Informatica

Informatica® Cloud Data Integration

JDBC V2 Connector

Informatica Cloud Data Integration JDBC V2 Connector
November 2024

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

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: 2024-12-18

Table of Contents

Preface

Use *JDBC V2 Connector* to learn how to use the JDBC V2 connection to read from or write to Aurora PostgreSQL and other databases that support the Type 4 JDBC driver by using Data Integration. Learn to create a connection, develop and run mappings, mapping tasks, dynamic mapping tasks, and data transfer tasks in Data Integration.

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 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 Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

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 Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

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

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to JDBC V2 Connector

You can use JDBC V2 Connector to connect to databases that support the JDBC Type 4 driver with Data Integration, such as PostgreSQL and Azure SQL.

When you use JDBC V2 Connector, you can create a JDBC V2 connection and use the connection in Data Integration mappings and tasks. You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality.

In advanced mode, the mapping can run on Amazon Web Services, Google Cloud Platform, the Microsoft Azure environment, Azure Kubernetes Service (AKS), or Elastic Kubernetes Service (EKS).

Note: JDBC V2 Connector provides generic connectivity to databases through the JDBC protocol. You can use this connector to connect to JDBC-compliant databases when Informatica does not provide specific database connectors or when you do not have access to the database connectors available for Cloud Data Integration. Informatica does not guarantee that the functionality or performance of JDBC V2 Connector will be equivalent to that of the available database connectors. You can connect to any JDBC-compliant databases you choose, but these databases might not have been evaluated for compatibility with this connector.

JDBC V2 Connector assets

Create assets in Data Integration to integrate data using JDBC V2 Connector.

When you use JDBC V2 Connector, you can include the following Data Integration assets:

- Data transfer task
- Dynamic mapping task
- Mapping
- Mapping task

Note: Dynamic mapping task applies only to mappings in advanced mode.

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

CHAPTER 2

Connections for JDBC V2

Create a JDBC V2 connection to access data from any database that supports the Type 4 JDBC driver, such as PostgreSQL and Azure SQL.

You can use a JDBC V2 connection to specify sources, targets and lookups in a mappings and mapping tasks.

Prerequisites

Before you create a JDBC V2 connection to read from or write to databases that support the JDBC Type 4 driver, complete the prerequisites.

Install the Type 4 JDBC driver

To read from or write to JDBC V2 objects, you need to install the Type 4 JDBC driver on the Secure Agent machine.

1. Download the latest Type 4 JDBC driver version that your database supports from the third-party vendor site.
   If you want to use JDBC V2 Connector to connect to Aurora PostgreSQL, download the Aurora PostgreSQL driver. Informatica has certified Aurora PostgreSQL driver 42.2.6 for JDBC V2 Connector.
2. Install the Type 4 JDBC driver for the database on the Secure Agent machine and perform the following tasks:
   a. Navigate to the following directory on the Secure Agent machine:
      `<Secure Agent installation directory>/ext/connectors/thirdparty/`
   b. Create a folder and add the driver based on the type of mapping that you want to configure.
      For mappings, add the driver in the following folder:

      `informatica.jdbc_v2/common`

      For mappings in advanced mode, add the driver in the following folders:

      `informatica.jdbc_v2/common`

      `informatica.jdbc_v2/spark`
3. Restart the Secure Agent.
   If you update the driver on the Secure Agent machine while the mapping in advanced mode runs, you need to restart the Secure Agent.

Connect to JDBC V2

Let's configure the JDBC V2 connection properties to connect to JDBC-compliant databases.

Before you begin

Before you get started, you'll need to install the Type 4 JDBC driver on the Secure Agent machine to establish a JDBC V2 connection.

Check out "Prerequisites" on page 7 to learn more about the configuration prerequisites.

Connection details

The following table describes the basic connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.
Type	JDBC V2
Runtime Environment	The name of the runtime environment where you want to run tasks. Select a Secure Agent or serverless runtime environment. For more information about how to configure a serverless environment, see "Use the serverless runtime environment" on page 10.
JDBC Driver Class Name	Name of the JDBC driver class. For example, to connect to Aurora PostgreSQL, specify the following driver class name: org.postgresql.Driver For more information about which driver class to use with specific databases, see the corresponding third-party vendor documentation.
Connection String	Connection string to connect to the database. Use the following format to specify the connection string: `jdbc:<subprotocol>:<subname>` For example, the connection string for the Aurora PostgreSQL database type is `jdbc:postgresql://<host>:<port>[/dbname]`. For more information about the connection string to use with specific drivers, see the corresponding third-party vendor documentation. You can also connect to SSL-enabled Aurora PostgreSQL databases in mappings in advanced mode. For more information, see "Connect to SSL-enabled databases for mappings in advanced mode" on page 10.
User Name	The user name to connect to the database.
Password	The password to connect to the database.

Advanced settings

The following table describes the advanced connection properties:

Property	Description
Database Type	The database type to which you want to connect. Select one of the following database types: - PostgreSQL. Connect to the Aurora PostgreSQL database hosted in the Amazon Web Services or the Microsoft Azure environment. - Azure SQL Database. Connect to Azure SQL Database hosted in the Microsoft Azure environment. - Others. Connect to any database that supports the Type 4 JDBC driver, such as Salesforce Data Cloud. Default is Others.
Schema Name	The schema name used for the JDBC object. If you don't specify the schema name, all the schemas available in the database are listed. To read from or write to Oracle public synonyms, enter PUBLIC.
Connection Environment SQL	The SQL statement to set up the database environment when you connect to a PostgreSQL database. The database environment applies for the entire session that uses this connection. For example, you can enter this statement to set the time zone: `SET timezone to 'America/New_York';`
Additional Security Properties	Masks sensitive and confidential data of the connection string that you don't want to display in the session log. Specify the part of the connection string that you want to mask. When you create a connection, the string you enter in this field appends to the string that you specified in the **Connection String** field.
Enable Auto Commit[1]	Specifies whether the driver supports connections to automatically commit data to the database when you run an SQL statement. When disabled, the driver does not support connections to automatically commit data even if the auto-commit mode is enabled in the JDBC driver. Default is disabled.
Support Mixed-Case Identifiers	Indicates whether the database supports case-sensitive identifiers. When enabled, the Secure Agent encloses all identifiers within the character selected for the SQL Identifier Character property. Default is disabled.
SQL Identifier Character	Type of character that the database uses to enclose delimited identifiers in SQL queries. The available characters depend on the database type. Select **None** if the database uses regular identifiers. When the Secure Agent generates SQL queries, it does not place delimited characters around any identifiers. Select a character from the list based on what delimiter the database uses for identifiers. When the Secure Agent generates SQL queries, it encloses delimited identifiers within this character.

[1] Doesn't apply to mappings in advanced mode.

Connect to SSL-enabled databases for mappings in advanced mode

You can use JDBC V2 connection in mappings in advanced mode to connect to an SSL-enabled JDBC-complaint database. To run mappings in advanced mode with SSL-enabled JDBC-complaint databases, you need to download the SSL certificates to the Secure Agent machine, and then perform certain prerequisite tasks.

1. Specify the JDBC URL in the JDBC V2 connection properties.
   To connect to an SSL-enabled Aurora PostgreSQL database, specify the following JDBC URL:

   `jdbc:postgresql://<host>:<port>/dbname?sslmode=verify-ca&sslrootcert=<Location of the SSL certificate on the Secure Agent machine>` , where the values for sslmode supports `verify-ca` and `verify-ca`.

   For example, `jdbc:postgresql://aurorapostgres-appsdk.abc.ap-south-1.rds.amazonaws.com:5432/JDBC_V2?sslmode=verify-full&sslrootcert=/data/home/qamercury/cloud_td/Aurora_cert/rds-combined-ca-bundle.pem`.
2. After you specify the JDBC URL in the JDBC V2 connection properties, in the advanced session properties of the mapping task, select **advanced.custom.property** as the session property name.
3. In the session property value, specify the following value:
   Spark.NeedUserCredentialFileForAdapter=true&:Spark.UserCredentialDirOnDIS=<Location of the SSL certificate on the Secure Agent machine>
   - Spark.NeedUserCredentialFileForAdapter. When you set this property to true, the Secure Agent copies the SSL certificate from the Secure Agent machine to the advanced cluster.
   - Spark.UserCredentialDirOnDIS. When you set this property to the location of the SSL certificates, the Secure Agent uses the specified location to get the SSL certificate.
     This property is optional. If you do not specify this property, the Secure Agent considers the following default location: `/infa/user/credentials`

Use the serverless runtime environment

You can use a serverless runtime environment hosted on AWS or Azure to connect to JDBC-compliant databases.

Before you configure a JDBC V2 connection using the serverless runtime environment, perform the following tasks:

- Add the JDBC driver JAR files in the Amazon S3 bucket or Azure container in your AWS or Azure account.
- Configure the .yml serverless configuration file.

Add the JDBC driver JAR files in the Amazon S3 bucket or Azure container in your AWS or Azure account

Perform the following steps to configure a JDBC V2 connection in a serverless runtime environment:

1. Create the following structure for the serverless agent configuration in AWS or Azure:
   `<Supplementary file location>/serverless_agent_config`
2. Add the JDBC driver files in the Amazon S3 bucket or Azure container in the following location in your AWS or Azure account:
   `<Supplementary file location>/serverless_agent_config/common`

3. For mappings in advanced mode, additionally add the JDBC driver files in the following location in the Amazon S3 bucket or Azure container:
   `<Supplementary file location>/serverless_agent_config/spark`

Configure the .yml serverless configuration file

Perform the following steps to configure the .yml serverless configuration file in the serverless runtime environment:

1. Copy the following code snippet to a text editor based on the mappings that you want to run in a serverless environment:
   - For mappings that do not apply in advanced mode, add the following code snippet:

     ```
     version: 1
     agent:
       dataIntegrationServer:
         autoDeploy:
           jdbcv2:
             common:
               - fileCopy:
                   sourcePath: common/<Driver_filename>
               - fileCopy:
                   sourcePath: common/<Driver_filename>
     ```
   - For mappings in advanced mode, add the following code snippet:

     ```
     version: 1
     agent:
       elasticServer:
         autoApply:
           jdbcv2:
             common:
               - fileCopy:
                   sourcePath: common/<Driver_filename>
               - fileCopy:
                   sourcePath: common/<Driver_filename>
             spark:
               - fileCopy:
                   sourcePath: spark/<Driver_filename>
               - fileCopy:
                   sourcePath: spark/<Driver_filename>
     ```

   where the source path is the directory path of the driver files in AWS or Azure.
2. Ensure that the syntax and indentations are valid, and then save the file as `serverlessUserAgentConfig.yml` in the following AWS or Azure location: `<Supplementary file location>/serverless_agent_config`
   When the .yml file runs, the JDBC driver files are copied from the AWS or Azure location to the serverless agent directory.

For more information about how to configure and use the serverless environment, see "Serverless runtime environment setup" in *Runtime Environments* in the Administrator help.

CHAPTER 3

Mappings in JDBC V2 Connector

Use the Mapping Designer in Data Integration to define and configure a mapping for JDBC V2 sources and targets.

Add the Source and Target transformations in the mapping canvas and configure the JDBC V2 source and target properties. In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality. You can also use Monitor to monitor the jobs.

When you run a JDBC V2 mapping task based on either a mapping or a mapping in advanced mode and it completes, you can view the number of processed and failed rows for each source and target in the job details page.

JDBC V2 sources in mappings

When you configure a mapping to use a JDBC V2 source, you can configure the source properties.

Specify the name and description of the JDBC V2 source. Configure the source and advanced source properties for the JDBC V2 object.

You can parameterize the source and advanced source properties for the JDBC source object in a mapping. You can also parameterize the JDBC V2 source connection and source object with values specified in a parameter file when you configure a mapping in advanced mode.

The following table describes the JDBC V2 source properties that you can configure in a Source transformation:

Property	Description
Connection Name	Name of the source connection.
Source Type	Source type. Select one of the following source types: - **Single Object**. Select to specify a single JDBC V2 object. - **Multiple Objects**. Select to specify multiple JDBC V2 objects. Use the advanced relationship option to define the relationship for the objects that you want to join. - **Parameter**. Select to specify a parameter name. You can configure the source object in a mapping task associated with a mapping that uses this Source transformation. - **Query**. Select to define a valid and supported SQL query in the Source transformation.
Object	Source object for the mapping.

Property	Description
Filter	A simple filter or an advanced filter condition to remove rows at the source. You can improve efficiency by filtering early in the data flow. Specify a simple filter by including a field name, operator, and value. Configure an advanced filter to define a more complex filter condition that can include multiple conditions using the AND or OR logical operators. You can choose from the following filter conditions: - **Not parameterized**. Use a basic filter to specify the object, field, operator, and value to select specific records. - **Completely parameterized**. Use a parameter to represent the field mapping. - **Advanced**. Use an advanced filter to define a more complex filter condition.
Sort	Not applicable.

The following table describes the JDBC advanced source properties that you can configure in a Source transformation:

Advanced Source Property	Description
Pre SQL	The SQL query that the Secure Agent runs before reading data from the source.
Post SQL[1]	The SQL query that the Secure Agent runs after reading data from the source.
Fetch Size	The number of rows that the Secure Agent fetches from the database in a single call.
Table Name	Overrides the table name used in the metadata import with the table name that you specify.
Schema Name	Overrides the schema name of the source object. If you specify the schema name both in the connection and the source properties, the Secure Agent uses the schema name specified in the source properties.
SQL Override	The SQL statement to override the default query and the object name that is used to read data from the JDBC V2 source.
Tracing Level	Determines the amount of details that logs contain. You can select one of the following tracing levels: - **Terse**. The Secure Agent logs initialization information, error messages, and notification of rejected data. - **Normal**. The Secure Agent logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. Summarizes session results, but not at the level of individual rows. - **Verbose Initialization**. In addition to normal tracing, the Secure Agent logs additional initialization details, names of index and data files used, and detailed transformation statistics. - **Verbose Data**. In addition to verbose initialization tracing, the Secure Agent logs each row that passes into the mapping. Also notes where the Secure Agent truncates string data to fit the precision of a column and provides detailed transformation statistics. When you configure the tracing level to verbose data, the Secure Agent writes row data for all rows in a block when it processes a transformation.
[1]Doesn't apply to mappings in advanced mode.	

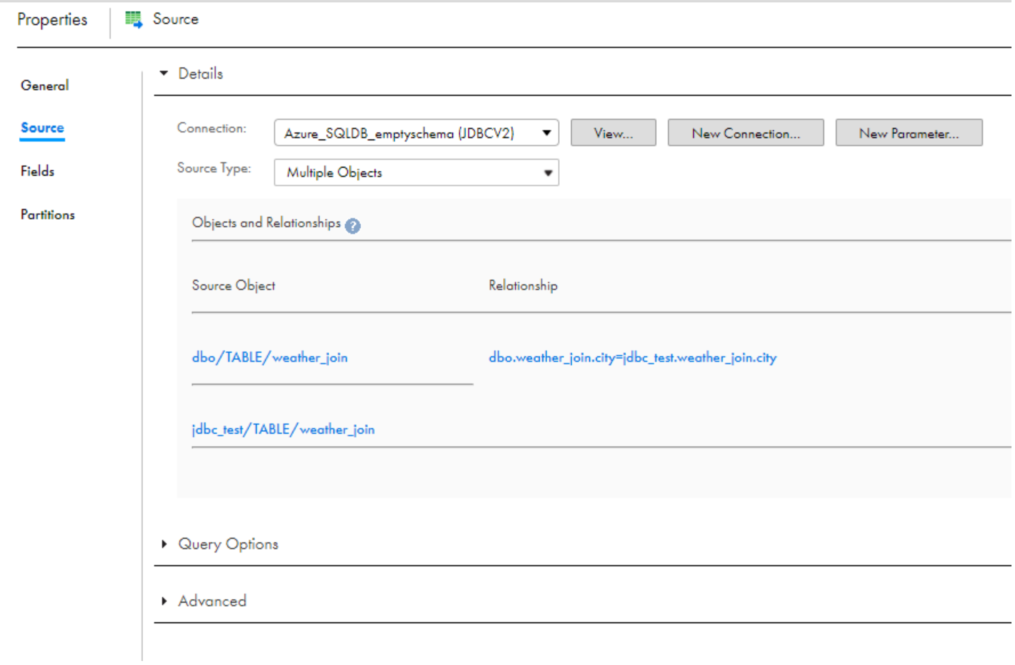
Joining multiple objects

When you create a Source transformation, you can select multiple JDBC V2 objects as the source type and then configure an advanced relationship to combine the tables. Ensure that the source tables do not contain the same name.

1. In the Source transformation, click the **Source Type** as **Multiple Objects**.
2. From the **Actions** menu, select **Add Source Object**, and then select the source object that you want to add from the displayed list:
3. Click **Related Objects Actions** menu, and then select **Advanced Relationship** to define the relationship between the tables:

   Note: The **Add Related Objects** option is not applicable.

   The following image shows an example of an advanced relationship condition defined between the JDBC V2 tables:



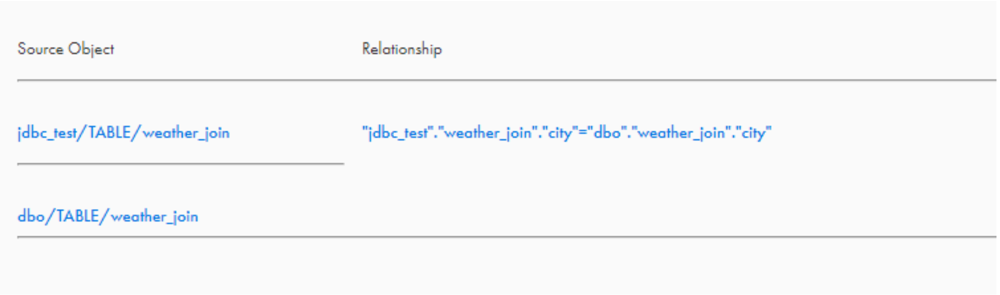
Connection: Azure_SQLDB_emptyschema (JDBCV2)

Source Type: Multiple Objects

Source Object	Relationship
dbo/TABLE/weather_join	dbo.weather_join.city=jdbc_test.weather_join.city
jdbc_test/TABLE/weather_join	

If you join tables that are from different schemas, you must manually add the schema names.

The following image shows an example of an advanced relationship condition defined between two JDBC V2 tables where the schema names are different:



Source Object	Relationship
jdbc_test/TABLE/weather_join	"jdbc_test"."weather_join"."city"="dbo"."weather_join"."city"
dbo/TABLE/weather_join	

JDBC V2 targets in mappings

When you configure a mapping to use a JDBC V2 target, you can configure the target properties.

Specify the name and description of the JDBC target. Configure the target and advanced target properties for the JDBC object in a Target transformation.

You can parameterize the target and advanced target properties for the JDBC target object in a mapping. You can also parameterize the JDBC V2 target connection and target object with values specified in a parameter file when you configure a mapping in advanced mode.

The following table describes the target properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the target connection.
Target Type	Type of target object. Select one of the following target types: - Single. Select to specify a single JDBC object. - Parameter. Select to specify a parameter name. You can configure the target object in a mapping task associated with a mapping that uses this Target transformation.
Object	Name of the target object. You can select an existing object.
Create New at Runtime	Creates a target table at runtime using the object name and path you specify. You can create a target table at runtime only for PostgreSQL or Azure SQL Database. Ensure that you have selected PostgreSQL or Azure SQL Database in the JDBC V2 connection advanced properties based on the database that you want to connect. To create a target table at runtime, provide the following parameters: - **Object name**. Specify the name for the target table. - **Path**. Specify the target table name and schema in the following format: `<Schema-Name>/<TableType>`, where the mandatory value for TableType is *TABLE*.
Operation	Type of the target operation. Select Insert, Update, Upsert, Delete, or Data Driven.
Update columns	The temporary key column that identifies rows in the target table to update, upsert, or delete data.
Data Driven Condition	Flags rows for an insert, update, delete, or reject operation based on the data driven expression you specify. Appears only when the operation type is **Data Driven**. **Note:** To configure a data driven expression in a mapping, you need to specify the expression that uses the IIF function.

The following table describes the JDBC advanced target properties that you can configure in a Target transformation:

Advanced Target Property	Description
Pre SQL	The SQL statement to run before writing data to the target.
Post SQL	The SQL statement to run after writing data to the target.

Advanced Target Property	Description
Truncate Target	Truncates the target table before inserting records to the target.
Reject Truncated/ Overflow Rows	Writes truncated and overflow data to the reject file. If you select Reject Truncated/Overflow Rows, the Data Integration Service sends all truncated rows and any overflow rows to the reject file.
Table Name	Overrides the table name used in the metadata import with the table name that you specify.
Schema Name	Overrides the schema name of the target object. If you specify the schema name both in the connection and the target properties, the Secure Agent considers the schema name specified in the target properties.
Update Mode	Determines the mode how the Secure Agent writes data to the target. Select one of the following modes: - Update As Update. Updates all rows flagged for update if the entries exist. - Update Else Insert. Updates all rows flagged for update if the entries exist in the target. If the entries do not exist, the agent then inserts the entries.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. By default, the agent forwards rejected rows to the next transformation.

Create a target table at runtime

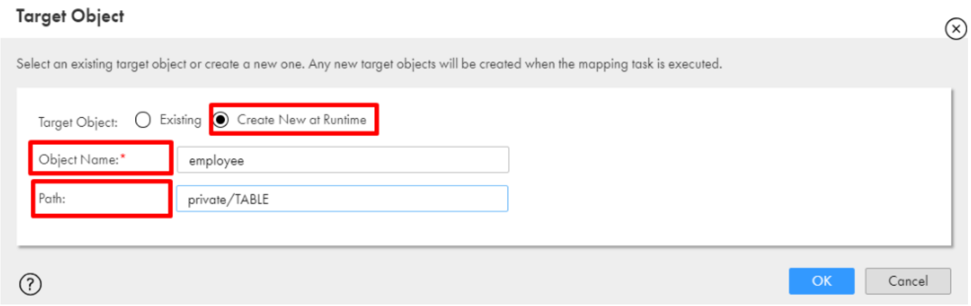
You can use an existing target or create a target in a mapping. If you choose to create the target, the agent creates the target when you run the task.

To specify the target properties, perform the following tasks:

1. Select the Target transformation in the mapping.
2. On the **Incoming Fields** tab, configure field rules to specify the fields to include in the target.
3. To specify the target, click the **Target** tab.
4. Select the target connection.
5. For the target type, choose **Single Object** or **Parameter**.
6. Specify the target object or parameter.

7. To specify a target object, perform the following tasks:

   a. Click **Select** and choose a target object. You can select an existing target object or create a new target object at runtime and specify the object name.



Target Object

Select an existing target object or create a new one. Any new target objects will be created when the mapping task is executed.

Target Object: ○ Existing ● Create New at Runtime

Object Name:* employee

Path: private/TABLE

   You must specify the target table name.

   b. To create a target object at runtime, select **Create New at Runtime**.

   c. Enter the name of the target table that you want to create name in the **Object Name** field.

   d. In the **Path** field, specify the target table name and schema in the following format: `<Schema-Name>/<TableType>`, where the mandatory value for TableType is *TABLE*.

      For example, *private/TABLE*

      In the example, the table is created in the schema named "private" in the database server. If you do not specify a value for the path, the Secure Agent considers the schema name you specified in the connection. If you did not specify a schema name in the connection, the Secure Agent creates the table in the default schema.

   e. Click **OK**.

8. Specify the advanced properties for the target, if required.

Rules and guidelines for creating a target at runtime

When you configure a mapping with the **Create New at Runtime** option, consider the following rules:

- Do not edit the metadata in the target object. If you edit the metadata, the changes are not reflected at runtime.
- When a target table exists, the Secure Agent uses the same target table.
- When you configure an override of the schema name and table while creating a target at runtime, the Secure Agent creates an empty table in the default database.

JDBC V2 lookups in mappings

In a mapping, you can configure a Lookup transformation to represent a JDBC V2 object.

Specify the name and description of the JDBC V2 lookup.

You can parameterize the JDBC V2 lookup connection and lookup object with values specified in a parameter file when you configure a mapping in advanced mode.

You can create the following lookups when you configure field mappings in a mapping task:

- Connected lookup

- Unconnected lookup

Note: You can't configure a Lookup transformation in a data transfer task.

For more information about the Lookup transformation, see *Transformations*.

The following table describes the JDBC V2 lookup properties that you can configure in a Lookup transformation:

Property	Description
Connection Name	Name of the lookup connection.
Source Type	Source type. Select **Single Object** or **Parameter**.
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the lookup object or click **New Parameter** to define a new parameter for the lookup object. The **Parameter** property appears only if you select parameter as the lookup type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.
Lookup Object	Name of the lookup object for the mapping.
Multiple Matches	Behavior when the lookup condition returns multiple matches. Select from the following options: - Return any row - Return all rows - Report error
Filter	Not applicable.
Sort	Not applicable.

The following table describes the JDBC V2 lookup advanced properties that you can configure in a Lookup transformation:

Advanced Source Property	Description
Pre SQL	The SQL query that the Secure Agent runs before reading data from the source.
Post SQL[1]	The SQL query that the Secure Agent runs after reading data from the source.
Fetch Size	The number of rows that the Secure Agent fetches from the database in a single call.
Table Name	Overrides the table name used in the metadata import with the table name that you specify.
Schema Name	Overrides the schema name of the source object. If you specify the schema name both in the connection and the source properties, the Secure Agent uses the schema name specified in the source properties.

Advanced Source Property	Description
SQL Override	The SQL statement to override the default query and the object name that is used to read data from the JDBC V2 source.

[1] Doesn't apply to mappings in advanced mode.

Mappings in advanced mode example

You work for a multi-national hospital that maintains medical records of patient emergencies for more than 50 specialties in their database across the globe. The hospital has more than 10,000 units spread across 500 locations.

Medical investigators from the hospital want to analyze the medical data of the treatment performance of specific inpatient hospital events within the hospitals. The management wants to use this analysis to provide better care for patients and help achieve health equity.

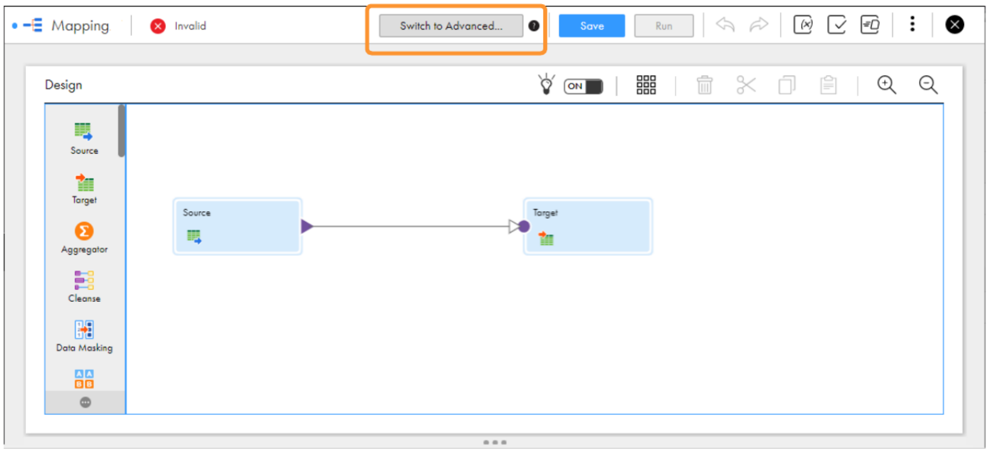
To avoid low performance and high-cost challenges, the hospital plans to port its entire data from all the hospital branches spread globally to Aurora PostgreSQL within a short span of time. Create a mapping in advanced mode to read data from the hospital branches and write data to the Aurora PostgreSQL target.

1. In Data Integration, click **New** > **Mappings** > **Mapping**.

   The **New Mapping** dialog box appears.

2. In the Mapping Designer, click **Switch to Advanced**.

   The following image shows the **Switch to Advanced** button in the Mapping Designer:



3. In the **Switch to Advanced** dialog box, click **Switch to Advanced**.

   The Mapping Designer updates the mapping canvas to display the transformations and functions that are available in advanced mode.

4. Enter a name, location, and description for the mapping.

5. Add a Source transformation, and specify a name and description in the general properties.

6. On the **Source** tab, perform the following steps to provide the source details to read data from the source:
    a. In the **Connection** field, select the required source connection.
    b. In the **Source Type** field, select the type of the source.
    c. In the **Object** field, select the required object.
    d. In the **Advanced Properties** section, provide the appropriate values.
7. Add a Target transformation, and specify a name and description in the general properties.
8. On the **Target** tab, perform the following steps to provide the target details to write data to the Amazon S3 target:
    a. In the **Connection** field, select the JDBC V2 target connection.
    b. In the **Target Type** field, select the type of the target.
    c. In the **Object** field, select the required object.
    d. In the **Operation** field, select the required operation.
    e. In the **Advanced Properties** section, provide appropriate values for the advanced target properties.
9. Click **Save** > **Run** the mapping.

    In Monitor, you can monitor the status of the logs after you run the mapping.

Partitioning for mappings in advanced mode

When you configure a JDBC V2 mapping in advanced mode to read from or write data to a database that supports the Type 4 JDBC driver, you can configure partitioning for the JDBC V2 source or target to optimize the mapping in advanced mode performance at run time. The partition type controls how the advanced cluster distributes data among partitions at partition points.

The advanced cluster distributes rows of data based on the partition key you define. Before you configure a partition, ensure that you select the partition key in the **Fields** tab of the Source or Target transformation in a JDBC V2 mapping. If the imported table already has a primary key defined, the cluster application considers that as the default partition key. Before you import, ensure that the table has only one partition key.

After you import, you can change the partition key on the **Fields** tab. From the **Options** list, select **Edit Metadata** and then select the partition key. Ensure that you do not define more than one partition key for the source or target.

The default number of partitions is 1. If the number of partitions you specified for the target is 1, the cluster application considers the same number of partitions you specified for the source as the target partition number.

JDBC V2 data types supported for partitioning

The following table lists the JDBC V2 data types supported as partition keys:

Data types	Supported
Smallint	Yes
Integer	Yes

Data types	Supported
Bigint	Yes
Decimal	Yes
Numeric	Yes
Real	Yes
Double	Yes
Smallserial	Yes
Serial	Yes
Bigserial	Yes
Char	-
Charn	-
Varchar	-
Varcharn	-
Text	-
Bytes	-
Boolean	-
Date	-
Time	-
Timestamp	-
Bit	-

Rules and guidelines for JDBC V2 objects in mappings

Consider the following rules and guidelines for JDBC V2 objects used as sources and targets in mappings:

Schema names and table names

- If you change the schema name in the connection, the updated schema name does not reflect in the user interface in the existing mapping object. However, the updated schema reflects at runtime.
- In a mapping that uses the **Create Target** option, you can't override the schema name and table name.

- The mapping fails when the target table name contains double quotes.

Import objects

When you import a JDBC V2 object and search for special characters, the search results do not display the objects.

Multiple objects

- You can't parameterize multiple JDBC V2 source objects.
- When you join multiple Azure SQL Database tables, the schemas must have different table names.

Data types

- When you preview the data for time(4), time(5) and time(6) data types, the data is truncated beyond precision 3.
- Do not specify a filter in a mapping for the Azure SQL Database source that contains the Datetime data type.
- When you read or write data that contains time(4), time(5), and time(6) data types, the data is truncated beyond precision 3.
- When you write data that contains the Timestamp data type to a PostgreSQL or Azure SQL Database target using the **Create New at Runtime** option, the Timestamp data type value is appended with the time zone.
- When you create a new Azure SQL or PostgreSQL Database target at runtime and the source data contains the Time data type, the Secure Agent writes the date time value until microseconds.

Partitioning

You can't configure partitioning in a mapping.

Salesforce Data Cloud read

- When you read data from Salesforce Data Cloud and the source contains Date or Datetime data type, the mapping stops responding. To resolve the issue, change the Date or Datetime datatype fields to String in the source object and in the corresponding field mapping.

SQL query

- Ensure that the list of selected columns, the data types, and the order of the columns that appear in the query matches the columns, data types, and order in which they appear in the source object.
- You can't use an SQL query to read data from a Salesforce data lake.

Oracle synonym

You can't read from or write to a synonym of an Oracle synonym.

Rules and guidelines for JDBC V2 objects in mappings configured in advanced mode

Consider the following rules and guidelines for JDBC V2 objects for mappings configured in advanced mode:

Unicode and special characters

- When you import multiple source objects, ensure that the table and column names do not contain Unicode characters.

- Ensure that the source or target table names and field names do not contain special characters.
- When you import a JDBC V2 object and search for special characters, the search results do not display the objects.
- The mapping fails when the target table name contains double quotes.

Schema names and table names

- If you change the schema name in the connection, the updated schema name does not reflect in the user interface in the existing mapping object. However, the updated schema reflects at runtime.
- When you join multiple Azure SQL Database tables, the schemas must have different table names.

Data types

- When you preview the data for time(4), time(5) and time(6) data types, the data is truncated beyond precision 3.
- Do not specify a filter in a mapping for the Azure SQL Database source that contains the Datetime data type.
- When you read or write data that contains time(4), time(5), and time(6) data types, the data is truncated beyond precision 3.
- When you write data that contains the Timestamp data type to a PostgreSQL or Azure SQL Database target using the **Create New at Runtime** option, the Timestamp data type value is appended with the time zone.
- When you create a new Azure SQL or PostgreSQL Database target at runtime and the source data contains the Time data type, the Secure Agent writes the date time value until microseconds.
- When you read from or write to an Oracle table that contains the Number data type, even though the precision does not exceed the maximum allowed value of 38, the mapping adds additional digits to the data. To avoid issues with the Number data type precision, select the **Apply Custom Schema** advanced property in the Source or Target transformations, based on the operation that you configure.

Parameterization

- When you configure the field mapping and the source or target object is parameterized, set the field mapping to **Automap**. If you set the field mapping to **Manual**, the mapping does not run in advance mode.
- You can't parameterize multiple JDBC V2 source objects.

Target operations

- The number of fields in the source and target must be identical when you configure the update mode in the Target transformation.
- To perform an update, upsert, delete, or data driven operation on the target, you must specify a column in the **Update Columns** field even though you define a primary key in the target table.
- When you specify a column in the **Update Columns** field, the Secure Agent inserts rows that contain duplicate data and also updates or upserts data in the target table when both of the following conditions are true:

  - The source table contains columns that have duplicate data.
  - The target table does not have a primary key.
- You can't use multiple PostgreSQL sources in a mapping when you perform an update, upsert, delete, or data driven operation on the target.
- You can't perform an update, upsert, delete, or data driven operation on Oracle synonyms.

- To perform a data driven operation on the target, ensure that the column selected in the **Update Columns** field does not contain duplicate or null values.
- When you run a mapping with an Upsert or Data Driven target operation, the number of processed and failed rows for the target operation is not displayed on the **Individual Task Results** page.

Object metadata

- To retain the same metadata in the target table after you perform an update, upsert, delete, or data driven operation on the target, select the **Truncate Target** option in the Target transformation.

Partitioning

When you configure partitioning for the source or target object that contains an Oracle synonym and the synonym contains a primary key, select the partition key in the **Fields** tab of the Source or Target transformation based on the operation that you configure.

Oracle synonym

You can't read from or write to a synonym of an Oracle synonym.

APPENDIX A

JDBC V2 data type reference

Data Integration uses the JDBC type 4 driver to read data. The Secure Agent converts the JDBC data type to the transformation data type, and uses the transformation data type to move data across platforms.

When Data Integration writes to a JDBC V2 target, the Secure Agent converts the transformation data type to the corresponding JDBC V2 data type.

The following table compares the JDBC V2 data types that Data Integration supports to the transformation data types:

Aurora PostgreSQL Data Type	Transformation Data Type	Description
integer	integer	-2147483648 to 2147483647, precision 10, scale 0
smallint	integer	-32768 to 32767, precision 10, scale 0
bigint	bigint	-9223372036854775808 to 9223372036854775807, precision 19, scale 0
real	double	Precision 15
float	double	Precision 15
double_precision	double	Precision 15
decimal	decimal	Precision 1 to 38, scale 0 to 38
numeric	decimal	Precision 1 to 38, scale 0 to 38
boolean	string	Precision 6
char	string	1 to 10485760 characters
varchar	string	1 to 10485760 characters
date	date/time	January 1, 0001, through December 31, 9999
time	date/time	00:00:00.000 through 23:59:59.999

Aurora PostgreSQL Data Type	Transformation Data Type	Description
timestamp	date/time	Date range: January 1, 0001, through December 31, 9999 time range: 00:00:00 through 23:59:59.997
nchar	string	1 to 10485760 characters

JDBC V2 and transformation data types for Create New at Runtime option

The following table lists the transformation data types that Data Integration supports and the corresponding Aurora PostgreSQL data types when you use the **Create New at Runtime** option:

Transformation Data Type	PostgreSQL Data Type
double	float
varbinary	bytea
binary	bytea
tinyint	integer
time	timestamp
smallint	integer
numeric	numeric
real	float
float	float
char	varchar
nchar	varchar
varchar	varchar
nvarchar	varchar
timestamp	timestamp
boolean	varchar
decimal	numeric

Transformation Data Type	PostgreSQL Data Type
bigint	integer
integer	integer

The following table lists the transformation data types that Data Integration supports and the corresponding Azure SQL Database data types when you use the **Create New at Runtime** option:

Transformation Data Type	Azure SQL Database Data Type
date	datetime2(7)
binary	varbinary
tinyint	int
time	datetime2(7)
smallint	int
numeric	decimal
char	nvarchar
nchar	nvarchar
varchar	nvarchar
nvarchar	nvarchar
real	float
float	float
timestamp	datetime2(7)
boolean	varchar
decimal	decimal
bigint	bigint
integer	int

Rules and guidelines for data types

Consider the following rules when you import data types:

- When you import data from Aurora PostgreSQL that contains the varchar data type and the data size is less than 100 MB, the original precision of the Varchar data type is retained. However, if the data size is more than 100 MB, the Secure Agent imports the Varchar data type with a default precision of 4000.
- Consider the following rules when you use the **Create New at Runtime** option to write to an Azure SQL Database target:
  - When the precision of the char(n), nchar(n), varchar(n), and nvarchar(n) data types is more than 4000, the Secure Agent writes these data types as ntext. If the precision is less than 4000, the Secure Agent writes these data types as nvarchar.
  - When the precision of the binary(n) and varbinary(n)) data types is more than 8000, the Secure Agent writes these data types as images. If the precision is less than 8000, the Secure Agent writes these data types as varbinary.

INDEX