaIR\bin\idsup.bat`.

To get the IBM DB2 UDB command prompt, manually update the shortcut to launch the IBM DB2 UDB command prompt, `db2cmd.exe`. For example, `C:\SQLLIB\bin\db2cmd.exe /C C:\InformaticaIR\bin\idsup.bat`.



### On AIX, when I ran the `dblist` command, I received an error message.

If the ODBC Driver Manager is linked with an archived library instead of a shared library, the `dblist` command might fail with the following message:

```
The module has an invalid magic number.
```

To resolve this issue, link the ODBC Driver Manager with a shared library.

To link the ODBC Driver Manager with a shared library, perform the following tasks:

1. Run the following commands to extract the shared library:

```
cd <IBM DB2 UDB Installation Directory>/sqllib/lib
ar -x -X 64 libdb2.a
mv shr_64.o db2_64.o
```

2. Open the configuration file of the ODBC Driver Manager, which is usually `odbcinst.ini`, and update the value of the ODBC driver with the shared library, `db2_64.o`.

## CHAPTER 5

# Upgrading MDM Registry Edition

This chapter includes the following topics:

- [Upgrade Overview, 42](#)
- [Upgrading the MDM Registry Edition Components in Console Mode, 42](#)
- [Disabling Authorization, 46](#)

## Upgrade Overview

After you upgrade MDM Registry Edition, you can use the latest version of the MDM Registry Edition Console to connect to the Rulebase and the database that you created by using the earlier version of MDM Registry Edition. This process upgrades the Rulebase objects to the latest version. You cannot use the earlier version of the MDM Registry Edition Console to connect to the upgraded Rulebase.

You must first upgrade the server components and then the client components. The upgraded server components work with the earlier versions of the client components, but the upgraded client components do not work with the earlier versions of the server components. Informatica recommends that you perform the upgrade process on all the computers that has the MDM Registry Edition components.

## Upgrading the MDM Registry Edition Components in Console Mode

Optionally, if you plan to use a parameter file, create a parameter file with `instparms.txt` as its name, and add the required parameters to the file.

Consider the following guidelines when you specify parameters in the parameter file:

- Use a space or a new line to separate the parameters.
- Specify the parameters in any order.
- If you use a parameter both in the parameter file and in the console, the latest occurrence of the parameter takes precedence. The installer reads the parameter file first and then the parameters specified in the console. The values of the parameters specified in the console override the values of the parameters specified in the file.

To upgrade the MDM Registry Edition components, perform the following tasks:

1. Extract the installer files.
2. Optionally, if you plan to use a parameter file, perform the following tasks:
  - a. Create a parameter file and name it as `instparms.txt`.
  - b. Add the required parameters to the file, and save the file.
3. From the root directory of the installer, run the following command:

```
install [<Parameter File>] --install-type upg|force-upgrade <Parameter List>
```

Configure the following parameters:

#### Parameter File

Optional. Absolute path of the parameter file, `instparms.txt`. For example:

```
install c:\store\instparms.txt
```

#### --install-type upg|force-upgrade

Type of installation. Enter one of the following values:

- `upg`. Performs an upgrade of an existing installation. If the installer identifies that the hotfix is applied, the installer ends the upgrade process.
- `force-upgrade`. Performs a forced upgrade of an existing installation. If the installer identifies that the hotfix is applied, the installer reapplies the hotfix.

#### Parameter List

List of parameters that you can add to the installer. If you do not specify any parameter, the installer considers the existing value.

## List of Parameters

You can add the following parameters to the MDM Registry Edition installer:

#### -1<Log File Name>

Optional. Absolute path of the log file. If you specify this parameter, the installer captures all the log entries and the debugging messages in the specified file.

If you do not specify this parameter, the installer creates a log file in the temporary directory with the `iirin` prefix, a three digit number, and the `.log` extension. The installer sequentially increments the three digit number.

For example, if the first log file name is `iirin001.log`, the subsequent log file name is `iirin002.log`. The installer returns an error if the log files with numbers up to 999 already exist.

Consider the following guidelines when you specify this parameter:

- You cannot use this parameter within a parameter file.
- On Windows, ensure that you define either the `TMP` or `TEMP` environment variable. If you do not define the environment variable, the installer returns an error.
- On UNIX, the temporary directory can be `$HOME/tmp`, `/usr/tmp`, `/tmp`, or `/var/tmp`.

#### -2<Error File Name>

Optional. Absolute path of the error file. If you specify this parameter, the installer captures all the error messages in the specified error file.

If you do not specify this parameter, the installer creates an error file in the temporary directory with the `iirin` prefix, a three digit number, and the `.err` extension. The installer sequentially increments the three digit number.

For example, if the first error file name is `iirin001.err`, the subsequent error file name is `iirin002.err`. The installer returns an error if the error files with numbers up to 999 already exist.

Consider the following guidelines when you specify this parameter:

- You cannot use this parameter within a parameter file.
- On Windows, ensure that you define either the `TMP` or `TEMP` environment variable. If you do not define the environment variable, the installer returns an error.
- On UNIX, the temporary directory can be `$HOME/tmp`, `/usr/tmp`, `/tmp`, or `/var/tmp`.

#### **--debug-level <Debug Level>**

Optional. Level of debug information that you want to capture in the log file. The debug level value must be an integer that is greater than 0.

The default value is 0. If the debug level value is 0, the installer does not log any debug information.

#### **--install-dir <Installation Directory>**

Required. Absolute path for the installation directory.

#### **--install-dir-remote <Installation Directory of Remote Server>**

Required if you install only the client components. Absolute path of the installation directory if you have installed the server components on a separate computer.

You must provide the same directory that you provided when you installed the server components.

For example, if you install the server components in the `/user/InformaticaIR` directory on UNIX, use the same directory even though you install the client components on Windows.

#### **--install-type all | cli | lic | upg | force-upgrade | uni**

Required. Type of installation. Enter one of the following values:

- `all`. Installs both the server and the client components.
- `cli`. Installs the client components.
- `upg`. Performs an upgrade of an existing installation.
- `force-upgrade`. Performs a forced upgrade of an existing installation.
- `uni`. Uninstalls an existing installation.

#### **--jre <JRE Executable>**

Required if you do not add the absolute path of the `bin` directory of the JDK or JRE installation to the `PATH` environment variable. Absolute path of the Java executable. You can optionally set the `JAVA_OPTS` environment variable to pass additional options to the JRE.

The Java executable must not contain spaces, and you cannot use this parameter within a parameter file.

#### **--odbc-dbtype <Database Type>**

Required if you configure the ODBC settings. Type of database. Enter one of the following values:

- `msq`. Indicates Microsoft SQL Server.
- `ora`. Indicates Oracle.
- `udb`. Indicates IBM DB2 UDB.

**--odbc-driver <ODBC Driver Name>**

Required if you configure the ODBC settings. Name of the ODBC driver that communicates with the database.

**--odbc-dsn <ODBC Data Source Name>**

Optional. Data Source Name (DSN). If you do not specify any value, it defaults to the service name. All database drivers except Oracle Database require a valid DSN.

**--odbc-native-service <Native Database Service>**

Required if you configure the ODBC settings. Name of the database service that you configure in the database.

**--odbc-service-name <Service Name>**

Required if you configure the ODBC settings. Service name that MDM Registry Edition uses to refer to the database service.

**--progress-window**

Optional. Displays a window with the progress messages during the installation or uninstallation. On UNIX, you can see the messages if you have a graphics display server.

**--server-co <Host Name>:<Port Number>**

Host name or IP address of the Connection Server and the port number on which the server listens.

**--server-cs <Host Name>:<Port Number>**

Host name or IP address of the Console Server and the port number on which the server listens.

**--server-cx <Host Name>:<Port Number>**

Optional. Host name or IP address of the XML Console Server and the port number on which the server listens.

**--server-ht <Host Name>:<Port Number>**

Optional. Host name or IP address of the HTTP Search Server and the port number on which the server listens.

The server starts only if you specify this parameter.

**--server-n3 <Host Name>:<Port Number>**

Optional. Host name or IP address of the SSA-NAME3 Server and the port number on which the server listens.

**--server-rb <Host Name>:<Port Number>**

Host name or IP address of the Rulebase Server and the port number on which the server listens.

**--server-se <Host Name>:<Port Number>**

Host name or IP address of the Search Server and the port number on which the server listens.

**--server-xm <Host Name>:<Port Number>**

Optional. Host name or IP address of the XML Search Server and the port number on which the server listens.

The server starts only if you specify this parameter.

**--server-xs <Host Name>:<Port Number>**

Optional. Host name or IP address of the XML Synchronization Server and the port number on which the server listens.

The server starts only if you specify this parameter.

#### RELATED TOPICS:

- [“Step 2. Configure a Parameter File” on page 15](#)

## Disabling Authorization

If your environment does not use an authorization process, set the `SSANOSECAUTHORIZATION` environment variable to Yes, and restart the MDM Registry Edition Server. The `SSANOSECAUTHORIZATION` environment variable disables the authorization process. By default, the authorization process is enabled.

**Note:** If you want to set up an authorization process, contact Informatica Global Customer Support for assistance.

## CHAPTER 6

# Uninstalling MDM Registry Edition

This chapter includes the following topics:

- [Uninstallation Overview, 47](#)
- [Uninstalling MDM Registry Edition in Graphical Mode, 47](#)
- [Uninstalling MDM Registry Edition in Console Mode, 48](#)

## Uninstallation Overview

You can uninstall MDM Registry Edition either in the graphical mode or in the console mode. Uninstallation removes all the components of MDM Registry Edition. You cannot uninstall a specific component of MDM Registry Edition.

## Uninstalling MDM Registry Edition in Graphical Mode

You can uninstall MDM Registry Edition in graphical mode on Windows and UNIX.

On UNIX, the installer requires an X Window server. On Windows, log in to the computer as an administrator before you start the uninstallation.

1. Stop all the MDM Registry Edition clients and servers and ensure that the MDM Registry Edition Server processes are not running.
2. Perform one of the following tasks:
  - Run the `install --uninstall` command from the `<MDM Registry Edition Installation Directory>` directory that you want to uninstall.
  - Run the `install --uninstall=<MDM Registry Edition Installation Directory>` command. The command uninstalls the installation directory that you specify.
  - On Windows: From the Windows Start menu, click **Programs > Informatica > Identity Resolution [Version] > Uninstall Informatica IR Products**.
  - On UNIX: Run the `uninstall` script from the `<MDM Registry Edition Installation Directory>` directory.

A confirmation message appears.

3. Click **Yes** to uninstall MDM Registry Edition, or click **No** to exit.

# Uninstalling MDM Registry Edition in Console Mode

You can uninstall MDM Registry Edition in console mode on Windows and UNIX.

On Windows, log in to the computer as an administrator before you start the uninstallation.

1. Stop all the MDM Registry Edition clients and servers and ensure that the MDM Registry Edition Server processes do not run.
2. Run the following command from the <MDM Registry Edition Installation Directory> directory:  

```
install --install-type uni
```



# INDEX

[.NET Framework 11](#)

## A

[authorization 46](#)

## C

components

- [client 8](#)
- [populations 8](#)
- [server 8](#)

configuration

- [database 20](#)
- [IBM DB2 on z/OS 32](#)
- [IBM DB2 UDB 27](#)
- [MDM Registry Edition security server 35](#)
- [Microsoft SQL Server 24](#)
- [ODBC settings 22, 25, 29](#)
- [optimizer mode 20](#)
- [Oracle 20](#)
- [Oracle instance parameters 24](#)

console mode

- [testing 34](#)

create

- [IBM DB2 UDB 28](#)
- [IBM DB2 UDB tables 28](#)
- [Microsoft SQL Server API components 27](#)
- [Microsoft SQL Server tables 26](#)
- [Microsoft SQL users 24](#)
- [Oracle tables 21](#)
- [Oracle users 21](#)

## D

database

- [IBM DB2 UDB 27](#)
- [Microsoft SQL Server 24](#)
- [Oracle 20](#)
- [source 20](#)
- [target 20](#)
- [testing 32](#)

[database configuration 20](#)

[database connections 35](#)

## E

environment variables

- [IBM DB2 UDB 31](#)
- [MDM Registry Edition Server 37](#)

## G

graphical mode

- [installation requirements 12](#)
- [testing 33](#)

## I

[IBM DB2 on z/OS 32](#)

IBM DB2 UDB

- [creating tables 28](#)
- [creating users 28](#)
- [database configuration parameters 28](#)
- [Driver for ODBC and CLI 31](#)
- [environment variables 31](#)
- [ODBC configuration 29](#)
- [user-defined functions 31](#)

installation

- [console mode 14](#)
- [graphical mode 12](#)
- [MDM Registry Edition 13, 15](#)

installation requirements

- [X Window Server 12](#)

[installation tasks 9](#)

installer

- [parameters 15, 43](#)

## J

[Java Development Kit 11](#)

[Java Runtime Environment 11](#)

[JAVA\\_HOME 11](#)

[JAVA\\_OPTS 11](#)

[JDK 11](#)

[JRE 11](#)

## L

[libssaoci9.so 11](#)

license

- [information file 12](#)
- [key file 12](#)

## M

MDM Registry Edition

- [components 8](#)
- [installation 12, 14](#)
- [starting 37](#)
- [system requirements 10](#)
- [uninstallation 47, 48](#)

[MDM Registry Edition Security Server 35](#)

- Microsoft SQL Server
  - API components [27](#)
  - creating tables [26](#)
  - creating users [24](#)
  - ODBC configuration [25](#)
- Microsoft SQL Server API components [27](#)

## O

- ODBC
  - IBM DB2 UDB [29](#)
  - Microsoft SQL Server [25](#)
  - Oracle [22](#)
- ODBC drivers [11](#)
- optimizer mode [20](#)
- Oracle
  - creating users and tables [21](#)
  - instance parameters [24](#)
  - ODBC configuration [22](#)
  - optimizer mode [20](#)
- oracle ODBC drivers
  - libssaoci9.so [11](#)
  - ssaoci9 [11](#)

## P

- parameters [15](#), [43](#)
- PATH [11](#)
- post-installation tasks [19](#)
- pre-installation tasks
  - MDM Registry Edition [10](#)
- process identification number [34](#)

## S

- security framework
  - MDM Registry Edition Security Server [35](#)

- source database [20](#)
- ssaoci9 [11](#)
- starting
  - MDM Registry Edition Console Client [37](#)
  - MDM Registry Edition Server [37](#)
- stopping
  - MDM Registry Edition Server [38](#)
- system requirements [10](#)

## T

- target database [20](#)
- testing
  - MDM Registry Edition [33](#)
- troubleshooting
  - installation process [39](#)

## U

- uninstallation [47](#), [48](#)
- upgrade
  - MDM Registry Edition [42](#)
- upgrading
  - MDM Registry Edition [42](#)

## X

- X Window Server
  - installation requirements [12](#)