



Informatica® Data Integration - Free & PayGo

Databricks Delta Connector

© Copyright Informatica LLC 2018, 2023

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, 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.

See patents at <https://www.informatica.com/legal/patents.html>.

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: 2023-04-04

Table of Contents

Preface	5
Informatica Resources.	5
Informatica Documentation.	5
Informatica Intelligent Cloud Services web site.	5
Informatica Intelligent Cloud Services Communities.	5
Informatica Intelligent Cloud Services Marketplace.	5
Data Integration connector documentation.	6
Informatica Knowledge Base.	6
Informatica Intelligent Cloud Services Trust Center.	6
Informatica Global Customer Support.	6
 Chapter 1: Introduction to Databricks Delta Connector.....	 7
Databricks Delta Connector assets.	8
Databricks compute resources.	8
Prepare to use the SQL endpoint.	8
Configure Spark parameters for the SQL endpoint.	8
Configure environment variables for the SQL endpoint.	9
Prepare to use the Databricks cluster.	9
Configure Spark parameters for Databricks cluster.	9
Configure Secure Agent properties for Databricks cluster.	10
 Chapter 2: Connections for Databricks Delta.....	 12
Databricks Delta connection properties.	12
AWS cluster properties.	15
Azure cluster properties.	16
 Chapter 3: Mappings and mapping tasks with Databricks Delta connector....	 17
Databricks Delta sources in mappings.	17
Databricks Delta target in mappings.	19
Create a target table at runtime.	21
Rules and guidelines for create target at runtime.	22
Parameterization.	22
Dynamic schema handling.	23
Rules and guidelines for mappings.	23
 Chapter 4: Databricks Delta pushdown optimization.....	 25
Pushdown optimization types.	25
Pushdown optimization preview.	26
Pushdown optimization using a Databricks Delta connection.	26
Read from and write to Databricks Delta.	26

Read from Amazon S3 and write to Databricks Delta.	27
Read from Microsoft Azure Data Lake Storage Gen2 and write to Databricks Delta.	27
Configuring pushdown optimization for a Databricks Delta mapping task.	27
Pushdown compatibility.	28
Transformations with Databricks Delta.	29
Features.	31
Configuring a custom query for the Databricks Delta source object.	34
Pushdown optimization for multiple targets.	35
Single commit for pushdown optimization.	35
Rules and guidelines for pushdown optimization	36
Chapter 5: Data type reference.	39
Databricks Delta and transformation data types.	39
Index.	41

Preface

Use *Databricks Delta Connector* to learn how to read from or write to Databricks Delta by using Data Integration. Learn to create a connection and develop mappings, 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, go to <https://status.informatica.com/> and click **SUBSCRIBE TO UPDATES**. You can then 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 Customer Support Center by telephone or online.

For online support, click **Submit Support Request** in Informatica Intelligent Cloud Services. You can also use Online Support to log a case. Online Support requires a login. You can request a login at <https://network.informatica.com/welcome>.

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 Databricks Delta Connector

You can use Databricks Delta Connector to securely read data from or write data to Databricks Delta.

You can create a Databricks Delta connection and use the connection in mappings and mapping tasks. You can use Databricks Delta Connector only on the Linux operating system.

The following section explains how the Secure Agent communicates with Databricks Delta during the design time and runtime:

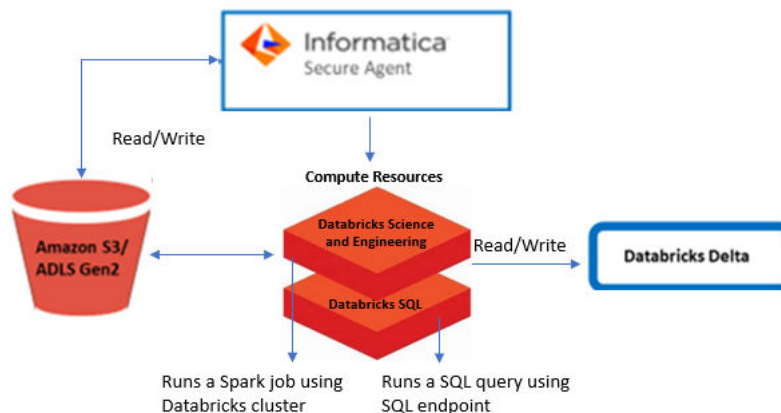
Design time data flow for mappings

During the mapping design, the Secure Agent communicates with the Databricks SQL endpoint or Databricks analytics cluster for metadata-related operations.

Runtime data flow for mappings

During the runtime, the Secure Agent communicates with the Databricks SQL endpoint or Databricks data engineering cluster to read or write data.

The following image shows how the Secure Agent connects to Databricks Delta to read or write data in Data Integration mappings:



The Secure Agent uses Amazon S3 in AWS environment or Azure Data Lake Storage Gen2 in Azure environment for staging the data.

If you use the Databricks SQL endpoint, the Secure Agent starts the Databricks SQL endpoint and then connects to the SQL endpoint to read data from or write data to Databricks Delta tables. When you use Databricks Delta as a source, the Secure Agent runs a SQL query on the Databricks SQL endpoint to read data from a Databricks Delta table. When you use Databricks Delta as a target, the Secure Agent runs a SQL query on the Databricks SQL endpoint to read data from the staging location and write to a Databricks Delta table.

If you use the Databricks cluster, the Secure Agent creates a Databricks data engineering cluster to read data from or write data to Databricks Delta tables. When you use Databricks Delta as a source, the Secure Agent runs a spark job in the Databricks data engineering cluster to read data from a Databricks Delta table and write to the staging location. When you use Databricks Delta as a target, the Secure Agent runs a spark job in the Databricks data engineering cluster to read data from the staging location and write to a Databricks Delta table.

Databricks Delta Connector assets

Create assets in Data Integration to integrate data using Databricks Delta Connector.

When you use Databricks Delta Connector, you can include the following Data Integration assets:

- Data transfer task
- Mapping
- Mapping task

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

Databricks compute resources

When you configure a mapping that reads from or writes to Databricks Delta, the Secure Agent, by default, connects to the Databricks SQL endpoint to run the SQL query and process the mapping.

However, you can enable the Secure Agent properties to connect to the Databricks cluster instead of the SQL endpoint to process the mapping. The Secure Agent runs a Spark job in the Databricks data engineering cluster to read data from or write data to Databricks Delta tables.

Prepare to use the SQL endpoint

You can configure Spark parameters and environment variables for the Databricks SQL endpoint.

You must complete the following prerequisites before you use the SQL endpoint to connect to Databricks Delta:

- Configure Spark parameters for SQL endpoint.
- Configure environment variables for SQL endpoint.

Configure Spark parameters for the SQL endpoint

Before you use the Databricks SQL endpoint to run mappings, ensure to configure the Spark parameters for the SQL endpoint on the Databricks SQL Admin console.

On the Databricks SQL Admin console, navigate to **SQL Warehouse Settings > Data Security**, and configure the Spark parameters for AWS or Azure under **Data access configuration**.

Configuration on AWS

Add the following Spark configuration parameters and restart the SQL endpoint:

- `spark.hadoop.fs.s3a.access.key` <S3 Access Key value>
- `spark.hadoop.fs.s3a.secret.key` <S3 Secret Key value>
- `spark.hadoop.fs.s3a.endpoint` <S3 Staging Bucket endpoint value>

For example, the S3 staging bucket endpoint value is `s3.ap-south-1.amazonaws.com`

Ensure that the access and secret key configured has access to the S3 buckets where you store the data for Databricks Delta tables.

Configuration on Azure

Add the following Spark configuration parameters and restart the SQL endpoint:

- `spark.hadoop.fs.azure.account.oauth2.client.id.<storage-account-name>.dfs.core.windows.net` <value>
- `spark.hadoop.fs.azure.account.auth.type.<storage-account-name>.dfs.core.windows.net` OAuth
- `spark.hadoop.fs.azure.account.oauth2.client.secret.<storage-account-name>.dfs.core.windows.net` <Value>
- `spark.hadoop.fs.azure.account.oauth.provider.type.<storage-account-name>.dfs.core.windows.net` `org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider`
- `spark.hadoop.fs.azure.account.oauth2.client.endpoint.<storage-account-name>.dfs.core.windows.net` `https://login.microsoftonline.com/<Tenant ID>/oauth2/token`

Ensure that the client ID and client secret configured has access to the file systems where you store the data for Databricks Delta tables.

Configure environment variables for the SQL endpoint

Set the following environment variables in the Secure Agent before you connect to the Databricks SQL endpoint:

- `export LANGUAGE="en_US.UTF-8"`
- `export LC_ALL="en_US.UTF-8"`

After you set the environmental variables, you must restart the Secure Agent.

Prepare to use the Databricks cluster

You can configure Spark parameters and Secure Agent properties for the Databricks cluster.

If you want to connect to the Databricks clusters to process the mapping, you must complete the following prerequisites:

- Configure Spark parameters for Databricks cluster.
- Enable Secure Agent properties for Databricks cluster.

Configure Spark parameters for Databricks cluster

Before you connect to the Databricks cluster, you must configure the Spark parameters on AWS and Azure.

Configuration on AWS

Add the following Spark configuration parameters for the Databricks cluster and restart the cluster:

- `spark.hadoop.fs.s3a.access.key <value>`
- `spark.hadoop.fs.s3a.secret.key <value>`
- `spark.hadoop.fs.s3a.endpoint <value>`

Ensure that the access and secret key configured has access to the buckets where you store the data for Databricks Delta tables.

Configuration on Azure

Add the following Spark configuration parameters for the Databricks cluster and restart the cluster:

- `fs.azure.account.oauth2.client.id.<storage-account-name>.dfs.core.windows.net <value>`
- `fs.azure.account.auth.type.<storage-account-name>.dfs.core.windows.net <value>`
- `fs.azure.account.oauth2.client.secret.<storage-account-name>.dfs.core.windows.net <Value>`
- `fs.azure.account.oauth.provider.type.<storage-account-name>.dfs.core.windows.net org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider`
- `fs.azure.account.oauth2.client.endpoint.<storage-account-name>.dfs.core.windows.net https://login.microsoftonline.com/<Tenant ID>/oauth2/token`

Ensure that the client ID and client secret configured has access to the file systems where you store the data for Databricks Delta tables.

Configure Secure Agent properties for Databricks cluster

When you configure mappings, the Databricks SQL endpoint processes the mapping by default.

However, to connect to Databricks analytics or Databricks data engineering clusters, you must enable the following Secure Agent properties for design time and runtime:

- **Design time.** To import metadata, set JRE_OPTS to `-DUseDatabricksSql=false` for the Tomcat JRE type in the Secure Agent properties.
- **Runtime.** To run mappings, set JVMOption to `-DUseDatabricksSql=false` for the DTM type in the Secure Agent properties. To run mappings enabled with pushdown optimization, set JVMOption to `-DUseDatabricksSqlForPdo=false` for the DTM type in the Secure Agent properties.

Setting the property for runtime processing

Before you can run mappings, perform the following steps:

1. In **Administrator**, select the Secure Agent listed on the **Runtime Environments** tab.
2. Click **Edit**.
3. In the **System Configuration Details** section, select Data Integration Server as the **Service** and DTM as the **Type**.
4. Edit the **JVMOption** field.
 - a. To run mappings, set the value to `-DUseDatabricksSql=false`.

DTM	JVMOption2	<code>-DUseDatabricksSql=false</code>
-----	------------	---------------------------------------

- b. To run mappings enabled with pushdown optimization, set the value to `-DUseDatabricksSqlForPdo=false`.

DTM	JVMOption3	<code>'-DUseDatabricksSqlForPdo=false'</code>
-----	------------	---

Setting the property for design time processing

Before you can import metadata and design mappings, perform the following steps:

1. In **Administrator**, select the Secure Agent listed on the **Runtime Environments** tab.
2. Click **Edit**.
3. In the **System Configuration Details** section, select Data Integration Server as the **Service** and Tomcat JRE as the **Type**.
4. Edit the **JRE_OPTS** field and set the value to `-DUseDatabricksSql=false`.

Tomcat JRE	JRE_OPTS	<code>'-Xrs -DUseDatabricksSql=false'</code>
------------	----------	--

CHAPTER 2

Connections for Databricks Delta

Create a Databricks Delta connection to connect to Databricks Delta and read data from or write data to Databricks Delta. You can use Databricks Delta connections to specify sources or targets in mappings and mapping tasks.

In Administrator, create a Databricks Delta connection on the **Connections** page and associate it with a Data Integration task. Define the source properties to read from Databricks Delta or define the target properties to write to Databricks Delta tables.

Databricks Delta connection properties

When you set up a Databricks Delta connection, configure the connection properties.

The following table describes the Databricks Delta 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	The Databricks Delta connection type.
Runtime Environment	Name of the runtime environment where you want to run the tasks. Specify a Secure Agent or a Hosted Agent.
Databricks Host	The host name of the endpoint the Databricks account belongs to. Use the following syntax: <code>jdbc:spark://<Databricks Host>:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/<Org Id>/<Cluster ID>;AuthMech=3;UID=token;PWD=<personal-access-token></code> Note: You can get the URL from the Advanced Options of JDBC or ODBC in the Databricks Delta analytics cluster or all purpose cluster. The value of PWD in Databricks Host, Organization Id, and Cluster ID is always <personal-access-token>.

Property	Description
Cluster ID	<p>The ID of the Databricks analytics cluster.</p> <p>You can get the cluster ID from the JDBC URL.</p> <p>Use the following syntax:</p> <pre>jdbc:spark://<Databricks Host>:443/ default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/<Org Id>/ <Cluster ID>;AuthMech=3;UID=token;PWD=<personal-access-token></pre>
Organization Id	<p>The unique organization ID for the workspace in Databricks.</p> <p>Use the following syntax:</p> <pre>jdbc:spark://<Databricks Host>:443/ default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/<Organization Id>/<Cluster ID>;AuthMech=3;UID=token;PWD=<personal-access-token></pre>
Databricks Token	<p>Personal access token to access Databricks.</p> <p>Ensure that you have permissions to attach to the cluster identified in the Cluster ID property.</p> <p>For mappings, you must have additional permissions to create data engineering clusters.</p>
SQL Endpoint JDBC URL	<p>Databricks SQL endpoint JDBC connection URL.</p> <p>Use the following syntax:</p> <pre>jdbc:spark://<Databricks Host>:443/ default;transportMode=http;ssl=1;AuthMech=3;httpPath=/sql/1.0/endpoints/ <SQL endpoint cluster ID>;</pre> <p>For Data Integration, this field is required to connect to the Databricks SQL endpoint. Ensure that you set the required environment variables in the Secure Agent.</p> <p>Note: The Databricks Host, Organization ID, and Cluster ID properties are not considered if you configure the SQL Endpoint JDBC URL property.</p> <p>For more information on Databricks Delta SQL endpoint, contact Informatica Global Customer Support.</p>
Database	<p>The database in Databricks Delta that you want to connect to.</p> <p>For Data Integration, by default, all databases available in the workspace are listed.</p>
JDBC Driver Class Name	<p>The name of the JDBC driver class.</p> <p>Specify the driver class name as <code>com.simba.spark.jdbc.Driver</code>.</p>
Cluster Environment	<p>The cloud provider where the Databricks cluster is deployed.</p> <p>Choose from the following options:</p> <ul style="list-style-type: none"> - AWS - Azure <p>Default is AWS.</p> <p>The connection attributes depend on the cluster environment you select. For more information, see the AWS cluster properties and Azure cluster properties sections.</p>
Min Workers	<p>The minimum number of worker nodes to be used for the Spark job.</p> <p>Mandatory for mappings and minimum value is 1.</p>
Max Workers	<p>The maximum number of worker nodes to be used for the Spark job.</p> <p>If you don't want to autoscale, set Max Workers = Min Workers or don't set Max Workers.</p>

Property	Description
DB Runtime Version	<p>The Databricks runtime version.</p> <p>Determines the version of Databricks cluster to spawn when you connect to Databricks cluster to process mappings.</p> <p>Choose from the following options:</p> <ul style="list-style-type: none"> - 7.3 LTS - 9.1 LTS <p>Default is 7.3 LTS.</p>
Worker Node Type	<p>The worker node instance type that is used to run the Spark job.</p> <p>For example, the worker node type for AWS can be i3.2xlarge. The worker node type for Azure can be Standard_DS3_v2.</p>
Driver Node Type	<p>The driver node instance type that is used to collect data from the Spark workers.</p> <p>For example, the driver node type for AWS can be i3.2xlarge. The driver node type for Azure can be Standard_DS3_v2.</p> <p>If you don't specify the driver node type, Databricks uses the value you specify in the worker node type field.</p>
Instance Pool ID	<p>The instance pool ID used for the Spark cluster.</p> <p>If you specify the Instance Pool ID to run mappings, the following connection properties are ignored:</p> <ul style="list-style-type: none"> - Driver Node Type - EBS Volume Count - EBS Volume Type - EBS Volume Size - Enable Elastic Disk - Worker Node Type - Zone ID
Enable Elastic Disk	<p>Enables the cluster to get additional disk space.</p> <p>Enable this option if the Spark workers are running low on disk space.</p>
Spark Configuration	<p>The Spark configuration to use in the Databricks cluster.</p> <p>The configuration must be in the following format:</p> <pre>"key1"="value1"; "key2"="value2"; ...</pre> <p>For example:</p> <pre>"spark.executor.userClassPathFirst"="False"</pre>
Spark Environment Variables	<p>The environment variables to export before launching the Spark driver and workers.</p> <p>The variables must be in the following format:</p> <pre>"key1"="value1"; "key2"="value2"; ...</pre> <p>For example:</p> <pre>"MY_ENVIRONMENT_VARIABLE"="true"</pre>

The following properties are required to launch the job cluster at run time for a mapping task:

- Min Workers
- Max Workers
- DB Runtime Version
- Worker Node Type
- Driver Node Type

- Enable Elastic Disk
- Spark Configuration
- Spark Environment Variables
- Zone ID
- EBS Volume Type
- EBS Volume Count
- EBS Volume Size

AWS cluster properties

When you set up a Databricks Delta connection, configure the connection properties based on the cluster environment you select.

The following table describes the Databricks Delta connection properties that apply when you select the AWS cluster environment:

Property	Description
S3 Authentication Mode	The authentication mode to access Amazon S3. Default is Permanent IAM credentials.
S3 Access Key	The key to access the Amazon S3 bucket.
S3 Secret Key	The secret key to access the Amazon S3 bucket.
S3 Data Bucket	The existing bucket to store the Databricks Delta data.
S3 Staging Bucket	The existing bucket to store staging files.
S3 Service Regional Endpoint	The S3 regional endpoint when the S3 data bucket and the S3 staging bucket need to be accessed through a region-specific S3 regional endpoint. Default is <code>s3.amazonaws.com</code> .
Zone ID	The zone ID for the Databricks job cluster. Applies only if you want to create a Databricks job cluster in a particular zone at runtime. For example, <code>us-west-2a</code> . Note: The zone must be in the same region where your Databricks account resides.
EBS Volume Type	The type of EBS volumes launched with the cluster.
EBS Volume Count	The number of EBS volumes launched for each instance. You can choose up to 10 volumes. Note: In a Databricks Delta connection, specify at least one EBS volume for node types with no instance store. Otherwise, cluster creation fails.
EBS Volume Size	The size of a single EBS volume in GiB launched for an instance.

Azure cluster properties

When you set up a Databricks Delta connection, configure the connection properties based on the cluster environment that you select.

The following table describes the Databricks Delta connection properties that apply when you select the Azure cluster environment:

Property	Description
ADLS Storage Account Name	The name of the Microsoft Azure Data Lake Storage account.
ADLS Client ID	The ID of your application to complete the OAuth Authentication in the Active Directory.
ADLS Client Secret	The client secret key to complete the OAuth Authentication in the Active Directory.
ADLS Tenant ID	The ID of the Microsoft Azure Data Lake Storage directory that you use to write data.
ADLS Endpoint	The OAuth 2.0 token endpoint from where authentication based on the client ID and client secret is completed.
ADLS Data Filesystem Name	The name of an existing file system to store the Databricks Delta data.
ADLS Staging Filesystem Name	The name of an existing file system to store the staging data.

CHAPTER 3

Mappings and mapping tasks with Databricks Delta connector

Use a mapping to define data flow logic, such as specific ordering of logic or joining sources from different systems. Use the Data Integration Mapping Designer to configure mappings.

When you configure a mapping to describe the flow of data from source and target, you can also add transformations to transform data. A transformation includes field rules to define incoming fields. Links visually represent how data moves through the data flow.

After you create a mapping, you can run the mapping or you can deploy the mapping in a mapping task. The mapping task allows you to process data based on the data flow logic defined in a mapping.

You can configure parameters in a mapping and add the mapping to a mapping task. You can use the same mapping in multiple mapping tasks and define the parameters for each mapping task.

When you create a task, you can associate the task with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can also configure advanced session properties. You can monitor tasks that are currently running in the activity monitor and view details about completed tasks in the activity log.

Databricks Delta sources in mappings

In a mapping, you can configure a Source transformation to represent a Databricks Delta object.

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

Property	Description
Connection	Name of the source connection. Select a source connection or click New Parameter to define a new parameter for the source connection.
Source Type	Type of the source object. Select any of the following source objects: <ul style="list-style-type: none">- Single Object- Multiple Objects. You can use implicit joins and advanced relationships with multiple objects.- Query. Applies only for mappings enabled with full pushdown optimization.- Parameter. Select Parameter to define the source type when you configure the task.

Property	Description
Object	Name of the source object.
Query	<p>Click on Define Query and enter a valid custom query.</p> <p>The Query property appears only if you select Query as the source type.</p> <p>You can parameterize a custom query object at runtime in a mapping.</p> <p>Applicable only when you select full pushdown optimization for a mapping that reads from Databricks Delta.</p>

The following table describes the Databricks Delta query options that you can configure in a Source transformation:

Property	Description
Query Options	<p>Filters the source data based on the conditions you specify. Click Configure to configure a filter option.</p> <p>The Filter option filters records and reduces the number of rows that the Secure Agent reads from the source. Add conditions in a read operation to filter records from the source. You can specify the following filter conditions:</p> <ul style="list-style-type: none"> - Not parameterized. Use a basic filter to specify the object, field, operator, and value to select specific records. - Completely parameterized. Use a parameter to specify the filter query. - Advanced. Use an advanced filter to define a complex filter condition. <p>Note: You can use Contains, Ends With, and Starts With operators to filter records only on SQL endpoints.</p>

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

Property	Description
Database Name	Overrides the database name provided in connection and the database name provided during metadata import.
Table Name	Overrides the table name used in the metadata import with the table name that you specify.
Staging Location	<p>Relative directory path to store the staging files.</p> <ul style="list-style-type: none"> - If the Databricks cluster is deployed on AWS, use the path relative to the Amazon S3 staging bucket. - If the Databricks cluster is deployed on Azure, use the path relative to the Azure Data Lake Store Gen2 staging filesystem name.
Job Timeout	<p>Maximum time in seconds that is taken by the Spark job to complete processing. If the job is not completed within the time specified, the Databricks cluster terminates the job and the mapping fails.</p> <p>If the job timeout is not specified, the mapping shows success or failure based on the job completion.</p>

Property	Description
Job Status Poll Interval	Poll interval in seconds at which the Secure Agent checks the status of the job completion. Default is 30 seconds.
DB REST API Timeout	The Maximum time in seconds for which the Secure Agent retries the REST API calls to Databricks when there is an error due to network connection or if the REST endpoint returns 5xx HTTP error code. Default is 10 minutes.
DB REST API Retry Interval	The time Interval in seconds at which the Secure Agent must retry the REST API call, when there is an error due to network connection or when the REST endpoint returns 5xx HTTP error code. This value does not apply to the Job status REST API. Use job status poll interval value for the Job status REST API. Default is 30 seconds.
Tracing Level	Sets the amount of detail that appears in the log file. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.

Databricks Delta target in mappings

In a mapping, you can configure a Target transformation to represent a Databricks Delta object.

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

Property	Description
Connection	Name of the target connection. Select a target connection or click New Parameter to define a new parameter for the target connection.
Target Type	Target type. Select one of the following types: <ul style="list-style-type: none"> - Single Object. - Parameter. Select Parameter to define the target type when you configure the task.
Object	Name of the target object.
Create Target	Creates a target. Enter a name for the target object and select the source fields that you want to use. By default, all source fields are used. You can select an existing target object or create a new target object at runtime. You cannot parameterize the target at runtime.

Property	Description
Operation	<p>Defines the type of operation to be performed on the target table.</p> <p>Select from the following list of operations:</p> <ul style="list-style-type: none"> - Insert (Default) - Update - Upsert - Delete - Data Driven <p>When you use an upsert operation, you must configure the Update Mode in target details as Update else Insert.</p> <p>If the key column gets null value from the source, the following actions take place for different operations:</p> <ul style="list-style-type: none"> - Update. Skips the operation and does not update the row. - Delete. Skips the operation and does not delete the row. - Upsert. Inserts a new row instead of updating the existing row.
Update Columns	<p>The fields to use as temporary primary key columns when you update, upsert, or delete data on the Databricks Delta target tables. When you select more than one update column, the mapping task uses the AND operator with the update columns to identify matching rows.</p> <p>Applies to update, upsert, delete and data driven operations.</p>
Data Driven Condition	<p>Flags rows for an insert, update, delete, or reject operation based on the expressions that you define.</p> <p>For example, the following IIF statement flags a row for reject if the ID field is null. Otherwise, it flags the row for update:</p> <pre>IIF (ISNULL(ID), DD_REJECT, DD_UPDATE)</pre> <p>Required if you select the data driven operation.</p>

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

Advanced Property	Description
Target Database Name	Overrides the database name provided in the connection and the database selected in the metadata browser for existing targets.
Target Table Name	Overrides the table name at runtime for existing targets.
Write Disposition	<p>Overwrites or adds data to the existing data in a table. You can select from the following options:</p> <ul style="list-style-type: none"> - Append. Appends data to the existing data in the table even if the table is empty. - Truncate. Overwrites the existing data in the table.
Staging Location	<p>Relative directory path to store the staging files.</p> <ul style="list-style-type: none"> - If the Databricks cluster is deployed on AWS, use the path relative to the Amazon S3 staging bucket. - If the Databricks cluster is deployed on Azure, use the path relative to the Azure Data Lake Store Gen2 staging filesystem name.

Advanced Property	Description
Job Timeout	Maximum time in seconds that is taken by the Spark job to complete processing. If the job is not completed within the time specified, the Databricks cluster terminates the job and the mapping fails. If the job timeout is not specified, the mapping shows success or failure based on the job completion.
Job Status Poll Interval	Poll interval in seconds at which the Secure Agent checks the status of the job completion. Default is 30 seconds.
DB REST API Timeout	The Maximum time in seconds for which the Secure Agent retries the REST API calls to Databricks when there is an error due to network connection or if the REST endpoint returns 5xx HTTP error code. Default is 10 minutes.
DB REST API Retry Interval	The time Interval in seconds at which the Secure Agent must retry the REST API call, when there is an error due to network connection or when the REST endpoint returns 5xx HTTP error code. This value does not apply to the Job status REST API. Use job status poll interval value for the Job status REST API. Default is 30 seconds.
Update Mode	Defines how rows are updated in the target tables. Select from the following options: <ul style="list-style-type: none"> - Update As Update: Rows matching the selected update columns are updated in the target. - Update Else Insert: Rows matching the selected update columns are updated in the target. Rows that don't match are appended to the target.

Create a target table at runtime

You can use an existing target or create a target to hold the results of a mapping. If you choose to create the target, the agent creates the target if it does not exist already when you run the task.

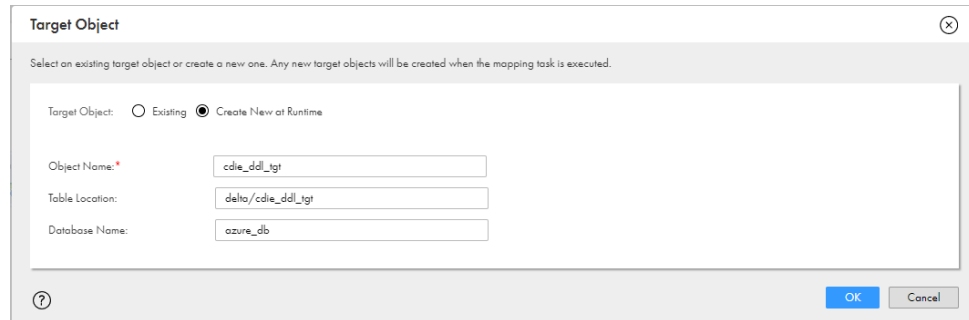
You can create both managed and external tables.

To specify the target properties, perform the following tasks:

1. Select the Target transformation in the mapping.
2. To specify the target, click the **Target** tab.
3. Select the target connection.
4. For the target type, choose **Single Object** or **Parameter**.
5. Specify the target object or parameter.

6. 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.



- 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 in the **Object Name** field. Specify the table name in lowercase.
- d. Enter the location of the target table data in the **Table Location** field.
The table location is relative to the data bucket or data filesystem name specified in the connection. External table is created if you specify the table location.
- e. In the **Database Name** field, specify the Databricks database name.
The database name that you specify in the connection properties takes precedence.
- f. Click **OK**.

Rules and guidelines for create target at runtime

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

- When a source object consists of Date data type and you use the default create target option in a mapping, the date data gets corrupted. To resolve this issue, navigate to **Edit Metadata** option in the **Target fields** of the target and change **Native Type** to Date.
- When you enable dynamic schema handling in a task and create target at runtime, you must provide the complete path of the target table in the Database Name. Ensure that the table name is in lowercase. For example, database_name/TABLE.

Parameterization

You can parameterize the following properties when you create mappings:

- Source properties. Source type, source connection, query options in source, database name, table name, and advanced properties in the source.
- Target properties. Target type, target connection, target database name, target table name, and target advanced properties.

Dynamic schema handling

You can choose how Data Integration handles changes that you make to the data object schemas. To refresh the schema every time the mapping task runs, you can enable dynamic schema handling in the task.

Configure schema change handling on the **Schedule** page when you configure the task.

The following table describes the schema change handling options:

Option	Description
Asynchronous	Default. Data Integration refreshes the schema when you edit the mapping or mapping task, and when Informatica Intelligent Cloud Services is upgraded.
Dynamic	Data Integration refreshes the schema every time the task runs. You can choose from the following options to refresh the schema: <ul style="list-style-type: none">- Alter and apply changes. Data Integration applies the following changes from the source schema to the target schema:<ul style="list-style-type: none">- New fields. Alters the target schema and adds the new fields from the source.- Don't apply DDL changes. Data Integration does not apply the schema changes to the target.- Drop current and recreate. Drops the existing target table and then recreates the target table at runtime using all the incoming metadata fields from the source.

For more information, see the "Schema change handling" topic in *Tasks* in the Data Integration help.

Rules and guidelines for mappings

Consider the following rules and guidelines for Databricks Delta objects used as sources and targets in mappings:

- When you specify SESSSTARTTIME variable in a query in a mapping task to return the Datetime values, specify the query in the following format:
`:select to_timestamp('$$$SESSSTARTTIME', 'MM/dd/yyyy HH:mm:ss.SSSSSS') as t;`
- When you run multiple concurrent mappings to write data to Databricks Delta targets, a transaction commit conflict error might occur and the mappings might fail.
- View objects are displayed in the Table panel instead of the View panel while importing a Databricks Delta object. This issue occurs when the Databricks cluster is deployed on AWS cloud service.
- To avoid java heap space error when you read or write complex files, set the JVM options for type DTM to increase the -Xms and -Xmx values in the system configuration details of the Secure Agent. The recommended values for -Xms is 512 MB and -Xmx is 1024 MB.
- When you import views, the Select Source Object dialog box does not display view objects.
- When you test the Databricks Delta connection, the Secure Agent does not validate the values you specify in the Org ID connection parameter.
- You cannot use the Hosted Agent as a runtime environment when you configure a mapping to run on the SQL endpoint to read or write data that contains unicode characters.
- The number of clusters that the Secure Agent creates to run the mapping depends on the number of Databricks Delta connections used in the transformations in a mapping. For example, if multiple transformations use the same Databricks Delta connection, the mapping runs on a single cluster.

- When you keep the mapping designer idle for more than 15 minutes, the metadata fetch throws an exception.
- If you change the database name in the connection and run the existing mappings, the mappings start failing. After you change the database name in the connection, you must reimport the objects in the existing mappings before you run the mappings.
- Use the following formats to run the mapping successfully, when you import a Databricks Delta source object containing Date or Boolean data types with a simple source filter conditions:
 - Boolean = 0 or 1
 - Date = YYYY-MM-DD HH24:MM:SS.US
- When you run a mapping with source column data type as string containing TRUE / FALSE value and write data to target with Boolean data type column of a Databricks Delta table, the Secure Agent writes data as 0 to the target.
- When the Databricks analytics cluster is down and you perform a test connection or import an object, the connection is timed out after 10 minutes.
- When you parameterize the source or target connection in a mapping and you do not specify the database name, ensure that you specify the database name in lowercase when you assign a default value for the parameter.
- When you parameterize the source filter condition or any expressions in a mapping, ensure that you specify the table name in lowercase when you add the source filter condition or the expression in the mapping task. Otherwise, the Secure Agent throws the following exception:


```
Invalid expression string for filter condition
```
- When you run a mapping to write data to a Databricks Delta target using create target at runtime and the target table already exists, ensure that the target table schema is same. Otherwise, the mapping fails.
- When you run a mapping to write data to multiple Databricks Delta targets that use the same Databricks Delta connection and the Secure Agent fails to write data to one of targets, the mapping fails and the Secure Agent does not write data to the remaining targets.
- When you use the `Create New at Runtime` option to create a Databricks target, you can parameterize only the target connection and the table name using a parameter file. You cannot parameterize other properties such as `Path` or `DBname`.

CHAPTER 4

Databricks Delta pushdown optimization

When you run a task configured for pushdown optimization, the task converts the transformation logic to an SQL query. The task sends the query to the database, and the database executes the query.

The amount of transformation logic that you can push to the database depends on the database, transformation logic, and task configuration. The Secure Agent processes all transformation logic that it cannot push to the database.

Pushdown optimization types

When you apply pushdown optimization, the task pushes transformation logic to the source or target database based on the optimization type you specify in the task properties. Data Integration translates the transformation logic into SQL queries or Databricks Delta commands to the Databricks Delta database. The database runs the SQL queries or Databricks Delta commands to process the transformations.

You can configure the following types of pushdown optimization in a mapping:

None

The task does not push down the transformation logic to the Databricks Delta database.

Full

The task pushes as much of the transformation logic as possible to process in the Databricks Delta target database.

Data Integration analyses all the transformations from the source to the target. If all the transformations are compatible in the target, it pushes the entire mapping logic to the target. If it cannot push the entire mapping logic to the target, Data Integration first pushes as much transformation logic to the source database and then pushes as much transformation logic as possible to the target database.

When a transformation is not supported in the mapping, the task partially pushes down the mapping logic to the point where the transformation is supported for pushdown optimization. However, this applies only to Databricks SQL endpoints.

When you enable full pushdown optimization, you can determine how Data Integration handles the job when pushdown optimization does not work. You can use the If pushdown mode is not possible, cancel the task option to set the task to fail or run without pushdown optimization.

Full pushdown optimization is enabled by default in mapping tasks.

Source

The task pushes as much as the transformation logic as possible to process in the Databricks Delta source database. This applies only to Databricks SQL endpoints.

Pushdown optimization preview

Before you can run a mapping task configured for pushdown optimization, you can preview if pushdown optimization is possible when you create the mapping. You can preview pushdown optimization from the **Pushdown Optimization** panel in the Mapping Designer.

After you select the required pushdown optimization options and run the preview, Data Integration creates and runs a temporary pushdown preview mapping task. When the job completes, Data Integration displays the SQL queries to be executed and any warnings in the **Pushdown Optimization** panel. The warning messages help you understand which transformations in the configured mapping are not applicable for pushdown optimization. If pushdown optimization fails, Data Integration lists any queries generated up to the point of failure. You can edit the mapping and fix the required transformations before you run the mapping for pushdown optimization.

You can also view the temporary job created under **My Jobs** and download the session log to view the queries generated.

For more information about how to preview pushdown optimization, see the topic "Pushdown optimization preview" in *Mappings* in the Data Integration help.

Pushdown optimization using a Databricks Delta connection

You can configure pushdown optimization for a mapping that contains a Databricks Delta connection. Pushdown optimization enhances the mapping performance. You can configure full or source pushdown when you read data from a Databricks Delta source and write to a Databricks Delta target.

Read from and write to Databricks Delta

You can configure pushdown optimization in a mapping to read from and write to Databricks Delta using a Databricks Delta connection.

Example

You work in a motorbike retail company with more than 30,000 dealerships and 2000 inspection centers globally. The company stores millions of records in Databricks Delta hosted on Azure. You want to use Data Integration to perform some transformations on the data before you write back to Databricks Delta.

Use a Databricks Delta connection in the mapping to read from the Databricks Delta source and write the processed data to the Databricks Delta target. Configure full pushdown optimization in the mapping to enhance the performance.

Read from Amazon S3 and write to Databricks Delta

You can configure pushdown optimization for a mapping that uses an Amazon S3 V2 connection in the Source transformation to read from Amazon S3 and a Databricks Delta connection in the Target transformation to write to Databricks Delta.

Example

You work for a healthcare organization. Your organization offers a suite of services to manage electronic medical records, patient engagement, telephonic health services, and care coordination services. The organization uses infrastructure based on Amazon Web Services and stores its data on Amazon S3. The management plans to load data to a data warehouse to perform healthcare analytics and create data points to improve operational efficiency. To load data from an Amazon S3 based storage object to Databricks Delta, you must use ETL and ELT with the required transformations that support the data warehouse model.

Use an Amazon S3 V2 connection to read data from a file object in an Amazon S3 source and a Databricks Delta connection to write to a Databricks Delta target. Configure full pushdown optimization in the mapping to optimize the performance.

Read from Microsoft Azure Data Lake Storage Gen2 and write to Databricks Delta

You can configure pushdown optimization for a mapping that uses an Microsoft Azure Data Lake Storage Gen2 connection in the Source transformation to read from Microsoft Azure Data Lake Storage Gen2 and a Databricks Delta connection in the Target transformation to write to Databricks Delta.

Example

You want to load data from an Microsoft Azure Data Lake Storage Gen2 based storage object to Databricks Delta for analytical purposes. You want to transform the data before it is made available to users. Use an Microsoft Azure Data Lake Storage Gen2 connection to read data from a Microsoft Azure Data Lake Storage Gen2 source and a Databricks Delta connection to write to a Databricks Delta target. Configure full pushdown optimization in the mapping task to optimize the performance of loading data to Databricks Delta. Pushdown optimization enhances the performance of the task and reduces the cost involved.

Configuring pushdown optimization for a Databricks Delta mapping task

Perform the following steps to configure pushdown optimization for a Databricks Delta mapping task:

1. Create a Databricks Delta connection.
2. Create a mapping to read data from a Databricks Delta source and write data to a Databricks Delta target.
3. Create a mapping task.
 - a. Select the configured mapping.
 - b. In the **Pushdown Optimization** section on the **Schedule** tab, set the pushdown optimization value to **Full** or **To Source**.
 - c. Save the task and click **Finish**.

When you run the mapping task, the transformation logic is pushed to the Databricks Delta database.

Pushdown compatibility

You can configure the task to push transformations, functions, and operators to the database.

When you use pushdown optimization, the Secure Agent converts the expression in the transformation by determining equivalent operators and functions in the database. If there is no equivalent operator and function, the Secure Agent processes the transformation logic.

Functions with Databricks Delta

When you use pushdown optimization, Data Integration converts the expression in the transformation by determining equivalent functions in the database. If there is no equivalent function, Data Integration processes the transformation logic.

The following table summarizes the availability of pushdown functions that you can push to the Databricks Delta database by using full or source pushdown optimization:

Function	Function
ABS()	RAND()
AVG()	REG_EXTRACT()
CEIL()	REG_MATCH()
CHR()	REG_REPLACE
CONCAT()	REVERSE()
COSH()	ROUND(NUMBER)
COUNT()	RPAD()
FIRST()	RTRIM()
GET_DATE_PART()	SQRT()
GREATEST()	STDDEV()
IN()	SUBSTR()
INITCAP()	SUM()
INSTR()	SYSTIMESTAMP()
IS_DATE()	TANH()
IS_NUMBER()	TO_BIGINT
IS_SPACES()	TO_CHAR(DATE)
LAST()	TO_DATE()
LAST_DAY()	TO_DECIMAL()
LN()	TO_FLOAT()

Function	Function
LPAD()	TO_INTEGER()
LTRIM()	TRUNC(DATE)
MAX()	UPPER()
MIN()	VARIANCE()
POWER()	-

Operators with Databricks Delta

When you use pushdown optimization, the Secure Agent converts the expression in the transformation by determining equivalent operators in the database. If there is no equivalent operator, the Secure Agent processes the transformation logic.

The following table lists the pushdown operators that you can push to Databricks Delta:

Operator	Operator
+	=
-	>=
*	<=
/	!=
%	AND
	OR
>	NOT
<	

Variables with Databricks Delta

You can use full pushdown to push the `SESSSTARTTIME` variable to the Databricks Delta database.

Transformations with Databricks Delta

When you configure pushdown optimization, the Secure Agent tries to push the configured transformation to Databricks Delta.

You can use full or source pushdown to push the following transformations to Databricks Delta:

- Aggregator
- Expression
- Filter

- Joiner
- Lookup
- Sorter
- Union
- Router
- Rank
- SQL

Note: Router transformation is not applicable for source pushdown optimization.

Aggregator transformation

You can configure full pushdown optimization to push an Aggregator transformation to process in Databricks Delta.

Aggregate calculations

You can perform the following aggregate calculations:

- AVG
- COUNT
- FIRST
- LAST
- MAX
- MIN
- SUM
- STDDEV
- VARIANCE

Incoming ports

When you configure an Aggregator transformation and the incoming port is not used in an aggregate function or in a group by field, the output is not deterministic as the ANY_VALUE() function returns any value from the port.

You can pass only single arguments to the LAST, STDDEV, and VARIANCE functions.

Lookup transformation

You can configure full pushdown optimization to push a Lookup transformation to process in Databricks Delta. This applies to both connected and unconnected lookups.

You can add the following lookups:

- Cached
- Uncached
- Unconnected with cached

When you configure a connected lookup, select the **Multiple Matches** property value as **Return all rows** in the lookup properties for pushdown optimization to work.

You can nest the unconnected lookup function with other expression functions.

When you configure an unconnected Lookup transformation, consider the following rules:

- You must select the **Multiple Matches** property value as **Report error** in the unconnected lookup properties for pushdown optimization to work.
- You can only configure an Expression transformation for an output received from an unconnected lookup.

SQL Transformation

You can use an SQL transformation to push supported scalar functions to Databricks Delta.

When you configure pushdown optimization for a mapping, you can use scalar functions in a SQL transformation and run queries with the Databricks Delta target endpoint.

You can use only the SELECT clause SQL statement to push down a function. The following snippet demonstrates the syntax of a simple SELECT SQL query:

```
SELECT <function_name1>(~Arg~), <function_name2> (~Arg~)...
```

You can push an SQL transformation with the following restrictions:

- You can configure only an SQL query in the SQL transformation. You cannot enable a stored procedure when you push down to Databricks Delta.
- The SQL query must be a simple SELECT statement without 'FROM' and 'WHERE' arguments. The SQL transformation only supports functions with simple SELECT statement.
- When you specify a SELECT query, you must also specify the column name and number of columns based on the functions. For example, when you specify the query `select square(~AGE~), sqrt(~SNAME~)`, you must specify two output columns for AGE and SNAME functions each, otherwise the mapping fails.
- If any SQL error occurs, the error is added to the `SQLException` field by default. However, when you run a mapping enabled with pushdown optimization, the `SQLException` field remains as Null.
- The `NumRowsAffected` field records the number of rows affected while computing the output buffer. However, for SQL transformation, the `NumRowsAffected` is 0, as the query runs for all the records at the same time.
- You cannot include special characters in the query, as SQL transformation does not support special characters in the arguments.
- You can use an SQL transformation when the SELECT statement is present only in the query property. You cannot configure an SQL transformation with a parameterized query, as dynamic parameter support is limited, and the query fails with a DTM error.

Features

You can configure pushdown optimization for a mapping that reads from the following sources and writes to a Databricks Delta target:

- Databricks Delta source
- Amazon S3 source
- Microsoft Azure Data Lake Storage Gen2 source

When you configure a mapping, some parameters are not supported for a mapping enabled for pushdown optimization. You can refer to the list of parameters that each source supports.

Databricks Delta sources, targets, lookups

You must configure a Databricks Delta connection with simple or hybrid mode when you enable pushdown optimization in a mapping task.

Source properties

When you configure pushdown optimization, the mappings support the following advance properties for a Databricks Delta source:

- Source Object Type
 - Single
 - Multiple
 - Query
 - Parameter

Note: When you use the query source type to read from Databricks Delta, you can choose to retain the field metadata and save the mapping. Even if you edit the query and run the mapping, the field metadata specified at design time is retained.
- Query Options
 - Filter. You can use both simple and advanced filter conditions.
- Database Name
- Table Name

Note:

- Query is applicable only for full pushdown optimization.
- Contains, Ends With, and Starts With filter operators are not applicable when you use source filter to filter records.

Target properties

When you configure pushdown optimization, the mappings support the following properties for an Databricks Delta target:

- Target Object Type
 - Single
 - Parameter
 - Create New at Runtime
- Operation
 - Insert
 - Update
 - Upsert
 - Delete
- Create Target
- Target Database Name
- Target Table Name
- Update Mode
- Write Disposition for Insert operation.

Lookup properties

When you configure pushdown optimization, the mappings support the following advance properties for a Databricks Delta lookup:

- Source Object Type
 - Single
 - Query
 - Parameter
- Database Name
- Table Name

Note: Query is applicable only for full pushdown optimization.

Note: If you configure advanced properties that are not supported, the Secure Agent either ignores the properties or logs an pushdown optimization validation error in the session logs file. If the Secure Agent logs an error in the session log, the mappings run in the Informatica runtime environment without full pushdown.

Supported features for Amazon S3 V2 source

When you configure pushdown optimization, the mappings support the following properties for an Amazon S3 V2 source:

- Source connection parameter
- Source Type - Single, query
- Parameter
- Format - Avro, ORC, Parquet, JSON, and CSV
- Source Type - File and directory. XML source type is not applicable.
- Folder Path
- File Name

When you configure pushdown optimization, the mapping supports the following transformations:

- Filter
- Expression
- Aggregator
- Sorter
- Router
- Joiner
- Lookup
- Union
- Rank

For information about the configurations for the listed options, see the help for the Amazon S3 V2 Connector.

Supported features for Microsoft Azure Data Lake Storage Gen2 source

When you configure pushdown optimization, the Microsoft Azure Data Lake Storage Gen2 connection supports the following properties:

- Account Name
- Client ID
- Client Secret
- Tenant ID
- File System Name
- Directory Path
- Adls Gen2 End-point
- Server-side Encryption

When you configure pushdown optimization, the mappings support the following properties for a Microsoft Azure Data Lake Storage Gen2 source:

- Source connection, connection parameter
- Source Type - Single, parameter
- Format - Avro, Parquet, JSON, ORC, and CSV.
- Intelligent Structure Model
- Formatting Options
- Filesystem Name Override
- Source Type - File, Directory
- Directory Override - Absolute path; Relative path
- File Name Override - Source object
- Allow Wildcard Characters

When you configure pushdown optimization, the mapping supports the following transformations:

- Filter
- Expression
- Aggregator
- Sorter
- Router
- Joiner
- Lookup
- Union
- Rank

For information about the configurations for the listed options, see the help for the Microsoft Azure Data Lake Storage Gen2 Connector.

Configuring a custom query for the Databricks Delta source object

You can push down a custom query to Databricks Delta.

Before you run a task that contains a custom query as the source object, you must set the **Create Temporary View** session property in the mapping task properties.

Note: If you do not set the **Create Temporary View** property, the mapping runs without pushdown optimization.

Perform the following task to set the property:

1. In the mapping task, navigate to the **Pushdown Optimization** section on the **Schedule** tab.
2. Select **Create Temporary View**.
3. Click **Finish**.

Pushdown optimization for multiple targets

When you enable full pushdown for a mapping to write to multiple Databricks Delta targets, you can further optimize the write operation.

To optimize, you can configure an insert, update, upsert, or delete operation for each target.

You can select the same Databricks Delta target table in multiple Target transformations, configure a different operation for each of the Target transformations independent of each other.

Single commit for pushdown optimization

When you enable full pushdown optimization for a mapping to write to multiple Databricks Delta targets, you can configure the mapping to commit the configured operations for all the targets within a connection group together.

You can use single commit to combine the metadata from all the targets and send the metadata for processing in a single execution call. When you use single commit, the Secure Agent segregates the targets into connection groups based on equivalent connection attributes and commits the operations together for each connection group. This optimizes the performance of the write operation.

When you run a mapping with multiple targets, the Databricks Delta connections used for these multiple target transformations that have the same connection attribute values are grouped together to form connection groups. As all the targets in a connection group have the same connection attributes, only a single connection is established for each connection group which represents that particular connection group. The transactions on each connection group runs on a single Databricks cluster.

If the Secure Agent fails to write to any of the targets, the task execution stops and the completed transactions for the targets that belong to the same connection group are not rolled back.

To enable single commit to write to multiple targets, set the **EnableSingleCommit=Yes** custom property in the **Advanced Session Properties** section on the **Schedule** tab of the mapping task.

When you run a mapping with single commit enabled, you can view the row statistics details in the session logs.

Single commit is applicable only when you run a mapping on Databricks data engineering cluster.

Rules and guidelines for pushdown optimization

Use the following rules and guidelines when you enable a mapping for pushdown optimization to a Databricks Delta database:

Mapping with Databricks Delta source and target

Use the following rules and guidelines when you configure pushdown optimization in a mapping that reads from and writes to Databricks Delta:

- LAST function is a non-deterministic function. This function returns different results each time it is called, even when you provide the same input values.
- When you configure a Filter transformation or specify a filter condition, do not specify special characters.
- When you connect to Databricks clusters to process the mapping and define a custom query with multiple tables in the SELECT statement, the mapping displays incorrect data for fields that have the same name. This doesn't apply to Databricks runtime version 9.1 LTA or later.
- When you configure a mapping enabled for full pushdown optimization to read from multiple sources and you override the database name and table name from the advanced properties, the mapping fails.
- To configure a Filter transformation or specify a filter condition on columns of date or timestamp in a Databricks Delta table, you must pass the data through the TO_DATE() function as an expression in the filter condition.
- When you specify custom query as a source object, ensure that the SQL query does not contain any partitioning hints such as COALESCE, REPARTITION, or REPARTITION_BY_RANGE.
- When you configure a mapping enabled for full pushdown optimization on the Databricks Delta SQL engine, you cannot configure single commit to write to multiple targets.
- When you configure a mapping enabled for full pushdown optimization on the Databricks Delta SQL engine and push the data to the Databricks Delta target, ensure that you map all the fields in target. Else, the mapping fails.
- When you create a new target at runtime, you must not specify a database name and table name in the **Target Database Name** and **Target Table Name** in the target advanced properties.
- When you read data from a column of Date data type and write data into a column of Date data type, the pushdown query pushes the column of Date data type and casts the column to Timestamp data type.
- You cannot completely parameterize a multi-line custom query using a parameter file. If you specify a multi-line custom query in a parameter file, the mapping considers only the first line of the multi-line query.
- When you push the GREATEST() function to Databricks Delta and configure input value arguments of String data type, you must not specify the caseFlag argument.
- To push the TO_CHAR(DATE) function to Databricks Delta, use the following string and format arguments:
 - YYYY
 - YY
 - MM
 - MON
 - MONTH
 - DD
 - DDD
 - DY

- DAY
- HH12
- HH24
- MI
- Q
- SS
- SS.MS
- SS.US
- SS.NS
- To push the TO_DATE(string, format) function to Databricks Delta, you must use the following format arguments:
 - YYYY
 - YY
 - MM
 - MON
 - MONTH
 - DD
 - DDD
 - HH12
 - HH24
 - MI
 - SS
 - SS.MS
 - SS.US
 - SS.NS

Mapping with Amazon S3 source and Databricks Delta target

Use the following rules and guidelines when you configure pushdown optimization in a mapping that reads from an Amazon S3 source and writes to a Databricks Delta target:

- When you select the source type as directory in the advanced source properties, ensure that all the files in the directory contain the same schema.
- When you select query as the source type in lookup, you cannot override the database name and table name in the advanced source properties.
- When you include a source transformation in a mapping enabled with pushdown optimization, exclude the FileName field from the source. The FileName field is not applicable.
- When you parameterize a lookup object in a mapping enabled with pushdown optimization, the mapping fails as you cannot exclude the filename port at runtime.
- When you parameterize the source object in a mapping task, ensure that you pass the source object parameter value with the fully qualified path in the parameter file.
- You cannot use wildcard characters for the source file name and directory name in the source transformation.
- You cannot use wildcard characters for the folder path or file name in the advanced source properties.

- When you read from a partition folder that has a transaction log file, select the source type as Directory in the advanced source properties.
- You cannot configure dynamic lookup cache.
- When you use a Joiner transformation in a mapping enabled with pushdown optimization and create a new target at runtime, ensure that the fields do not have a not null constraint.
- Ensure that the field names in Parquet, ORC, AVRO, or JSON files do not contain Unicode characters.

Mapping with Azure Data Lake Storage Gen2 source and Databricks Delta target

Use the following rules and guidelines when you configure pushdown optimization in a mapping that reads from a Azure Data Lake Storage Gen2 source and writes to a Databricks Delta target:

- Mappings fail if the lookup object contains unsupported data types.
- When you select the source type as directory in the advanced source property, ensure that all the files in the directory contain the same schema.
- When you select query as the source type in lookup, you cannot override the database name and table name in the advanced source properties.
- When you include a source transformation in a mapping enabled with pushdown optimization, exclude the FileName field from the source. The FileName field is not applicable.
- When you parameterize a lookup object in a mapping enabled with pushdown optimization, the mapping fails as you cannot exclude the filename port at runtime.
- When you parameterize the source object in a mapping task, ensure that you pass the source object parameter value with the fully qualified path in the parameter file.
- You cannot use wildcard characters for the source file name and directory name in the source transformation.
- When you read from a partition folder that has a transaction log file, select the source type as Directory in the advanced source properties.
- You cannot configure dynamic lookup cache.
- When you use a Joiner transformation in a mapping enabled with pushdown optimization and create a new target at runtime, ensure that the fields do not have a not null constraint.
- Ensure that the field names in Parquet, ORC, AVRO, or JSON files do not contain Unicode characters.

CHAPTER 5

Data type reference

Databricks Delta native data types

Databricks Delta data types appear in the Fields tab for Source and Target transformations when you choose to edit metadata for the fields.

Transformation data types

Set of data types that appear in the remaining transformations. They are internal data types based on ANSI SQL-92 generic data types, which Data Integration uses to move data across platforms. Transformation data types appear in all remaining transformations in Data Integration tasks.

When the Data Integration application reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the Data Integration application writes to a target, it converts the transformation data types to the comparable native data types.

Databricks Delta and transformation data types

The following table compares the Databricks Delta native data type to the transformation data type:

Databricks Delta Data Type	Transformation Data Type	Range and Description
Binary	Binary	1 to 104,857,600 bytes.
Bigint	Bigint	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807. 8-byte signed integer.
Boolean	Integer	1 or 0.
Date	Date/Time	January 1,0001 to December 31,9999.
Decimal	Decimal	For Data Integration mappings: Precision 1 to 28 digits, scale 0 to 28.
Double	Double	Precision 15.
Float	Double	Precision 7.
Int	Integer	-2,147,483,648 to +2,147,483,647.

Databricks Delta Data Type	Transformation Data Type	Range and Description
Smallint	Integer	-32,768 to +32,767.
String	String	1 to 104,857,600 characters.
Tinyint	Integer	-128 to 127
Timestamp	Date/Time	January 1,0001 00:00:00 to December 31,9999 23:59:59.997443. Timestamp values only preserve results up to microsecond precision of six digits. The precision beyond six digits is discarded.

INDEX

C

Cloud Application Integration community
URL [5](#)
Cloud Developer community
URL [5](#)
connections
Databricks Delta [12](#)
Create target
rules and guidelines [22](#)
create target at runtime [21](#)

D

Data Integration community
URL [5](#)
data types [39](#)
Databricks Delta
connection properties [12](#)
pushdown optimization overview [25](#)
pushdown through Databricks Connection [26](#)
Databricks Delta connection
configuration [27](#)
Databricks Delta connections
overview [12](#)
Databricks Delta Connector
assets [8](#)
overview [7](#)
Databricks Delta connector rules and guidelines [23](#)

F

field delimiter [17](#)

I

Informatica Global Customer Support
contact information [6](#)
Informatica Intelligent Cloud Services
web site [5](#)

M

maintenance outages [6](#)
Mapping tasks
overview [17](#)
mappings
overview [17](#)

mappings (*continued*)
source properties [17](#)
target properties [19](#)

N

native data type [39](#)

P

properties
in mappings [17](#), [19](#)
pushdown optimization
functions [28](#), [29](#)
transformations [28](#), [29](#)
Pushdown optimization
preview [26](#)
Pushdown Optimization
Rules and Guidelines for Functions [36](#)
Pushdown optimization preview [26](#)

S

source properties [17](#)
status
Informatica Intelligent Cloud Services [6](#)
system status [6](#)

T

tracing level [17](#)
transformation data type [39](#)
transformations
pushdown optimization [29](#)
trust site
description [6](#)

U

upgrade notifications [6](#)

W

web site [5](#)