



Informatica® Cloud Data Integration

Databricks Delta Connector

© Copyright Informatica LLC 2018, 2022

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

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

Informatica, the Informatica logo, 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: 2022-02-09

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 overview.	7
Databricks Delta Connector task and object types.	8
Administration of Databricks Delta Connector.	8
License requirements.	8
Simba JDBC Driver configuration.	9
Databricks cluster prerequisites.	9
Databricks SQL engine requirements.	9
Chapter 2: Databricks Delta connections	11
Databricks Delta connections overview.	11
Databricks Delta connection properties.	11
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
Rules and guidelines for Databricks Delta objects.	23
Chapter 4: Elastic mappings and mapping tasks with Databricks Delta Connector	24
Elastic mappings overview.	24
Elastic mapping example.	24
Rules and guidelines for elastic mappings.	25
Chapter 5: Databricks Delta pushdown optimization	27
Databricks Delta pushdown optimization overview.	27

Pushdown optimization types.	27
Data Integration behavior with source and full pushdown optimization.	28
Pushdown optimization using a Databricks Delta connection.	28
Read from and write to Databricks Delta.	28
Configuring pushdown optimization for a Databricks Delta mapping task.	28
Pushdown compatibility.	29
Configuring a custom query for the Databricks Delta source object.	34
Pushdown optimization for multiple targets.	34
Single commit for pushdown optimization.	34
Rules and guidelines for pushdown optimization	35
Chapter 6: Data type reference.	37
Databricks Delta data type reference overview.	37
Databricks Delta and transformation data types.	37
Index.	39

Preface

Use *Databricks Delta Connector* to learn how to read from or write to Databricks Delta by using Cloud Data Integration. Learn to create a connection and develop mappings, elastic mappings, mapping tasks, dynamic mapping tasks, and data transfer tasks in Cloud 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

This chapter includes the following topics:

- [Databricks Delta Connector overview, 7](#)
- [Databricks Delta Connector task and object types, 8](#)
- [Administration of Databricks Delta Connector, 8](#)

Databricks Delta Connector overview

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

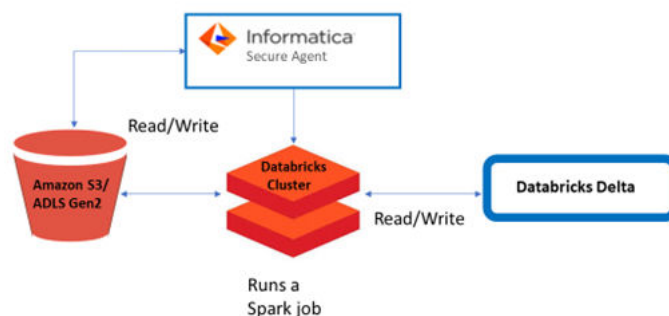
You can create a Databricks Delta connection and use the connection in mappings, elastic mappings, and mapping tasks. You can use Databricks Delta Connector only on the Linux operating system. The following section explains how does the Secure Agent communicate with Databricks Delta during the design time and the run time:

Design-time data flow for mappings and elastic mappings

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

Run-time data flow for mappings

The following image shows how the Secure Agent connects to Databricks Delta to read or write data in Cloud 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. The Secure Agent creates Databricks data engineering cluster to read data from or write data to Databricks Delta tables. When Databricks Delta connector is used 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 Databricks Delta connector is used 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.

Run-time data flow for elastic mappings

Elastic mappings make use of the elastic cluster to run a spark job and process data.

Databricks Delta Connector task and object types

The following table lists the Databricks Delta transformation types that you can include in Data Integration tasks:

Task Type	Source	Target
Mapping	Yes	Yes
Elastic mapping	Yes	Yes
Mapping task	Yes	Yes
Dynamic mapping task	Yes	Yes
Data transfer task	Yes	Yes

For more information about configuring dynamic mapping tasks and data transfer tasks, see *Tasks* in the Data Integration Help.

Administration of Databricks Delta Connector

License requirements

Ensure that the organization has the following licenses before you use Databricks Delta Connector:

- Cloudera 6.1
- UDTforHierarchy package
- saas-xmetadataaread package
- DataTransformation package

Simba JDBC Driver configuration

1. Download the `SimbaSparkJDBC4-2.6.16.1020` driver on the Secure Agent machine from the Databricks website.
2. Add the Simba driver at the following location in the Secure Agent machine:
`<AGENT_HOME>/ext/connectors/thirdparty/informatica.databricksdelta`
3. Restart the Secure Agent.

Databricks cluster prerequisites

Before you use the Databricks Delta Connector, ensure to meet the following prerequisites:

Databricks cluster 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 here must have access to the buckets where the data for Databricks Delta tables are stored.

Databricks cluster 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 here must have access to the file systems where the data for Databricks Delta tables are stored.

Databricks SQL engine requirements

You must enable separate properties for the Secure Agent to connect to the Databricks Delta SQL endpoint during the design time and runtime.

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

Setting the property for design time processing

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

1. In **Administrator**, select the Secure Agent listed on the **Runtime Environments** tab.
2. Click **Edit**.

- In the **System Configuration Details** section, select Data Integration Server as the **Service** and Tomcat JRE as the **Type**.
- Edit the **JRE_OPTS** field to add the following value:
`-DUseDatabricksSql=true`



Setting the property for runtime processing

Before you can run mappings enabled for pushdown optimization on the SQL endpoint, perform the following steps:

- In **Administrator**, select the Secure Agent listed on the **Runtime Environments** tab.
- Click **Edit**.
- In the **System Configuration Details** section, select Data Integration Server as the **Service** and DTM as the **Type**.
- Edit the **JVMOption** field to add the following value:
`-DUseDatabricksSqlForPdo=true`

DTM	JVMClassPath	'pmserversdk.jar'	<input type="checkbox"/>
DTM	JVMOption1	-.Xmx3072m	<input type="checkbox"/>
DTM	JVMOption2	'-DUseDatabricksSqlForPdo=true'	<input type="checkbox"/>

CHAPTER 2

Databricks Delta connections

This chapter includes the following topics:

- [Databricks Delta connections overview, 11](#)
- [Databricks Delta connection properties, 11](#)

Databricks Delta connections overview

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, elastic 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 create a Databricks Delta connection, you must configure the connection properties, such as the connection name, type, and runtime environment.

The following table describes the Databricks Delta connection properties:

Property	Description
Connection Name	Required. The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
Description	Description of the connection. The description cannot exceed 4,000 characters.
Type	Required. Select Databricks Delta.

Property	Description
Runtime Environment	Required. Name of the runtime environment that contains the Secure Agent to connect to Databricks Delta.
Databricks Host	<p>Required. The host name of the endpoint the Databricks account belongs to.</p> <p>Example:</p> <p>if the JDBC URL for the cluster is: <code>example_\\jdbc:spark://dbc-40d21191-cf1b.cloud.databricks.com:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/5275227018426481/1015-063325-1oge501;AuthMech=3;UID=token;PWD=<personal-access-token></code></p> <p>The host name is <code>dbc-40d21191-cf1b.cloud.databricks.com</code></p> <p>Note: You can get the URL from the Databricks Delta analytics cluster or all purpose cluster -> Advanced Options -> JDBC / ODBC.</p>
Org Id	<p>Required. The unique organization ID for the workspace in Databricks.</p> <p>For example, if the JDBC URL for the cluster is: <code>example_\\jdbc:spark://dbc-40d21191-cf1b.cloud.databricks.com:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/5275227018426481/1015-063325-1oge501;AuthMech=3;UID=token;PWD=<personal-access-token></code></p> <p>The organization ID is <code>5275227018426481</code>.</p>
Cluster ID	<p>Required. The ID of the Databricks analytics cluster. You can obtain the cluster ID from the JDBC URL.</p> <p>For example, if the JDBC URL for your cluster is: <code>example_\\jdbc:spark://dbc-40d21191-cf1b.cloud.databricks.com:443/default;transportMode=http;ssl=1;httpPath=sql/protocolv1/o/5275227018426481/1015-063325-1oge501;AuthMech=3;UID=token;PWD=<personal-access-token></code></p> <p>The cluster ID is <code>1015-063325-1oge501</code>.</p>
Databricks Token	<p>Required. Personal access token to access Databricks.</p> <p>You must 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>Example for the Databricks SQL endpoint JDBC URL:</p> <pre>jdbc:spark://dbc-40d21191-cf1b.cloud.databricks.com:443/default;transportMode=http;ssl=1;AuthMech=3;httpPath=/sql/1.0/endpoints/1234574fs2i7c734;</pre> <p>This field is required only when you use the Databricks Delta SQL engine to fetch metadata and run mappings enabled with full pushdown optimization. Ensure that you have set the <code>JRE_OPTS</code> property in Secure Agent properties.</p> <p>Note: The Databricks Host, Org ID, and Cluster ID properties are not considered if you configure the SQL Endpoint JDBC URL property.</p> <p>For more information on how to run mappings on Databricks Delta SQL Engine, 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>Required. The name of the JDBC driver class.</p> <p>You must specify the driver class name as <code>com.simba.spark.jdbc.Driver</code> for mappings and elastic mappings.</p>

Property	Description
Cluster Environment	<p>The cloud provider where the Databricks cluster is deployed.</p> <p>You can select from the following options:</p> <ul style="list-style-type: none"> - AWS - Azure <p>Default is AWS.</p> <p>The connection attributes differ depending on the cluster environment you select. For more information, see the AWS and Azure cluster environment 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> <p>Not applicable to elastic mappings.</p>
Max Workers	<p>The maximum number of worker nodes to be used for the Spark job.</p> <p>If you do not want autoscale, set Max Workers = Min Workers or do not set Max Workers.</p> <p>Not applicable to elastic mappings.</p>
DB Runtime Version	<p>The Databricks runtime version.</p> <p>Select 7.3 LTS from the list.</p> <p>Not applicable to elastic mappings.</p>
Worker Node Type	<p><i>Required.</i> The instance type of the machine used for the Spark worker node.</p> <p>Not applicable to elastic mappings.</p>
Driver Node Type	<p>The instance type of the machine used for the Spark driver node. If not provided, the value as in worker node type is used.</p> <p>Not applicable to elastic mappings.</p>
Instance Pool ID	<p>The instance pool 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 <p>Not applicable to elastic mappings.</p>
Enable Elastic Disk	<p>Enable this option for the cluster to dynamically acquire additional disk space when the Spark workers are running low on disk space.</p> <p>Not applicable to elastic mappings.</p>

Property	Description
Spark Configuration	<p>The Spark configuration to be used in the Databricks cluster. 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> <p>Not applicable to elastic mappings.</p>
Spark Environment Variables	<p>The environment variables that you need to export before launching the Spark driver and workers. 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> <p>Not applicable to elastic mappings.</p>

AWS cluster environment

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

Property	Description
S3 Authentication Mode	Required. The authentication mode to access Amazon S3. Default is Permanent IAM credentials.
S3 Access Key	Required. The key to access the Amazon S3 bucket.
S3 Secret Key	Required. The secret key to access the Amazon S3 bucket.
S3 Data Bucket	Required. The existing bucket where the Databricks Delta data is stored.
S3 Staging Bucket	Required. The existing bucket to store staging files. Not applicable to elastic mappings.
S3 Service Regional Endpoint	Use 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> to access the data bucket and staging bucket.
Zone ID	Required only if, at runtime, you want to create a Databricks job cluster in a particular zone. For example, <code>us-west-2a</code> . Note: The zone must be in the same region where your Databricks account resides. Not applicable to elastic mappings.
EBS Volume Type	The type of EBS volumes launched with the cluster. Not applicable to elastic mappings.

Property	Description
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, you must specify, at least one EBS volume for node types with no instance store, otherwise, cluster creation fails. Not applicable to elastic mappings.
EBS Volume Size	The size of a single EBS volume in GiB launched for an instance. Not applicable to elastic mappings.

Azure cluster environment

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

Property	Description
ADLS Storage Account Name	Required. The name of the Microsoft Azure Data Lake Storage account.
ADLS Client ID	Required. The ID of your application to complete the OAuth Authentication in the Active Directory.
ADLS Client Secret	Required. The client secret key to complete the OAuth Authentication in the Active Directory.
ADLS Tenant ID	Required. The ID of the Microsoft Azure Data Lake Storage directory that you use to read data or write data.
ADLS Endpoint	Required. The OAuth 2.0 token endpoint from where authentication based on the client ID and client secret is completed.
ADLS Data Filesystem Name	Required. The name of an existing file system where the Databricks Delta data is stored.
ADLS Staging Filesystem Name	Required. The name of an existing file system where the staging data is stored. Not applicable to elastic mappings.

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

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 or a dynamic mapping task. You can use the same mapping in multiple mapping tasks and define the parameters for each mapping task. You can use a dynamic mapping task to create and batch multiple jobs based on the same mapping. When you configure the dynamic mapping task, you can select the value and scope of each parameter for each job.

For more information about dynamic mapping tasks, see *Tasks* in the Data Integration help.

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. Parameterization is not applicable to elastic mappings.
Source Type	Type of the source object. Select any of the following source objects: <ul style="list-style-type: none"> - Single Object - Query. Applies only for mappings enabled with full pushdown optimization. - Parameter. Select Parameter to define the source type when you configure the task. Not applicable to elastic mappings.
Object	Name of the source object.
Query	Click on Define Query and enter a valid custom query. The Query property appears only if you select Query as the source type. You can parameterize a custom query object at runtime in a mapping.

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

Property	Description
Query Options	Filters the source data based on the conditions you specify. Click Configure to configure a filter option: <p>Filter</p> <p>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. Not applicable to elastic mappings. - Advanced. Use an advanced filter to define a complex filter condition.

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

Note: Advanced source properties are not applicable to elastic mappings.

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.

Property	Description
Staging Location	Relative directory path to store the staging files. <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	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.
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.

Property	Description
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 cannot parameterize the target at runtime.
Operation	Defines the type of operation to be performed on the target table. Select from the following list of operations: <ul style="list-style-type: none"> - Insert (Default) - Update - Upsert - Delete - Data Driven When you use an upsert operation, you must configure the Update Mode in target details as Update else Insert. If the key column gets null value from the source, the following actions take place for different operations: <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	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. Applies to update, upsert, delete and data driven operations.
Data Driven Condition	Flags rows for an insert, update, delete, or reject operation based on the expressions that you define. For example, the following IIF statement flags a row for reject if the ID field is null. Otherwise, it flags the row for update: <code>IIF (ISNULL(ID), DD_REJECT, DD_UPDATE)</code> Required if you select the data driven operation.

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. Not applicable to elastic mappings.
Target Table Name	Overrides the table name at runtime for existing targets. Not applicable to elastic mappings.
Write Disposition	Overwrites or adds data to the existing data in a table. You can select from the following options: <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.

Advanced Property	Description
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. <p>Not applicable to elastic mappings.</p>
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> <p>Not applicable to elastic mappings.</p>
Job Status Poll Interval	<p>Poll interval in seconds at which the Secure Agent checks the status of the job completion.</p> <p>Default is 30 seconds.</p> <p>Not applicable to elastic mappings.</p>
DB REST API Timeout	<p>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.</p> <p>Default is 10 minutes.</p> <p>Not applicable to elastic mappings.</p>
DB REST API Retry Interval	<p>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.</p> <p>This value does not apply to the Job status REST API. Use job status poll interval value for the Job status REST API.</p> <p>Default is 30 seconds.</p> <p>Not applicable to elastic mappings.</p>
Update Mode	<p>Defines how rows are updated in the target tables. Select from the following options:</p> <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. <p>Not applicable to elastic mappings.</p>

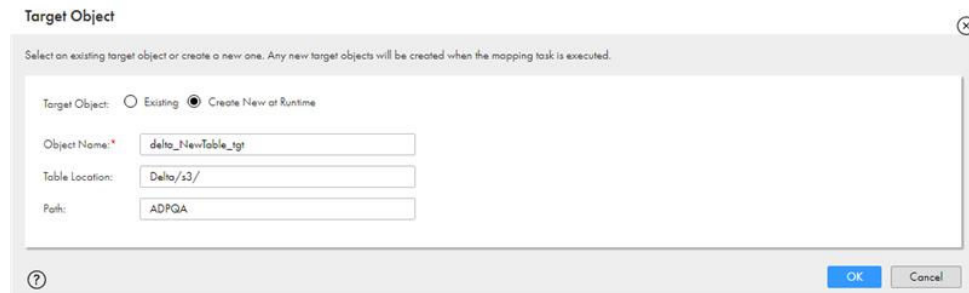
Create a target table at runtime

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.

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 and specify the object name.



You must specify the target object 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 in the **Object Name** field.
- 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.
- e. In the **Path** field, specify the Databricks database name.
- 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 run a mapping to write data to a Databricks Delta target and use the create target option at runtime, ensure that table data location is empty.
- When you run an elastic mapping to write data to a Databricks Delta target and use the create target option at runtime, you must provide the table location and path. Otherwise, the mapping fails.

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.

You can parameterize the following properties when you create elastic mappings:

- Source properties. Source type, source connection, query options in source, database name, and table name in the source.
- Target properties. Target type, target connection, target database name, and target table name.
- **Note:** Parameterization using parameter files is not applicable for elastic mappings.

Rules and guidelines for Databricks Delta objects

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

- 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.
- You must map all fields from the source to the target in a mapping and the mapping must contain all fields present in the target table. If the fields are unmapped, the mapping fails.
- 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 the 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 the targets, the mapping fails and the Secure Agent does not write data to the remaining targets.

CHAPTER 4

Elastic mappings and mapping tasks with Databricks Delta Connector

This chapter includes the following topics:

- [Elastic mappings overview, 24](#)
- [Elastic mapping example, 24](#)
- [Rules and guidelines for elastic mappings, 25](#)

Elastic mappings overview

When you use Data Integration Elastic, you can read and write data to Databricks Delta from Cloud Data Integration.

Use the Mapping Designer in Data Integration to define and configure an elastic mapping for Databricks Delta sources and targets. Add the Source or Target transformations in the mapping canvas and configure the Databricks Delta source or target properties.

Add the elastic mapping to a mapping task. When you run an elastic job, the Secure Agent starts the elastic cluster and pushes the job to the cluster for processing. You can use Monitor to monitor both the cluster and job. As you run additional jobs, the cluster is scaled up or scaled down accordingly.

Elastic mapping example

You work for a retail company that offers more than 50,000 products and the stores are distributed across the globe. The company ingests a large amount of customer engagement details from the transactional CRM system into Amazon S3.

The sales team wants to improve customer engagement and satisfaction at every touch point. To create a seamless customer experience and deliver personalized service across the various outlets, the retail company plans to load the data that is stored in the Amazon S3 bucket to Databricks Delta.

You can create an elastic mapping that runs on the elastic cluster to achieve faster performance when you read data from the Amazon S3 bucket and write data to the Databricks Delta target.

In the elastic mapping, you can choose to add transformations to process the raw data that you read from the Amazon S3 bucket and then write the curated data to Databricks Delta.

The following example illustrates how to create a simple elastic mapping to read from an Amazon S3 source and write to Databricks Delta target:

1. In Data Integration, click **New > Mappings > Elastic Mapping > Create**.
The **New Mapping** dialog box appears.
2. Enter a name, location, and description for the mapping.
3. Add a Source transformation, and specify a name and description in the general properties.
4. On the **Source** tab, perform the following steps to read data from the Amazon S3 source:
 - a. In the **Connection** field, select the Amazon S3 V2 connection.
 - b. In the **Source Type** field, select single object as the source type.
 - c. In the **Object** field, select the parquet file object that contains the customer details.
 - d. In the **Advanced Properties** section, specify the required parameters.
5. On the **Expression** tab, define an expression to change the file name part of the customer parquet file to uppercase based on your business requirement before you write data to the Databricks Delta target:
6. Add a Target transformation, and specify a name and description in the general properties.
7. On the **Target** tab, specify the details to write data to Databricks Delta:
 - a. In the **Connection** field, select the Databricks Delta target connection.
 - b. In the **Target Type** field, select single object.
 - c. In the **Object** field, select the Databricks Delta object to which you want to write the curated customer engagement data.
 - d. In the **Operation** field, select the insert operation.
 - e. In the **Advanced Properties** section, specify the required advanced target properties.
8. Click **Save > Run** to validate the mapping.
In Monitor, you can monitor the status of the logs after you run the task.

Rules and guidelines for elastic mappings

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

- When you do a data type conversion from Float to Double or use create target at run time, data loss is encountered.
- When you run an elastic mapping with a source column data type as String containing true or false value and the target data type as Boolean, the Secure Agent writes data as null to the target.
- Use the following formats when you import a Databricks Delta source object containing Boolean, Date, or Timestamp data types with a simple source filter conditions:
 - Boolean = 0 or 1
 - Date = YYYY-MM-DD HH24:MM:SS.US
 - Timestamp = YYYY-MM-DD HH24:MM:SS.US
- You cannot import a Databricks Delta table from Databricks Filesystem (DBFS) location when you run an elastic mapping.

- You cannot use the following features:
 - Data preview
 - View
 - Multipipe
- After you create and run a mapping configuration task, it is recommended to shut down the all purpose cluster. If you modify a mapping task or edit the connection linked to a mapping task, metadata is fetched again and the all purpose cluster restarts.
- When you do a data type conversion from Date or Timestamp to String, the Secure Agent writes the value only in the following default format for both Date and Timestamp:


```
MM/DD/YYYY HH24:MI:SS
```
- When you do a data type conversion from String to Date or Timestamp, the String value must be in the following format:


```
MM/DD/YYYY HH24:MI:SS
```

To use any other format, you must specify the format in the advanced session property of a mapping task for successful conversion. Null is populated in the target for the unmatched format.
- When you do a data type conversion from Bigint to Double, the target data is written in the exponential format.
- When you perform an update, upsert, or a data driven operation with an IIF condition that includes DD_DELETE or DD_UPDATE, ensure that the update column that you specified does not have duplicate rows. Otherwise, the mapping fails with the following error:


```
java.lang.UnsupportedOperationException: Cannot perform MERGE as multiple source rows
matched and attempted to update the same target row in the Delta table.
```
- When you perform an insert, update, upsert operation, or DD_UPDATE and the range of the data in source column is greater than the range of the target column, the mapping does not fail and leads to data truncation.
- When you specify a single constant in the data driven condition, the mapping ignores the data driven condition and the Secure Agent performs insert, update, or delete operation based on the constant. For example, if you specify the data driven condition as DD_INSERT, the mapping does not consider the update columns and depends on the Write Disposition property.
- When you specify a single constant with the IIF condition in the data driven condition such as IIF(COL_INT > 20 , DD_UPDATE), the Secure Agent inserts the data into the target even for those rows that do not satisfy the condition.
- When you specify the DD_REJECT constant in the data driven condition, the Secure Agent does not log the rejected rows in the error file or the session log.

CHAPTER 5

Databricks Delta pushdown optimization

This chapter includes the following topics:

- [Databricks Delta pushdown optimization overview, 27](#)
- [Pushdown optimization types, 27](#)
- [Pushdown optimization using a Databricks Delta connection, 28](#)

Databricks Delta pushdown optimization overview

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 behavior with source and full pushdown optimization

When you select full pushdown optimization for a mapping that reads from or writes to Databricks Delta, Data Integration analyzes the mapping from the source to the target or until it reaches a downstream transformation to determine whether to push down the entire mapping logic for processing to the database.

If all the transformations in the mapping are applicable for pushdown optimization, the task pushes down the entire mapping logic to the database.

When a transformation is not applicable for pushdown optimization in the mapping, the Secure Agent 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.

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 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.

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**.

- 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.

Supported functions for Databricks Delta mappings

The following table summarizes the availability of pushdown functions in an Databricks Delta database. Columns marked with an X indicate that the function can be pushed to the Databricks Delta database by using full pushdown optimization. Columns marked with a dash (-) symbol indicate that the function cannot be pushed to the database.

Function	Pushdown	Function	Pushdown	Function	Pushdown
ABORT()	-	INDEXOF()	-	REPLACECHR()	-
ABS()	X	INITCAP()	X	REPLACESTR()	-
ADD_TO_DATE()	-	INSTR()	X	RESPEC()	-
AES_DECRYPT()	-	IS_DATE()	-	REVERSE()	X
AES_ENCRYPT()	-	IS_NUMBER()	-	ROUND(DATE)	-
ASCII()	-	IS_SPACES()	-	ROUND(NUMBER)	X
AVG()	X	ISNULL()	-	RPAD()	X
CEIL()	X	LAG()	-	RTRIM()	X
CHOOSE()	-	LAST()	X	SET_DATE_PART()	-
CHR()	X	LAST_DAY()	X	SIGN()	-
CHRCODE()	-	LEAST()		SIN()	-
COMPRESS()	-	LENGTH()	-	SINH()	-
CONCAT()	X	LN()	X	SOUNDEX()	-
CONCAT_ARRAY()	-	LOG()	-		
CONVERT_BASE()	-	LOOKUP	-	SQRT()	X
COS()	-	LOWER()	-	STDDEV()	X
COSH()	X	LPAD()	X	STRUCT()	-

Function	Pushdown	Function	Pushdown	Function	Pushdown
COUNT()	X	LTRIM()	X	STRUCT_AS()	-
CRC32()	-		X	SUBSTR()	X
DATE_COMPARE()	-	MAKE_DATE_TIME()	-	SUM()	X
DATE_DIFF()	X	MAX()	X	SYSTIMESTAMP()	X
DECODE()	-	MD5()	-	TAN()	-
DECODE_BASE64()	-	MEDIAN()	-	TANH()	X
DECOMPRESS()	-	METAPHONE()	-	TO_BIGINT	X
ENCODE_BASE64()	-	MIN()	X	TO_CHAR(DATE)	X
ERROR()	-	MOD()	-	TO_CHAR(NUMBER)	-
EXP()	-	MOVINGAVG()	-	TO_DATE()	X
FIRST()	X	MOVINGSUM()	-	TO_DECIMAL()	X
FLOOR()	-	NPER()	-	TO_DECIMAL38()	-
FV()	-	PERCENTILE()	-	TO_FLOAT()	X
GET_DATE_PART()	X	PMT()	-	TO_INTEGER()	X
GET_TIMEZONE()	-	POWER()	X	TO_TIMESTAMP_TZ()	-
GET_TIMESTAMP()	-	PV()	-	TRUNC(DATE)	X
GREATEST()	X	RAND()	X	TRUNC(NUMBER)	-
IIF()	X	RATE()	-	UPPER()	X
IN()	X	REG_EXTRACT()	X		
		REG_MATCH()	X		
		REG_REPLACE	X	VARIANCE()	X

Supported operators for Databricks Delta mappings

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 can be used in an Databricks Delta database. Columns marked with an X indicate that the operator can be pushed to the Databricks Delta database by using full pushdown optimization.

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

Supported transformations for Databricks Delta mappings

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

The following table summarizes the availability of transformations that you can push down to Databricks Delta. Columns marked with an X indicate that the transformation can be pushed to Databricks Delta by using full pushdown optimization. Columns marked with a dash (-) symbol indicate that the transformation cannot be pushed to the database.

Transformations	Pushdown	Guidelines
Aggregator	X You can perform the following aggregate calculations: - AVG - COUNT - FIRST - LAST - MAX - MIN - SUM - STDDEV - VARIANCE	When you configure an Aggregator transformation, you must use each of the incoming ports either in an aggregate function or in a group by field to define how to group data for aggregate expressions.
Expression	X	

Transformations	Pushdown	Guidelines
Filter	X	
Joiner	X	
Lookup	X Both connected and unconnected lookups are applicable.	<p>You can add the following lookups:</p> <ul style="list-style-type: none"> - Cached - Uncached - Unconnected with cached <p>When you configure a cached lookup, select the Multiple Matches property value as Report all rows in the lookup properties for pushdown optimization to work.</p> <p>When you configure an unconnected Lookup transformation, consider the following rules:</p> <ul style="list-style-type: none"> - 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.
Sorter	X	
Union	X	
Router	X	
Update Strategy	-	
Sequence Generator	-	

Supported data types for Databricks Delta mappings

The following table lists the Databricks Delta data types that can be used for pushdown optimization:

Databricks Delta Data Type	Transformation Data Type
Binary	Binary
Bigint	Bigint
Boolean	Integer
Date	Date/Time
Decimal	Decimal Precision 1 to 17 digits, scale 0 to 17.
Double	Double
Float	Double
Int	Integer

Databricks Delta Data Type	Transformation Data Type
Smallint	Integer
String	String
Tinyint	Integer
Timestamp	Date/Time

Supported features for Databricks Delta mappings

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

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

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

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

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.

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.

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.

Rules and guidelines for pushdown optimization

Use the following rules and guidelines when you push functions to a Databricks Delta database:

- When you configure a Filter transformation or specify a filter condition, do not specify special characters.
- 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 configure pushdown optimization, ensure that the Expression transformation does not contain a variable port.

Sources and targets

Consider the following rules and guidelines for the Databricks Delta source or target when you configure pushdown optimization in a mapping:

- 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.

Functions

When you push functions to Databricks Delta, adhere to the following guidelines:

- 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

CHAPTER 6

Data type reference

This chapter includes the following topics:

- [Databricks Delta data type reference overview, 37](#)
- [Databricks Delta and transformation data types, 37](#)

Databricks Delta data type reference overview

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.

Databricks Delta Data Type	Transformation Data Type	Range and Description
Date	Date/Time	January 1,0001 to December 31,9999.
Decimal	Decimal	For Cloud Data Integration mappings: Precision 1 to 18 digits, scale 0 to 18. For elastic mappings: Precision 1 to 38, scale 0 to 38.
Double	Double	Precision 15.
Float	Double	Precision 7.
Int	Integer	-2,147,483,648 to +2,147,483,647.
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 23:59:59.997997.

INDEX

C

Cloud Application Integration community
URL [5](#)
Cloud Developer community
URL [5](#)
connections
Databricks Delta [11](#)
Create target
rules and guidelines [22](#)
create target at runtime [21](#)

D

Data Integration community
URL [5](#)
data types [37](#)
Databricks Delta
connection properties [11](#)
pushdown through Redshift V2 Connection [28](#)
Databricks Delta connection
configuration [28](#)
Databricks Delta connections
overview [11](#)
Databricks Delta connector
datetime format [25](#)
rules and guidelines [25](#)
Databricks Delta Connector
overview [7](#)
supported task and object types [8](#)
Databricks Delta connector rules and guidelines [23](#)

E

elastic mapping
example [24](#)

F

field delimiter [17](#)

G

Google BigQuery
pushdown optimization overview [27](#)

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](#)
source properties [17](#)
target properties [19](#)

N

native data type [37](#)

P

properties
in mappings [17, 19](#)
pushdown optimization
functions [29, 31](#)
transformations [29, 31](#)
Pushdown Optimization
Rules and Guidelines for Functions [35](#)

S

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

T

tracing level [17](#)
transformation data type [37](#)
transformations
pushdown optimization [31](#)
trust site
description [6](#)

U

upgrade notifications [6](#)

W

web site [5](#)