



Informatica® Development Platform

Informatica Connector Toolkit Getting Started Guide

© Copyright Informatica LLC 2014, 2022

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, the Informatica logo, Big Data Management, Informatica Cloud, and PowerCenter 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.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

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.

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: 2022-08-11

Table of Contents

Preface	5
Informatica Resources.	5
Informatica Network.	5
Informatica Knowledge Base.	5
Informatica Documentation.	6
Informatica Product Availability Matrices.	6
Informatica Velocity.	6
Informatica Marketplace.	6
Informatica Global Customer Support.	6
 Chapter 1: Getting Started Overview.....	 7
Informatica Connector Toolkit Overview.	7
Informatica Connector Perspective.	8
Connector Navigator.	8
Connector Progress.	8
Analyze the Data Source.	9
Data Connector Example: MySQL.	9
Building a Connector.	9
 Chapter 2: Installing and Upgrading the Informatica Connector Toolkit.....	 11
Installing the Informatica Connector Toolkit Overview.	11
Before You Begin.	12
Download the Welcome Kit.	12
Install Required Software.	12
Configure Environment Variables for OpenJDK on Linux.	13
Configure Environment Variables for OpenJDK on Windows.	13
Download the Cloud Data Integration Services Connector Toolkit.	14
Installation of Eclipse IDE.	14
Installing Informatica Connector Toolkit on Windows.	14
Installed Informatica Connector Toolkit Components.	15
Sample Connector Source Codes.	15
 Chapter 3: Creating a Connector.....	 16
Creating a Connector Overview.	16
Step 1. Select the Product Type and Define the Connector Properties.	18
Step 2. Define the Connection Attributes.	20
Test and Debug the Connection to the Data Source.	25
Step 3. Define the Connector Type System.	26
Step 4. Define Connector Metadata.	29
Test Metadata from the Data Source.	36

Step 5. Define the Run-Time Interfaces.	36
Test the Read and Write Capabilities of the Connector.	37
Test the Read Capability of the Connector.	38
Test the Write Capability of the Connector.	39
Informatica Connector Toolkit Eclipse Plug-in Tip.	41
Chapter 4: Exporting a Connector.	42
Exporting a Connector Overview.	42
Export the Connector.	43
Appendix A: Frequently Asked Questions.	44
Informatica Connector Toolkit Frequently Asked Questions.	44
Index.	46

Preface

The Cloud Data Integration Informatica® Connector Toolkit Getting Started Guide provides information about the Informatica connector toolkit and how to use the toolkit to develop connectors for the Informatica Intelligent Cloud Services™ platform. The Informatica connector toolkit provides a MySQL connector sample source code that you can use to develop a connector.

This guide assumes you have a working knowledge of Informatica services and Informatica Cloud and are familiar with data sources, ODBC, JDBC, and application programming interfaces.

Informatica Resources

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

Informatica Network

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

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

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

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrices

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

Informatica Velocity

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

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

Informatica Marketplace

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

Informatica Global Customer Support

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

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

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

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

CHAPTER 1

Getting Started Overview

This chapter includes the following topics:

- [Informatica Connector Toolkit Overview, 7](#)
- [Informatica Connector Perspective, 8](#)
- [Analyze the Data Source, 9](#)
- [Data Connector Example: MySQL, 9](#)
- [Building a Connector, 9](#)

Informatica Connector Toolkit Overview

Use the Informatica Connector Toolkit to build a connector that provides connectivity between a data source and the Cloud Data Integration.

Although Informatica supports generic ODBC connectivity and allows access to any data source that has a standards-compliant ODBC driver, building a connector by using the Informatica Connector Toolkit offers several advantages. In case where no ODBC driver is available, building a connector by using the Informatica Connector Toolkit might be the only solution.

When you use the Informatica Connector Toolkit to create a connector, you can build functionality related to the data source. You can preserve datatype integrity and metadata lineage of the data source when you retain the type system of the data source and perform optimal data type conversions.

The Informatica Connector Toolkit consists of libraries, plug-ins, and sample code to assist you in developing connectors for Cloud Data Integration. You can use the Informatica Connector perspective in the Eclipse IDE to quickly develop a connector in an Eclipse environment.

The Informatica Connector Toolkit simplifies the following processes:

- **Development.** You can use the wizards in the Informatica Connector perspective to rapidly develop a connector. The wizards simplify the use of internal components and dependencies when you develop a connector.
- **Testing.** After you define the connector components, you can test the connection, metadata, and run-time components of the connector.
- **Deployment.** You can deploy the connector on Cloud Data Integration Service.

The Informatica Connector Toolkit API is written in a combination of Java. The connection definition and metadata definition are available in Java. The run-time interfaces are available in Java.

Note: Run-time interfaces are available only in Java when you develop connectors for Informatica Cloud.

Informatica Connector Perspective

The views in the Informatica Connector perspective enable you to develop connectors for Informatica Cloud.

After you click the **Create New Connector** icon in the Eclipse Workbench toolbar, you can choose to switch to the Informatica Connector perspective. You can also open the Informatica Connector perspective from the **Window** menu in the Eclipse IDE.

The Informatica Connector perspective consists of the Connector Navigator view and Connector Progress view in the Eclipse Workbench window, and icons in the Eclipse toolbar. You can use the Connector Navigator and Connector Progress views to define, edit, or test the connector components and view the progress of the connector project. Use the **Create New Connector** icon in the Toolbar to create a connector project. You can use the **View or Create Messages** icon in the Eclipse toolbar to view or create messages for the connector components.

Connector Navigator

You can use the Connector Navigator view to add, define, edit, or test connector components, such as connection, metadata, and run-time.

After you create an Informatica Connector project, you can right-click folders in the Connector Navigator view to define, edit, or test connector components. You can also right-click the connector source files and edit them in the Eclipse Workbench editor or any other editor.

Connector Progress

You can use the Connector Progress view to view the completeness of the connector project and to define or edit the connector project components in each phase.

The Connector Progress view consists of the following phases:

Connectivity

You can define, edit, or test the connection component of a connector in the Connectivity phase.

Metadata

You can define and edit data types and native metadata object components of a connector in the Metadata phase. You can also define common metadata for Informatica Cloud connectors in the Metadata phase. You can test the connector metadata components in the Metadata phase.

Runtime

You can define, edit, and test the connector run-time components in the Run-time phase.

Export/Publish

You can publish the connector to a location or deploy the connector on local Informatica services and client directories in the Export/Publish phase.

Analyze the Data Source

Before you develop a connector for a data source, perform a detailed analysis of the data source.

Analyze the following aspects of the data source:

- Connection attributes that you need to connect to the data source.
- Informatica platform data objects that you need to create to read data and write data.
- Native data types and the possible mapping to the Informatica platform data types.
- APIs available for the following tasks:
 - Validate connection attributes and connect to the data source
 - Browse metadata from the data source
 - Read from or write to the data source
- Additional configuration attributes that you require to read or write data to the data source.

Data Connector Example: MySQL

You can use the MySQL connector sample source code in the Informatica Connector Toolkit to develop a connector for MySQL.

Use the MySQL connector sample source code to read data from or write data to MySQL. To use the MySQL connector sample source code, import the sample projects into Eclipse. Edit the connection, metadata, and run-time components if required and then publish the connector bundle.

Note: The MySQL connector sample source code in the Informatica Connector Toolkit is for illustration purposes only.

Building a Connector

Use the Informatica Connector Toolkit to develop and deploy a connector for the Informatica platform. You can use the Informatica Connector perspective in the Eclipse IDE to rapidly develop a connector.

To build a connector with the Informatica Connector Toolkit, perform the following tasks:

1. Install the Informatica Connector Toolkit and set up the Informatica Connector Toolkit Eclipse plug-in.
2. Create an Informatica connector project and define the following components:
 - Connection
 - Type system
 - Metadata
 - Runtime

Before you deploy the connector, you can test and debug the connection, metadata, and run-time components.

3. Publish the connector bundle or deploy and register the connector.

4. If you publish the connector bundle, run the server batch file to deploy and register the connector with Informatica services. Run the client batch file to deploy the client components.

CHAPTER 2

Installing and Upgrading the Informatica Connector Toolkit

This chapter includes the following topics:

- [Installing the Informatica Connector Toolkit Overview, 11](#)
- [Before You Begin, 12](#)
- [Installation of Eclipse IDE, 14](#)
- [Installing Informatica Connector Toolkit on Windows, 14](#)

Installing the Informatica Connector Toolkit Overview

Install the Informatica Connector Toolkit and set up the Informatica Connector Toolkit Eclipse plug-in. To set up the Informatica Connector Toolkit Eclipse plug-in, you install the required software on the machine where you plan to develop the connector.

The Informatica Connector Toolkit is part of the Informatica Development Platform (IDP). You can use the Informatica Connector Toolkit installer to install the Informatica Connector Toolkit on the machine in which you plan to develop an connector.

You can get the Informatica Connector Toolkit installer from the following sources:

- Informatica electronic software download site.
When you purchase an Informatica product and choose to download the software, you receive a site link, user ID, and password to access the Informatica electronic software download site. Follow the instructions on the download site to download the Informatica Connector Toolkit installation file.
- Informatica Technology Network.
If you are a registered user of the Informatica Technology Network, you can download the Informatica Connector Toolkit installation file from the Informatica Development Platform page. When you download the file, the Informatica Development Network provides you with a password. Use this password when you extract the files from the download file.

When you run the Informatica Connector Toolkit installer, the installer installs the Informatica Connector Toolkit Eclipse plug-in in the Eclipse IDE.

Before You Begin

Before you develop a connector with the Informatica Connector Toolkit, you need to install the required software and analyze the data source.

Run the Informatica Connector Toolkit installer to install the Informatica Connector Toolkit on the machine where you plan to develop the connector. The Informatica Connector Toolkit contains the binaries, tools, samples, and documents that you require to build a connector.

Download the Welcome Kit

The Informatica Partner Program provides a complete set of enablement, marketing, and sales resources and information so that partners can develop and promote their services and solutions in conjunction with Informatica.

Click the following URL to download the Partner Welcome KIT:

[Download](#)

Install Required Software

Install the following software on the machine where you plan to develop the connector:

- Eclipse IDE for Java EE Developers with support for Plug-in Development Environment (PDE). Informatica Connector Toolkit supports the following Eclipse versions:

Eclipse	Version	Java
eclipse-juno	SR1	1.7 and above
eclipse-juno	SR2	1.7 and above
eclipse-kepler	SR1	1.7 and above
eclipse-kepler	SR2	1.7 and above
eclipse-luna	SR1	1.7 and above
eclipse-luna	SR2	1.7 and above
eclipse-mars	Milestone 1 (4.5.0M1)	1.7 and above
eclipse-mars	SR1-SR2	1.7 and above
eclipse-neon	M5	1.8 and above
eclipse-oxygen	3a	1.8 and above
eclipse-2020-06	4.16.0	1.8.x

Note: For AIX, install Eclipse 4.2.2 or above.

- Azul OpenJDK version 1.8.0_275 or previous sub versions.
- MySQL_Cloud Connector or J JDBC driver version 8.0.13 or later if you use the MySQL_Cloud sample connector.

- YouTube Data API client library for Java version 3 or later if you use the YouTube sample connector.

Configure Environment Variables for OpenJDK on Linux

After you download the Azul OpenJDK version 1.8.0_275 or any previous sub version on a Linux machine, you must configure the environment variables.

Perform the following steps to configure the environment variables:

1. After you download the Azul OpenJDK, extract the .tar file.
2. Set the variable name to `JAVA_HOME` and the variable value to `<Java installation directory>/jdk`:
`export JDK_HOME=<Java installation directory>/jdk`
3. Delete the `JRE_HOME` environment variable.
4. Edit the `PATH` variable and add `JAVA_HOME/bin` to the variable value.
`export PATH=$JDK_HOME/bin:$PATH`
5. In the Informatica Connector Eclipse IDE, open **Windows > Preferences**.
6. Navigate to **Java > Installed JREs**.
7. Click **Add**.
8. Select **Standard VM** as the **JRE Type**.
9. Click **Next**.
10. In the **JRE home** property, specify the `<Java installation directory>/jdk` or click **Directory** to navigate to the `<Java installation directory>/jdk`.
11. Enter a name for the **JRE name** property.
12. Click **Finish**.
13. Click **Apply** and then click **OK**.
14. Restart the Informatica Connector Eclipse IDE.

Configure Environment Variables for OpenJDK on Windows

After you download the Azul OpenJDK version 1.8.0_275 or any previous sub version on a Windows machine, you must configure the environment variables.

Perform the following steps to configure the environment variables:

1. After you download the Azul OpenJDK, extract the .zip file.
2. Open the **Advanced System Properties** from the Windows Control Panel.
3. Click **Environment Variables**.
4. Under System variables, click **New**.
5. Set the variable name to `JAVA_HOME` and the variable value to `<Java installation directory>/jdk`.
6. Edit the `Path` variable and add `JAVA_HOME/bin` to the variable value.
7. In the Informatica Connector Eclipse IDE, open **Windows > Preferences**.
8. Navigate to **Java > Installed JREs**.
9. Click **Add**.
10. Select **Standard VM** as the **JRE Type**.
11. Click **Next**.

12. In the **JRE home** property, specify the `<Java installation directory>` or click **Directory** to navigate to the `<Java installation directory>`.
13. Enter a name for the **JRE name** property.
14. Click **Finish**.
15. Click **Apply** and then click **OK**.
16. Restart the Informatica Connector Eclipse IDE.

Download the Cloud Data Integration Services Connector Toolkit

Download the Cloud Data Integration Connector Toolkit based on your system requirement:

Operating System	URL
Windows 64-bit	Download
Linux 64-bit	Download

Installation of Eclipse IDE

You can install Eclipse IDE on the machine in which you plan to develop the connector. Use the Eclipse IDE package to install Eclipse IDE.

You must install Eclipse version supported by Informatica Connector Toolkit for the Informatica Connector Toolkit plug-in to work with the Eclipse IDE.

You can download the Eclipse IDE package from the following location:

<http://www.eclipse.org/downloads/>

Installing Informatica Connector Toolkit on Windows

Install the Informatica Connector Toolkit to install the Informatica Connector Toolkit Eclipse plug-in and other components that you require to build a connector.

1. Close all other applications.
2. Run the `install.bat` file from the root directory.
3. In the **Welcome** page, click **Next**.
The **Installation Directory** page appears.
4. Select the installation directory in which you want to install the Informatica Connector Toolkit.
5. Select the installation directory in which you installed Eclipse IDE.
6. Select **Install Apache log4j** to install the log4j plug-in version that Informatica Connector Toolkit requires. The installer installs the required Apache log4j plug-in by default. If you have log4j plug-in installed and do not want to install a different version of the library, clear the **Install Apache log4j** option.

7. Select **Install Apache Commons library** to install the commons library that Informatica Connector Toolkit requires.
8. Click **Next**.
The **Pre-Installation Summary** page appears.
9. Click **Install**.
The **Installing** page appears and displays the installation progress.
10. Click **Done** to complete the installation procedure and then exit the installer.

Installed Informatica Connector Toolkit Components

After you install the Informatica SDKs, you can find the Informatica Connector Toolkit in the installation directory where you installed the Informatica Connector Toolkit.

The Informatica Connector Toolkit installation includes the following components to assist you in developing a connector for the Informatica platform:

- Informatica Connector Toolkit API files. Library files of the Informatica Connector Toolkit API.
- Informatica Connector Toolkit Eclipse plug-in. You can use the wizards and menus that the Informatica Connector Toolkit plug-in adds to the Eclipse IDE to develop, test, and deploy a connector.
- Sample MySQL and YouTube. The sample MySQL and YouTube include source code that you can use as a model to build a connector.

Note: The sample MySQL and YouTube connectors are for illustration purposes only.

- Informatica Connector Toolkit API Reference. Online documentation for the Informatica Connector Toolkit API specification.

Sample Connector Source Codes

The Informatica Connector Toolkit installation includes the following sample connector source codes:

- Instagram
- MySQL
- MySQL_Cloud
- Redis
- YouTube

CHAPTER 3

Creating a Connector

This chapter includes the following topics:

- [Creating a Connector Overview, 16](#)
- [Step 1. Select the Product Type and Define the Connector Properties, 18](#)
- [Step 2. Define the Connection Attributes, 20](#)
- [Step 3. Define the Connector Type System, 26](#)
- [Step 4. Define Connector Metadata, 29](#)
- [Step 5. Define the Run-Time Interfaces, 36](#)
- [Test the Read and Write Capabilities of the Connector, 37](#)
- [Informatica Connector Toolkit Eclipse Plug-in Tip, 41](#)

Creating a Connector Overview

In this lesson, you create a connector that enables the Data Integration Service to read data from and write data to MySQL.

Lesson Concepts

You can use the Informatica Connector Toolkit Eclipse plug-in to build a connector to provide connectivity between a data source and Cloud Data Integration.

Create an Informatica connector project for each connector that you want to build. For each project, you can use the Informatica Connector Toolkit Eclipse plug-in to generate the connection, metadata, type system, and run-time java source code.

After you create the project, you define the connection. When you define the connection, you configure the connection attributes, the connection behavior, and add libraries required by the connection to the project. Then, define the type system to configure the connector data types and how they map to the data types supported by the Informatica platform. Then, define the native metadata definition to configure the connector metadata required for the data source operation that you want to implement. Finally, define the run-time interfaces. When you define the run-time interfaces, you add run-time libraries to the project and configure how the connector performs each data source operation. You can also test and debug the connector connection, metadata, and run-time components.

Lesson Objectives

In this lesson, you complete the following tasks:

- Create a project for the MySQL connector.

- Define the connection attributes for the MySQL connector and test the connection components.
- Define the type system for the MySQL data types.
- Define a native metadata definition for the MySQL connector to read and write data.
- Define run-time interfaces to configure how the MySQL connector performs read and write operations.
- Test the read capability and write capability of the connector.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisite:

- Install the Informatica Connector Toolkit. For more information, see [“Installing the Informatica Connector Toolkit Overview” on page 11](#).

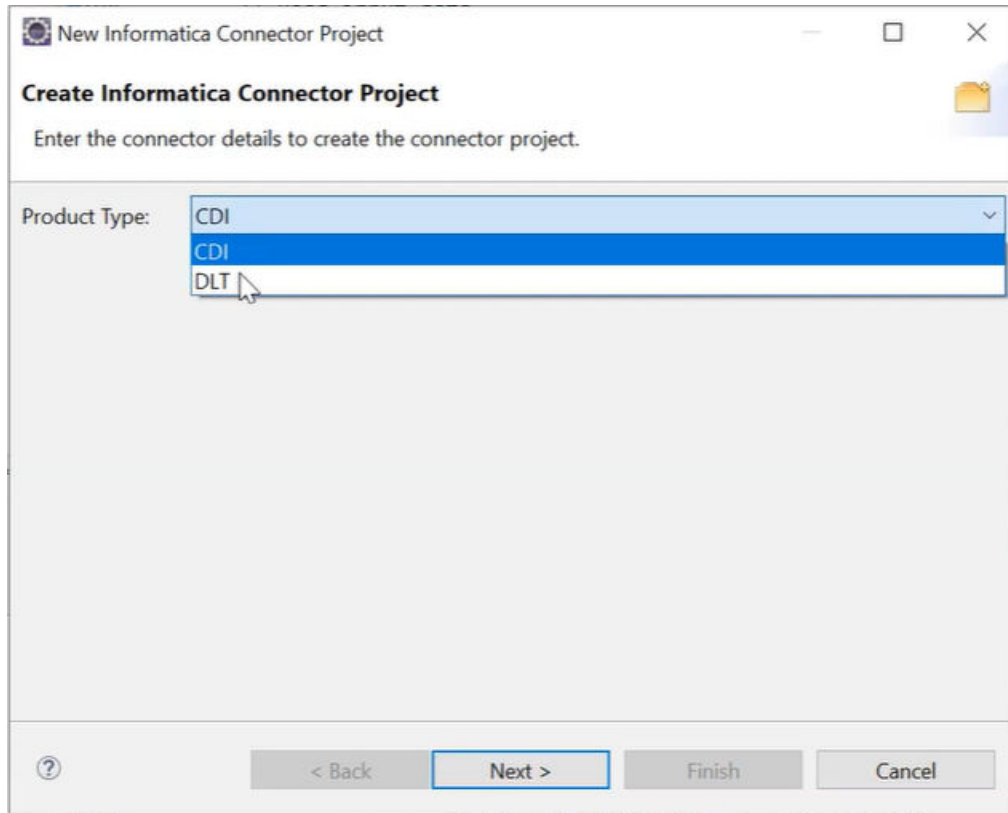
Lesson Timing

Set aside 30 minutes to complete this lesson.

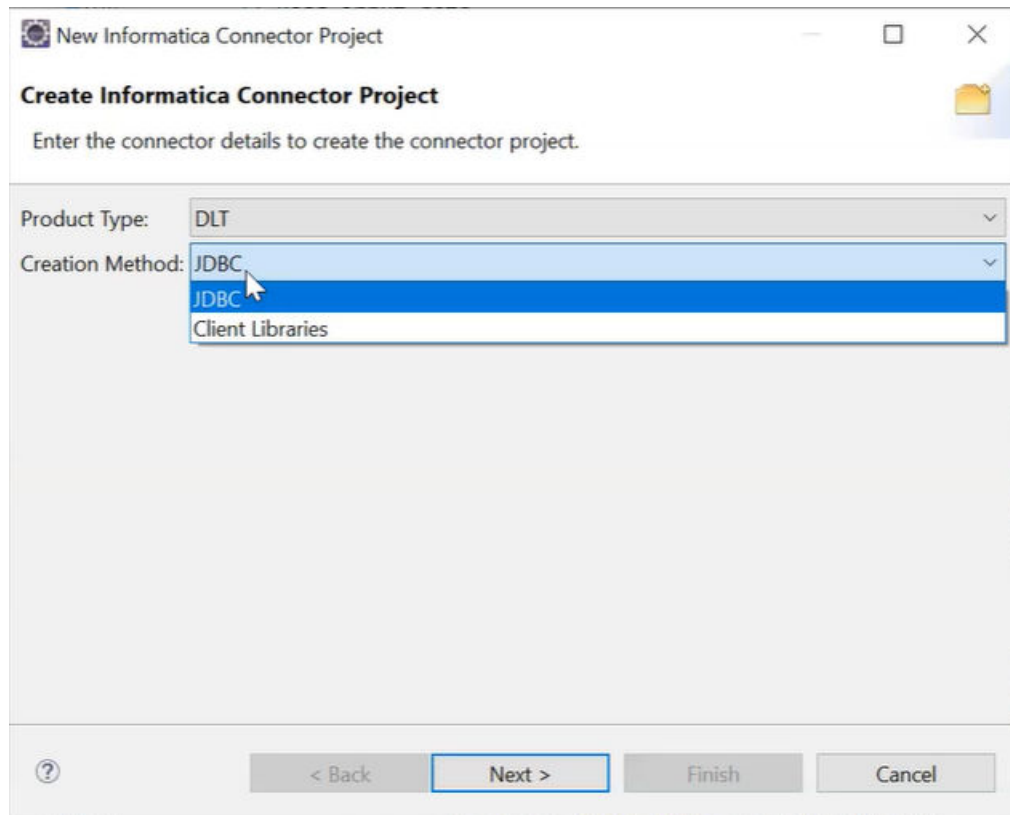
Step 1. Select the Product Type and Define the Connector Properties

Create a project for the MySQL connector and define the MySQL connector properties.

1. From the Eclipse IDE, click the **Create New Connector** button ().
The **Create Informatica Connector Project** dialog box appears.

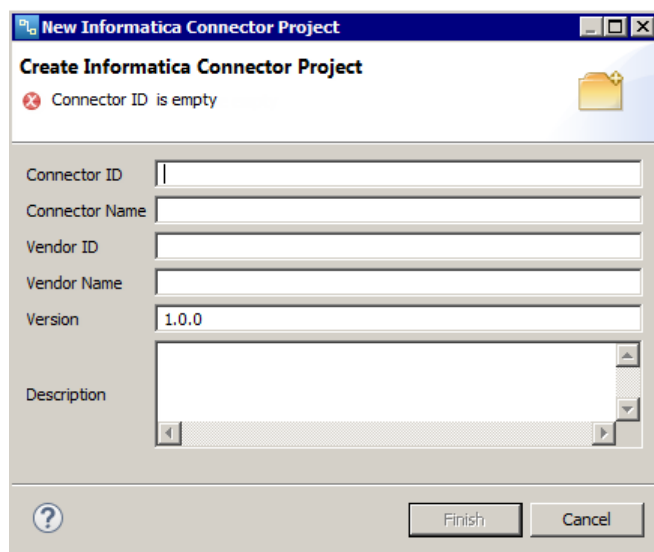


2. Select **CDI** or **DLT** as the product type to select Cloud Data Integration or Data Loader Task.
3. If you select **DLT**, select the creation method from the drop down.



4. Click **Next**.

The **Create Informatica Connector Project** dialog box appears.

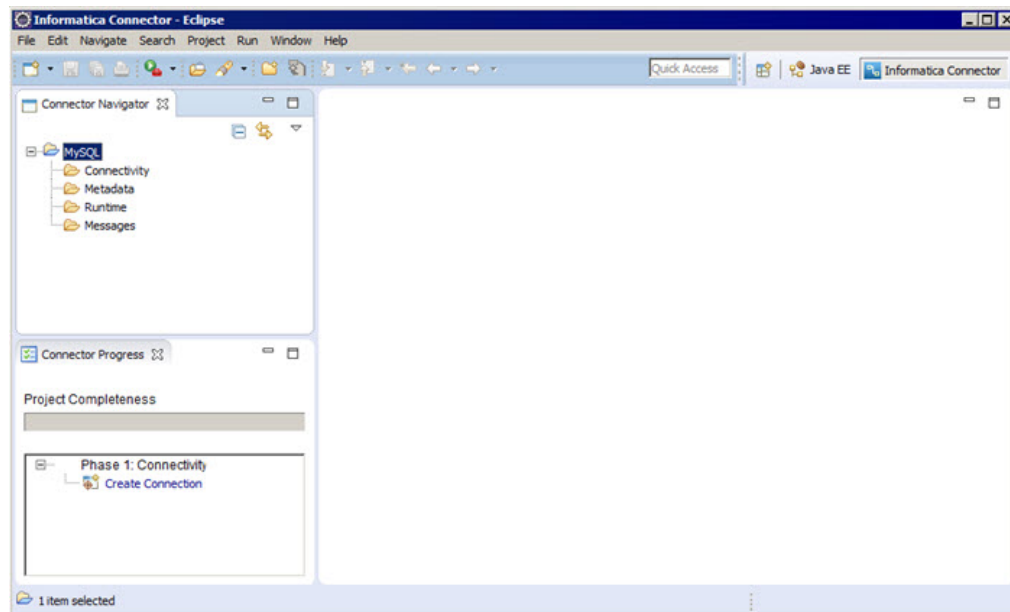


5. Enter the following MySQL connector project details:

Property	Value
Connector ID	mysql
Connector Name	MySQL
Vendor ID	Informatica
Vendor Name	Informatica Corporation
Version	1.0.0
Description	MySQL sample connector that demonstrates the Informatica Connector Toolkit capabilities.

6. Click **Finish**.

The MySQL connector project appears in the **Connector Navigator** view of the Eclipse IDE.



Step 2. Define the Connection Attributes

Define the connection attributes to specify the behavior and functionality of the connector when it connects to the MySQL database.

1. In the **Connector Navigator** view, right-click the project and select **Add Connection**.

The **Add Connection** dialog box appears.

2. Enter `MySQL` for the connection type name.
3. Select **Add New** and enter `Sample` for the category name.

The category name that you configure corresponds the connection object category.

4. Click **Next**.

The **Connection Attributes** page appears.

Add Connection

Connection Attributes

✖ Add attributes to proceed.

Name	Display Name	Data Type	Mandatory	Encrypted	Has Dependents

Add Edit Delete

? < Back Next > Save Generate Code Cancel

5. Click **Add** to enter each connection attribute.

The attributes that you configure correspond to the connection object properties.

The **Add Attribute** dialog box appears:

Add Attribute

Enter the attribute details

Name is empty

Name:

Display Name:

Description:

Data Type:

Default Value:

Min Length:

Max Length:

☐ Encrypted

☐ Mandatory

☐ Hidden

Allowed Values:

a. Enter the following values for the username attribute:

Property	Value
Name	username
Display Name	User Name
Description	The user name to connect to the MySQL database. [Required]
Datatype	String
Min Length	0
Max Length	16
Options	- Mandatory

- b. Enter the following values for the password attribute:

Property	Value
Name	password
Display Name	Password
Description	The password associated with your user name. [Required]
Datatype	String
Min Length	0
Max Length	16
Options	<ul style="list-style-type: none">- Encrypted- Mandatory

- c. Enter the following values for the host name attribute:

Property	Value
Name	hostname
Display Name	Host Name
Description	Host name of the MySQL server to connect. [Required]
Datatype	String
Default Value	localhost
Min Length	0
Max Length	16
Options	<ul style="list-style-type: none">- Mandatory

- d. Enter the following values for the port number attribute:

Property	Value
Name	port
Display Name	Port
Description	Port number of the MySQL server to connect. [Required]
Datatype	Integer

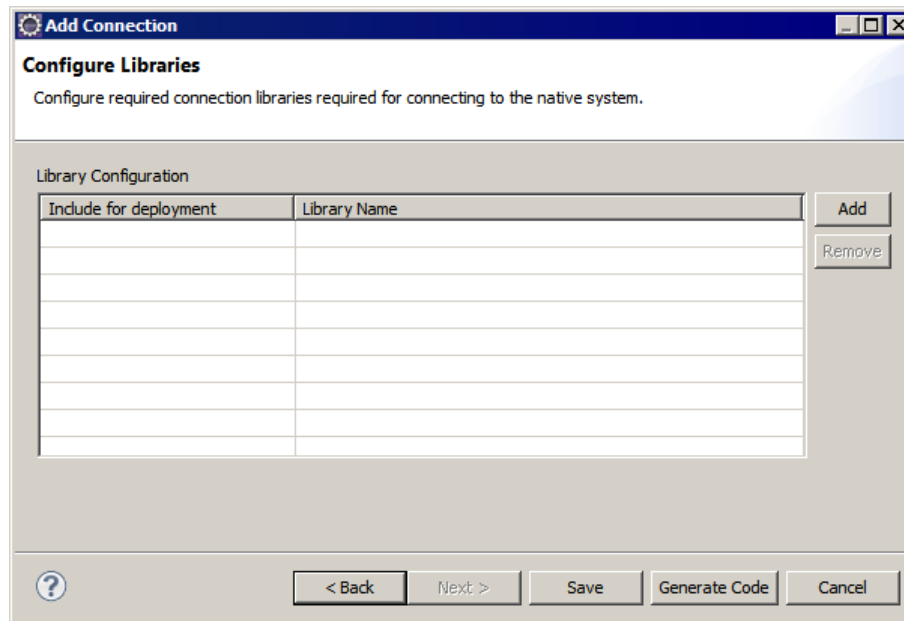
Property	Value
Default Value	3306
Options	- Mandatory

- e. Enter the following values for the catalog attribute:

Property	Value
Name	catalog
Display Name	Catalog
Description	Database catalog to connect. [Required]
Datatype	String
Min Length	0
Max Length	255
Options	- Mandatory

6. Click **Next**.

The **Configure Libraries** page appears.



7. Click **Add** to select each library that the connector requires to connect to the MySQL database.

8. Click **Generate Code**.

The Informatica Connector generates Java source code in the Connectivity folder of the connector project.

9. In the Connectivity folder of the connector project, double-click each file to review the source code and update it as required.

Update the following source code files in the Connectivity folder:

MySQLConnectInfoAdapter.java

This file contains optional methods for connection validation and attribute dependencies.

MySQLConnection.java

This file contains methods that open and close the connection to the MySQL service.

Note: If you regenerate code for the connection project, the Informatica Connector Toolkit does not regenerate code for the user-exposed source code visible in the Informatica perspective. You have to manually edit the source code and make changes if you add, remove, or modify connection attributes.

Test and Debug the Connection to the Data Source

After you define the connection attributes, you can test and debug the connection to the data source.

Create a debug configuration in Eclipse IDE from the **Run** menu. Select the connector project to debug and provide the connection type and connection properties. You can also set breakpoints in the code that you want to debug.

After you create a debug configuration, you can launch the **Test Connection** dialog box to test and debug the connection definition.

1. In the **Connector Progress** view, select **Test Connection**.

The **Test Connection** dialog box appears with the default Java Virtual Machine (JVM) environment settings. The Informatica Connector Toolkit uses the JVM settings to run the debug configuration.

2. Edit the JVM environment settings, if required. Ensure that you use the same port number that appears in JVM settings for the connection properties in the debug configuration.

3. Click **Next**.

The **Connection Details** page appears with the connection attributes that you defined for the connector.

4. Enter values for the connection attributes.

5. Click **Test Connection**.

A success message appears if the connection to the data source is successful.

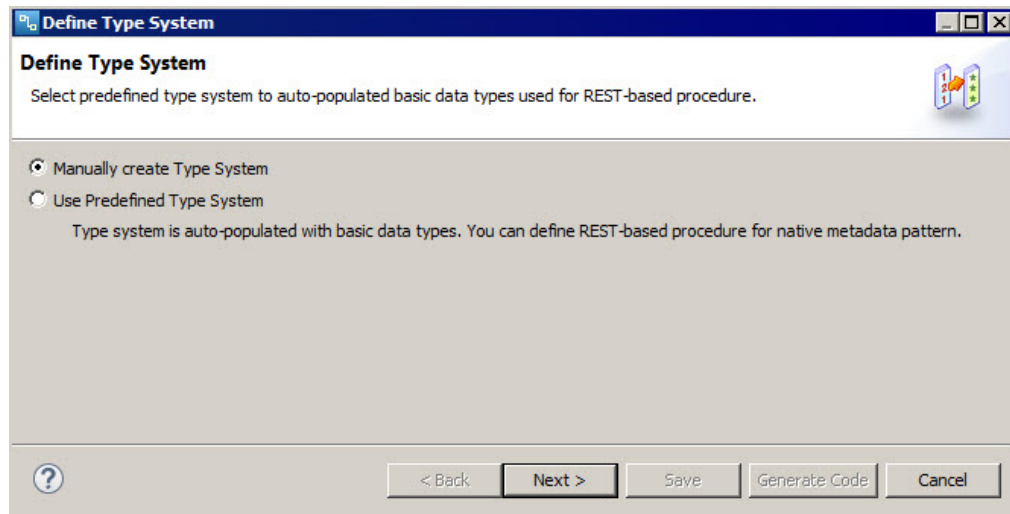
Note: You cannot use the Informatica Connector Toolkit to test attribute dependencies.

Step 3. Define the Connector Type System

Define the type system to specify the datatypes supported by the data source and how they map to the datatypes supported by the Informatica platform.

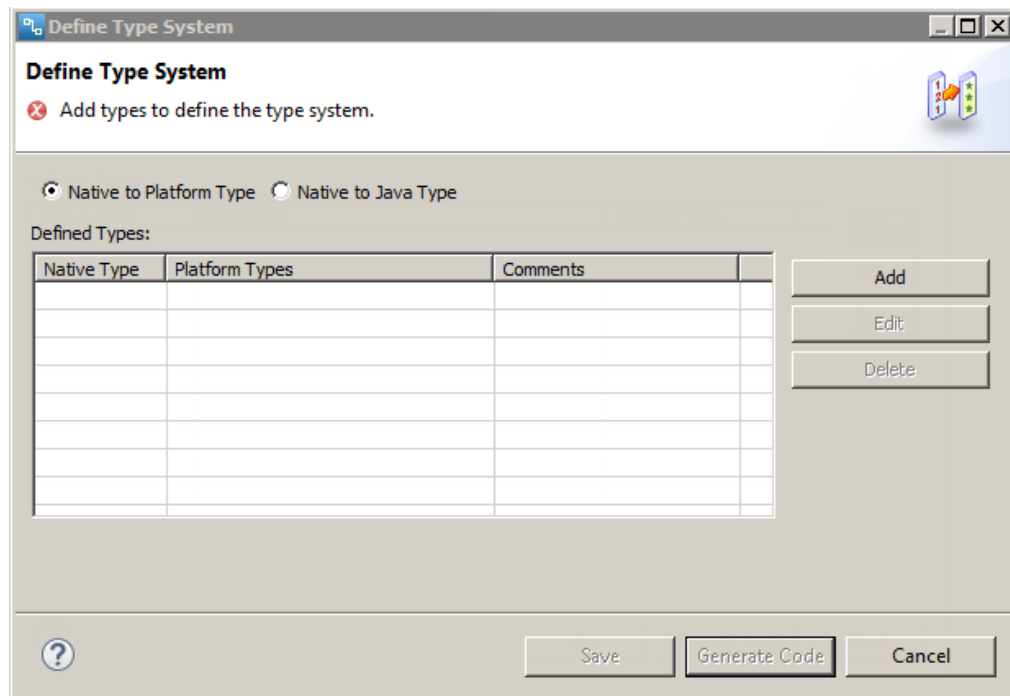
1. In the **Connector Navigator** view, right-click the project and select **Define Typesystem**.

The **Define Type System** dialog box appears.



2. To manually define a type system that matches the data types in the data source, select **Manually Create Type System**.

The **Define Type System** dialog box appears.



3. To map the native data types to the Informatica platform data types, select **Native to Platform Type**.

4. Click **Add** to configure each type system attribute.

The **Add Type** dialog box appears.

Add Type

Enter a name for the type.

Type Name:

Comments:

Best Platform Type for Read:

Length Attributes

Max Length:

Default Length:

Length Unit:

Restrictions: ☒ Allow Length Modifications ☒ Allow Variable Allocation

Platform Types	Mapped	Best Native Type to Write	Best Platform Type for Read	
integer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
bigInteger	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
string	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
binary	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
decimal	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
double	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

? OK Cancel

- a. Enter the following values for the Timestamp attribute:

Property	Value
Type Name	Timestamp
Comments	Stores a period of time in years, months, days, hours, minutes, and seconds.
Platform Type	Date

Property	Value
Max Scale	6
Default Scale	6
Supported Units	Year, Month, Day, Hour, Minute, and Second.

- b. To map the Informatica Platform DATE data type to the MySQL data type Timestamp as one of the Best Native Type to Write, select Mapped in the DATE row and then select Best Native Type to Write.
- c. Enter the following values for the Double attribute:

Property	Value
Type Name	Double
Comments	A double-precision, floating-point number that is accurate to approximately 15 decimal places. Permissible values are -1.7976931348623157E+308 to -2.2250738585072014E-308, 0, and 2.2250738585072014E-308 to 1.7976931348623157E+308.
Platform Type	Double
Max Precision	15
Default Precision	15
Precision Radix	Decimal
Max Exponent	11
Min Exponent	0
Exponent Radix	Binary

- d. To map the Informatica Platform Double data type to the MySQL data type Double as one of the Best Native Type to Write, select Mapped in the Double row and then select Best Native Type to Write.
- e. Enter the following values for the String attribute:

Property	Value
Type Name	String
Comments	Variable-sized field of characters.
Platform Type	string
Max Length	8000

Property	Value
Default Length	50
Length Unit	characters
Restrictions	Allow Length Modifications and Allow Variable Allocation.

- f. To map the Informatica Platform string type to the MySQL data type String as one of the Best Native Type to read, select Mapped in the String row and then select Best Native Type to Read.
5. Click **Generate Code**.

Step 4. Define Connector Metadata

Define the MySQL connector metadata. Create a native metadata definition for the MySQL connector to read from and write to MySQL data source.

1. In the Connector Navigator view, right-click the project and select **Add Native Metadata Definition**. The **Add Native Metadata Definition** dialog box appears.

2. Enter the native metadata details.

The following table describes the values to enter:

Property	Value
Name	table
Display Name	Table
Description	Represents a MySQL table.
Pattern Type	Record

3. Select **Add Record Extension**.
4. Click **Add** to configure each attribute.

The **Add Attribute** dialog box appears.

Add Attribute

Enter the attribute details

Name is empty

Name:

Display Name:

Description:

Data Type:

Default Value:

Min Length:

Max Length:

☐ Encrypted

☐ Mandatory

☐ Hidden

Allowed Values:

- a. Enter the following values for the queryHint attribute:

Property	Value
Name	tableType
Display Name	Table Type
Description	Represents the type of record.
Data Type	String
Min Length	0
Max Length	255

- b. Click **OK**.

The **Add Native Metadata Definition** dialog box appears.

5. Click **Next**.

The **Native Metadata Field Definition** page displays.

6. Select **Add Field Extension**.
7. Click **Add** to configure each attribute.
 - a. Enter the following values for the isNullable attribute:

Property	Value
Name	isNullable
Display Name	Is Nullable
Description	Indicates whether the column can contain null values.
Data Type	Boolean
Default Value	True

- b. Enter the following values for the ColumnDefaultValue attribute:

Property	Value
Name	ColumnDefaultValue
Display Name	ColumnDefaultValue
Description	Indicates the default values that the column can contain.
Data Type	String
Min Length	0
Max Length	255

- c. Click **OK**.
The **Native Metadata Field Definition** dialog box appears.

8. Click **Next**.
The **Native Metadata Read Capability** page appears.

9. Select **Enable Read Capability**.
10. Select the following options:
 - **Required** in the **Lookup** section.
 - **Filter** in the **Join and Filter** section.
 - **Platform** as the type of **Filter**.
 - **Sort** in the **Sort and Select** section.

Implement the `initDataAdapter` API in the `MysqlDataAdapter.java` file to add the lookup, filter, and sort logic.

11. Click **Add** to configure each attribute.
The **Add Attribute** dialog box appears.
 - a. Enter the following values for the preSQL attribute:

Property	Value
Name	preSQL
Display Name	Pre-SQL

Property	Value
Description	SQL queries to run before the connector reads data from the source.
Data Type	String
Min Length	0
Max Length	255
Supports Parameter	Partial Parameterization

- b. Enter the following values for the **rowLimit** attribute:

Property	Value
Name	rowLimit
Display Name	Row Limit
Data Type	Integer
Min Length	0
Max Length	10
Supports Parameter	No
Options	Override Partition

- c. Enter the following values for the **rowLimit** attribute:

Property	Value
Name	postSQL
Display Name	post SQL
Data Type	String
Min Length	0
Max Length	255
Supports Parameter	Partial Parameterization

- d. Enter the following values for the **rowLimit** attribute:

Property	Value
Name	rowOffset
Display Name	Row Offset
Data Type	Integer
Min Length	0
Max Length	0
Supports Parameter	Full Parameterization
Options	Override Partition

- e. Click **OK**.

The **Native Metadata Read Capability** dialog box appears.

12. Click **Next**.

The **Native Metadata Write Capability** page displays.

13. Select **Enable Write Capability**.

14. Click **Add**.

The **Add Attribute** dialog box appears.

- a. Enter the following values for the **truncateTargetTable** attribute:

Property	Value
Name	truncateTargetTable
Display Name	Truncate Target Table
Description	Indicates whether to truncate the target table before starting the write operation.
Data Type	Boolean
Default Value	False

- b. Enter the following values for the partitionID attribute

Property	Value
Name	partitionID
Display Name	Partition ID
Data Type	String
Min Length	0
Max Length	50
Supports Parameter	No
Override Partition	-Override Partition

15. Click **Next**.

The **Native Metadata Partitioning Capability** page appears.

16. To specify partitioning capability for the native metadata object, select the following options:
- Select **Supports Partitioning Capability for Read** to enable the partition capability for the read operation.
 - Select **Dynamic** to specify the partitioning method for the read operation.
 - Select **Supports Partitioning Capability for Write** to enable the partition capability for the write operation.
 - Select **Dynamic** to specify the partitioning method for the write operation.

Extend the `AutoPartitioningMetadataAdapter` class to implement the partition logic.

17. Click **Next**.

The **Import Dialog Box Settings** page appears.

18. To define the import options that appear in the Developer tool when a connector consumer creates a data object, select the following options:
 - Select the **Allow Multi Select** option to allow the connector user to select multiple importable objects.
 - Select the **Display Filter By Name** option to display the filter by name option.
 - Select the **Show Entity** option to show the entity details.
19. Click **Generate Code**.

The Informatica Connector generates source code in the Metadata and Runtime folders of the connector project.
20. In the Metadata folder of the connector project, double-click the `MySQLMetadataAdapter.java` file to review the source code and update it as required. `MySQLMetadataAdapter.java` contains methods for importing metadata.

Provide code for the following methods:

 - `populateObjectCatalog()`
 - `populateObjectDetails()`

For more information about these methods, see the *Informatica Connector Toolkit API Reference*.

Test Metadata from the Data Source

After you define the native metadata objects, you can test metadata that you import from the data source.

To debug when you test metadata from the data source, use the same debug configuration that you used to test the connection to the data source. You can also set breakpoints in the code that you want to debug.

After you define the debug configuration, you can launch the **Test Metadata** dialog box to test the connection definition and test the metadata import from the data source.

1. In the **Connector Progress** view, select `Test<NM0Name>` under the **Test Metadata** section.

The **Test Metadata** dialog box appears with the connection attributes that you defined for the connector.
2. Enter values for the connection attributes to connect to the data source.
3. Click **Connect**.

The connector retrieves the metadata from the data source and the metadata appears in the **Test Metadata** page.
4. Browse and verify the retrieved metadata.
5. Click **Close**.

Step 5. Define the Run-Time Interfaces

Use the run-time interfaces available in C to specify how the MySQL connector reads and writes comments.

1. In the **Connector Progress** view, right-click the project and select **Runtime > Set Up**.

The **Run-time Implementation** wizard appears.
2. Select **Implement in C** and click **Next**.

The **Supported Platforms** page appears.
3. Click **Add**.

The **Add Platform** page appears.

4. Select **NT** as the platform.
5. In the **Compile** section, click **Add**. The **Add Directory** dialog box appears.
 - a. Browse to the location that contains the library files required for the MySQL connector. For example, browse to `C:\mysql-connector-c-6.1.5-winx64\mysql-connector-c-6.1.5-winx64\include`.
 - b. Click **OK** to add the include paths.
6. Ensure that Visual Studio is selected as the IDE for the run-time project.
7. In the **Link** section, click **Add**. The **Add Library** dialog box appears.
 - a. Browse to the directory that contains the link library files required for the MySQL connector. For example, browse to `C:\mysql-connector-c-6.1.5-winx64\mysql-connector-c-6.1.5-winx64\lib`.
 - b. Click **OK** to add the library paths.
8. Click **OK** to return to the **Supported Platform** page.
9. Click **Finish**.

The **Run-time Implementation** wizard generates the C project files in the runtime folder.

10. Use Visual Studio to open the C++ project in the Runtime folder of the table NMO and implement the following functions:
 - `INFAADPInitPlugin`
 - `INFAADPDeinitPlugin`
 - `INFAADPInitDataSourceOperation`
 - `INFAADPDeinitDataSourceOperation`
 - `INFAADPInitdataSession`
 - `INFAADPDeinitDataSession`
 - `INFAADPBegindataSession`
 - `INFAADPEndDataSession`
 - `INFAADPDisConnect`
 - `INFAADPConnect`
 - `INFAADPRead`
 - `INFAADPWrite`

For more information about these methods, see the *Informatica Connector Toolkit API Reference*.

Test the Read and Write Capabilities of the Connector

Before you deploy the connector in the Informatica domain, you can test the read and write capabilities of the connector.

After you define the connector run-time components, you can use the **Test Read** and **Test Write** wizards to test the read and write capabilities of the connector. After you debug and fix issues in the read and write capabilities of the connector, you can deploy the connector in the Informatica domain.

You can test the read and write capabilities of the connector only for the Windows platform.

Test the Read Capability of the Connector

When you test the read capability of the connector, you test the connection definition, metadata of the data source, and operations that the connector supports. After you specify the test settings and run the test, you can view the result of the read operation, read operation statistics, and the log file.

To debug the code, use the same debug configuration that you used to test the connection and metadata components of the connector. You can also set breakpoints in the code that you want to debug.

After you define the debug configuration, you can launch the **Test Read** dialog box to test the read capabilities of the connector.

1. In the **Connector Progress** view, select the native metadata object that appears under **Test Read**.
The **Test Read** dialog box appears with the default JVM environment settings and tracing level. The Informatica Connector Toolkit uses the JVM settings to run the debug configuration.
2. If required, edit the JVM environment settings. Ensure that you use the same port number that appears in JVM settings for the connection properties in the debug configuration.
3. Select the required tracing level. The default is normal. Based on the amount of detail that you require in the log file, you can override the default tracing level.

You can set the following types of tracing level:

None

Does not override the default tracing level.

Terse

Logs initialization information and error messages and notification of rejected data.

Normal

Logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. Summarizes mapping results, but not at the level of individual rows. This is the default tracing level.

Verbose Initialization

In addition to normal tracing, logs additional initialization details, names of index and data files used, and detailed statistics.

Verbose Data

In addition to verbose initialization tracing, logs each row. You can also get detailed statistics on where string data was truncated to fit the precision of a column.

4. Click **Next**.
The connection attributes that you defined for the connector appears.
5. Enter values for the connection attributes to test the connection to the data source.
6. Click **Connect**.
The **Test Metadata** page appears with the metadata imported from the data source.
7. Select the native metadata objects and the corresponding native metadata fields to test the read operation.
8. Click **Next**.
Based on whether the selected native metadata object supports join and filter operations, the **Join Condition** page or **Filter Condition** page appears.
9. If the native metadata object supports join operation, specify an expression in the **Join Condition** page.

- To specify an Informatica platform expression for the join operation, perform the following steps:
 1. In the **Definition** section, select the native metadata object for which you want to specify the join condition.
 2. To change the join order, click **Move Up** or **Move Down**.
 3. Specify the join type in **Join condition**.
 4. To define a join condition, select values in the **Left Field**, **Operator**, and **Right Field** columns.
 5. To add additional join conditions, click **Add**.
 6. To remove a join condition, select the join condition and click **Remove**.
 - To specify a native expression for the join operation, enter the expression in the **Definition** section.
10. If the native metadata object supports filter operation, select the **Configure Filter** option in the **Filter Condition** page and specify the filter condition.
 - To specify an Informatica platform expression for the filter operation, perform the following steps:
 1. In the **Definition** section, click **Add** to add an Informatica platform expression.
 2. In the **Field** column, select the field to use in the expression.
 3. In the **Operation** column, select a conditional operator to use in the expression.
 4. In the **Value** column, enter a value for the conditional expression.
 - To specify a native expression for the filter operation, enter the expression in the **Definition** section.
 11. After you specify expressions for the native metadata object, click **Next**.
The **Read Capability** page appears.
 12. Specify values for the read capability attributes and then click **Run**.
The **Result** page appears. You can view the result of the read operation, read operation statistics, and the log file in the **Result** page.
 13. Click **Close**.

Test the Write Capability of the Connector

When you test the write capability of the connector, you test the components of the connector and write sample data to the data source. After you specify the test settings and run the test, you can view the result of the write operation, write operation statistics, and the log file.

To debug the code, use the same debug configuration that you used to test the connection and metadata components of the connector. You can also set breakpoints in the code that you want to debug.

After you define the debug configuration, you can launch the **Test Write** dialog box to test the write capabilities of the connector.

1. In the **Connector Progress** view, select the native metadata object that appears under **Test Write**.
The **Test Write** dialog box appears with the default JVM environment settings and tracing level. The Informatica Connector Toolkit uses the JVM settings to run the debug configuration.
2. If required, edit the JVM environment settings. Ensure that you use the same port number that appears in JVM settings for the connection properties in the debug configuration.
3. Select the required tracing level. The default is normal. Based on the amount of detail that you require in the log file, you can override the default tracing level.

You can set the following types of tracing level:

None

Does not override the default tracing level.

Terse

Logs initialization information and error messages and notification of rejected data.

Normal

Logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. Summarizes mapping results, but not at the level of individual rows. This is the default tracing level.

Verbose Initialization

In addition to normal tracing, logs additional initialization details, names of index and data files used, and detailed statistics.

Verbose Data

In addition to verbose initialization tracing, logs each row. You can also get detailed statistics on where string data was truncated to fit the precision of a column.

4. Click **Next**.

The connection attributes that you defined for the connector appears.

5. Enter values for the connection attributes to test the connection to the data source.

6. Click **Connect**.

The **Test Metadata** page appears with the metadata imported from the data source.

7. Select a native metadata object to test the write operation.

The metadata of the native metadata object along with the data type, scale, and precision appears in the **Test Write** page.

8. Select the columns to which you want to write data.

9. Click **Next**.

The **Test Data** page appears.

10. In the Test Data page, you can load test data from a file or you can generate test data.

- To load the test data from a file, perform the following steps:

1. Select the **Load from a File** option. You must load a comma-delimited TXT file or CSV file.

Note: The date and time data types in the file must have the following timestamp format: MM/DD/YYYY hh24:mm:ss

2. Click **Browse** and select the file that contains the test data.

- To generate test data, perform the following steps:

1. Select the **Auto generate data** option.

2. Enter the number of rows to generate. You can specify a maximum of 1000 rows.

3. Click **Generate**. The test data appears in the **Data Preview** section.

4. If required, you can edit the test data that appears in the **Data Preview** section.

11. Select an insert, update, or delete operation that you want to perform on the target object.

To perform an update or delete operation, the target object must contain a primary key. If you auto-generate the data, edit the value of the primary key column in the preview section to match with a record in the target object.

12. After you load a test data file or generate test data, click **Next**.

The **Write Capability** page appears.

13. Specify values for the write capability attributes and then click **Run**.

The **Result** page appears. You can view the result of the write operation, write operation statistics, and the log file in the **Result** page.

14. Click **Close**.

Informatica Connector Toolkit Eclipse Plug-in Tip

Use the following tip when you create a connector.

Use the Project Explorer view to view the artifacts and packages that the Informatica Connector Toolkit Eclipse Plug-in generates.

To display the **Project Explorer** view, click **Window > Show View > Other**. Then, expand the **General** folder, select **Project Explorer view**, and click **OK**.

CHAPTER 4

Exporting a Connector

This chapter includes the following topics:

- [Exporting a Connector Overview, 42](#)
- [Export the Connector, 43](#)

Exporting a Connector Overview

In this lesson, you will export the MySQL connector developed using the Informatica connector toolkit.

Lesson Concepts

Publish the connector project to export the connector bundle. This connector bundle consists of the files that are required by the Informatica Cloud to create connection objects and access the source data.

Lesson Objectives

In this lesson, you export the MySQL connector.

Lesson Prerequisites

Before you start this lesson, complete the following prerequisite:

- Install the Informatica Connector Toolkit. For more information, see [“Installing the Informatica Connector Toolkit Overview” on page 11](#).
- Create a Connector. For more information, see [“Creating a Connector Overview” on page 16](#).

Lesson Timing

Set aside 10 minutes to complete this lesson.

Export the Connector

You can use the Informatica Connector Toolkit to publish the connector.

1. In the **Connector Navigator** view, right-click the project and select **Publish Connector**.

The **Publish Informatica Connector** dialog box appears.

Export/Publish Informatica Connector

Export and Publish Settings
Provide the export and publish settings for the connector.

Export Location:

Version: 1.0.0

Export Connector

☒ Informatica Cloud

☒ Specify Plugin ID

Plugin ID: Plugin Version:

2. If you want to change the export location, click **Browse** to configure a different location.
3. If you want to specify the Plugin ID, select the **Specify Plugin ID** checkbox.
4. To view a summary of the artifacts included in the connector bundle, click **Next**.
5. Click **Finish**.

Note: By default, you can use the Informatica Connector Toolkit to build connector for Informatica Intelligent Cloud Services.

APPENDIX A

Frequently Asked Questions

This appendix includes the following topic:

- [Informatica Connector Toolkit Frequently Asked Questions, 44](#)

Informatica Connector Toolkit Frequently Asked Questions

Review the frequently asked questions to answer questions you might have when you use the Informatica Connector Toolkit to develop a connector.

What must I do if I get the message asking me to check the JDK version even if JDK version 1.8 is installed on the machine?

In the Eclipse IDE, navigate to **Window > Preferences > Java > Compiler** and set **Compiler compliance level** to 1.8.

When I generate code for native metadata, I get an error message that project description files are out of sync with the file system. Does this error impact code generation?

No. This error does not impact code generation. If you still want to resolve this error, you can select the **Refresh using native hooks or polling** option from the **Window > Preferences > General > Workspace** menu in the Eclipse IDE. If you choose to refresh using native hooks or polling, there may be considerable delay in generating code.

Is there a log file that I can refer when I encounter errors while I generate code or publish the Connector?

Refer to the <connector_name>_codebuilder.log file for any errors that you encounter when you generate code or publish the connector. The <connector_name>_codebuilder.log file is available in the <Eclipse workspace>/<connector project> folder.

After I published a connector, what must I do if I made changes only to the connection attributes, record extensions, field extensions, read capability, or write capability extensions?

If you added attributes, use the **Upgrade** option in the **Publish Informatica Connector** wizard to republish the connector. The Informatica Connector Toolkit supports upgrade of a connector only if the changes are additive. If you renamed or deleted any of the existing attributes, uninstall Informatica services, re-create the domain, and then install the connector.

How to enable a Cloud Data Integration connector developed using Informatica Connector Toolkit to use the proxy server?

To enable a Cloud Data Integration connector developed using Informatica Connector Toolkit to use the proxy server, see <https://kb.informatica.com/howto/6/Pages/22/564235.aspx>.

INDEX

A

attribute
connection [20](#)

C

capability
read [29](#)
write [29](#)
connection
attribute [20](#)
test [25](#)
connector
create [9](#)
install [43](#)
publish [43](#)

D

define
native metadata [29](#)
type system [26](#)

E

Eclipse
install [14](#)
extensions
field [29](#)
record [29](#)

I

Informatica Connector Toolkit
install [11, 14](#)
overview [7](#)

install
connector [43](#)
Eclipse [14](#)
Informatica Connector Toolkit [11, 14](#)

M

map
datatype [26](#)
MySQL
sample [9](#)

O

overview
Informatica Connector Toolkit [7](#)

P

publish
connector [43](#)

S

sample
MySQL [9](#)

T

test
connection [25](#)
metadata [36](#)
read capability [38](#)
write capability [39](#)