



Informatica® PowerExchange for Google  
BigQuery

10.5.6

# User Guide

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.

Informatica, the Informatica logo, and PowerExchange 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.

Subject to your opt-out rights, the software will automatically transmit to Informatica in the USA information about the computing and network environment in which the Software is deployed and the data usage and system statistics of the deployment. This transmission is deemed part of the Services under the Informatica privacy policy and Informatica will use and otherwise process this information in accordance with the Informatica privacy policy available at <https://www.informatica.com/in/privacy-policy.html>. You may disable usage collection in Administrator tool.

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.

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>.

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

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

Publication Date: 2024-05-31

# Table of Contents

<b>Preface .....</b>	<b>5</b>
Informatica Resources. ....	5
Informatica Network. ....	5
Informatica Knowledge Base. ....	5
Informatica Documentation. ....	5
Informatica Product Availability Matrices. ....	6
Informatica Velocity. ....	6
Informatica Marketplace. ....	6
Informatica Global Customer Support. ....	6
 <b>Chapter 1: Introduction to PowerExchange for Google BigQuery.....</b>	 <b>7</b>
PowerExchange for Google BigQuery Overview. ....	7
Introduction to Google BigQuery. ....	7
 <b>Chapter 2: PowerExchange for Google BigQuery Configuration.....</b>	 <b>8</b>
PowerExchange for Google BigQuery Configuration Overview. ....	8
Prerequisites. ....	8
 <b>Chapter 3: Configuring HTTP Proxy Options.....</b>	 <b>10</b>
Configuring HTTP Proxy Options for the Mercury Client. ....	10
Configuring HTTP Proxy Options for the Data Integration Service. ....	10
 <b>Chapter 4: Google BigQuery Connections.....</b>	 <b>12</b>
Google BigQuery Connections Overview. ....	12
Connection Modes. ....	12
Connection Mode Example. ....	13
Rules and Guidelines for Google BigQuery Connection Modes. ....	15
Google BigQuery Connection Properties. ....	16
Creating a Google BigQuery Connection. ....	18
 <b>Chapter 5: PowerExchange for Google BigQuery Data Objects.....</b>	 <b>19</b>
Google BigQuery Data Object Overview. ....	19
Google BigQuery Data Object Properties. ....	20
Read Modes. ....	20
Write Modes. ....	21
Pre SQL and Post SQL Commands. ....	21
Using Merge Query for Update, Upsert, and Delete Operations. ....	22
Google BigQuery Data Object Read Operation. ....	23
Output Properties of the Data Object Read Operation. ....	23
Google BigQuery Data Object Write Operation. ....	26

Input Properties of the Data Object Write Operation. . . . .	26
Creating an Google BigQuery Data Object. . . . .	30
Creating a Google BigQuery Data Object Operation. . . . .	31
Creating a Google BigQuery Target. . . . .	31
Rules and Guidelines for PowerExchange for Google BigQuery Data Object Operations. . . . .	32
<b>Chapter 6: PowerExchange for Google BigQuery Mappings.....</b>	<b>34</b>
PowerExchange for Google BigQuery Mappings Overview. . . . .	34
Mapping Validation and Run-time Environments. . . . .	34
<b>Chapter 7: Google BigQuery Lookup.....</b>	<b>36</b>
Google BigQuery Lookup Overview. . . . .	36
General Properties. . . . .	37
Ports Properties. . . . .	37
Run-time Properties. . . . .	38
Lookup Properties. . . . .	38
Adding a Google BigQuery Data Object Read Operation as a Lookup in a Mapping. . . . .	39
Rules and Guidelines for Google BigQuery Lookup Operation. . . . .	39
<b>Chapter 8: Google BigQuery Run-Time Processing.....</b>	<b>40</b>
Google BigQuery Run-Time Processing Overview. . . . .	40
Filter Expression. . . . .	40
Native Expression. . . . .	40
Platform Expression. . . . .	41
Partitioning. . . . .	41
Parameterization for Google BigQuery Sources. . . . .	42
Parameterization for Google BigQuery Targets. . . . .	43
<b>Appendix A: Google BigQuery Data Type Reference.....</b>	<b>44</b>
Data Type Reference Overview. . . . .	44
Google BigQuery and Transformation Data Types. . . . .	44
<b>Index. . . . .</b>	<b>46</b>

# Preface

Use the *Informatica PowerExchange for Google BigQuery User Guide* to learn how to read from and write to Google BigQuery by using the Developer tool. Learn to create a Google BigQuery connection, develop and run mappings in the native environment and in the Hadoop environments.

## Informatica Resources

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

### Informatica Network

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

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

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

### Informatica Knowledge Base

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

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

### Informatica Documentation

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

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

## Informatica Product Availability Matrices

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

## Informatica Velocity

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

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

## Informatica Marketplace

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

## Informatica Global Customer Support

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

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

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

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

## CHAPTER 1

# Introduction to PowerExchange for Google BigQuery

This chapter includes the following topics:

- [PowerExchange for Google BigQuery Overview, 7](#)
- [Introduction to Google BigQuery, 7](#)

## PowerExchange for Google BigQuery Overview

You can use PowerExchange for Google BigQuery to extract data from and load data to Google BigQuery.

You can use Google BigQuery objects as sources and targets in mappings. When you use Google BigQuery objects in mappings, you must configure properties specific to Google BigQuery. PowerExchange for Google BigQuery uses the Google APIs to integrate with Google BigQuery and Google Cloud Storage.

You can validate and run Google BigQuery mappings in the native environment or on the Spark engine in the Hadoop environment.

### Example

Your organization is an open source log data collector, which collects log data from multiple sources and unifies them.

Logs help you understand how systems and applications perform. As the scale and complexity of the system increases, it is difficult to manage multiple logs from different sources.

To overcome this problem, you can use PowerExchange for Google BigQuery to write data to a Google BigQuery target and query terabytes of logs in seconds. You can then use the data to fix and improve the system performance in near real time.

## Introduction to Google BigQuery

Google BigQuery is a fast, highly scalable, cost-effective and fully managed enterprise data warehouse that the Google Cloud Platform provides. You can store and analyze massive amounts of data using ANSI:2011 compliant SQL queries.

## CHAPTER 2

# PowerExchange for Google BigQuery Configuration

This chapter includes the following topics:

- [PowerExchange for Google BigQuery Configuration Overview, 8](#)
- [Prerequisites, 8](#)

## PowerExchange for Google BigQuery Configuration Overview

The PowerExchange for Google BigQuery installs with Informatica Services. You can enable PowerExchange for Google BigQuery with a license key.

To configure PowerExchange for Google BigQuery, complete the prerequisites.

## Prerequisites

To use PowerExchange for Google BigQuery, perform the following steps:

- Install and configure Informatica Services.
- Install and configure the Developer tool. You can install the Developer tool when you install Informatica clients.
- Create and configure a Model Repository Service and a Data Integration Service in the Informatica domain.
- Verify that you have write permissions on all the directories within the `<Informatica installation directory> directory`.
- Ensure that the PowerExchange for Google BigQuery license is activated.
- Ensure you have a Google service account to access Google BigQuery.
- Ensure you have the `client_email`, `project_id`, and `private_key` values for the service account. You will need to enter these details when you create a Google BigQuery connection in Developer tool.
- Ensure you have the project ID, dataset ID, source table name, and target table name when you create mappings in Developer Tool.



- Verify that you have read and write access to the Google BigQuery dataset that contains the source table and target table.
- When you read data from or write data to a Google BigQuery table, you must have the required permissions to run the mapping successfully. When you only read data from a Google BigQuery table, you must have the required permissions to read data successfully.
- If your organization passes data through a proxy, virtual private cloud, or protective firewall, you must configure your firewall to allow the `www.googleapis.com` and `www.accounts.google.com` URI for PowerExchange for Google BigQuery to transfer data through a proxy, virtual private cloud, or firewall.
- If you use bulk mode, verify that you have write access to the Google Cloud Storage path where the Data Integration Service creates the staging file.
- If you use staging mode, verify that you have read access to the Google Cloud Storage path where the Data Integration Service creates the staging file to store the data from the Google BigQuery source.

## CHAPTER 3

# Configuring HTTP Proxy Options

This chapter includes the following topics:

- [Configuring HTTP Proxy Options for the Mercury Client, 10](#)
- [Configuring HTTP Proxy Options for the Data Integration Service, 10](#)

## Configuring HTTP Proxy Options for the Mercury Client

You can configure the proxy server authentication settings in the `developerCore.ini` file for the Mercury client.

1. Ensure that you enable the proxy server settings from your web browser.
2. Access the `developerCore.ini` file from the following location: <PowerCenter Installation Directory>\clients\10.5.6\clients\DeveloperClient
3. Add the following properties to the `developerCore.ini` file:

```
-Djdk.http.auth.tunneling.disabledSchemes=(Value is a space/empty)
-DGBQ.http.proxyHost=<hostname>
-DGBQ.http.proxyPort=<port>
-DGBQ.http.proxyUser=<username if proxy is auth>
-DGBQ.http.proxyPassword=<password if proxy is auth>
```
4. Restart the Mercury client.

## Configuring HTTP Proxy Options for the Data Integration Service

You can configure the proxy server authentication settings for the Data Integration Service from the Administrator Console.

1. Ensure that you enable the proxy server settings from your web browser.
2. In the Administrator Console, navigate to the Data Integration Service for which you want to set the proxy server settings.

3. You can do one of the following options:

- In the Service Process, add the following JVM options in the custom properties section:

```
JVMOption1=-Djdk.http.auth.tunneling.disabledSchemes=(Value is a space/empty)
```

```
JVMOption2=-DGBQ.http.proxyHost=<hostname>
```

```
JVMOption3=-DGBQ.http.proxyPort=<port>
```

```
JVMOption4=-DGBQ.http.proxyUser=<username if proxy is auth>
```

```
JVMOption5=-DGBQ.http.proxyPassword=<password if proxy is auth>
```

(or)

- In the local process, add the following JVM options in the JVM Command line:

```
-Djdk.http.auth.tunneling.disabledSchemes=(Value is a space/empty) -
```

```
DGBQ.http.proxyHost=<hostname> -DGBQ.http.proxyPort=<port> -
```

```
DGBQ.http.proxyUser=<username if proxy is auth> -
```

```
DGBQ.http.proxyPassword=<password if proxy is auth>
```

4. Restart the Data Integration Service.

## CHAPTER 4

# Google BigQuery Connections

This chapter includes the following topics:

- [Google BigQuery Connections Overview, 12](#)
- [Connection Modes, 12](#)
- [Connection Mode Example, 13](#)
- [Rules and Guidelines for Google BigQuery Connection Modes, 15](#)
- [Google BigQuery Connection Properties, 16](#)
- [Creating a Google BigQuery Connection, 18](#)

## Google BigQuery Connections Overview

Use a Google BigQuery connection to access a Google BigQuery database.

Use the Google BigQuery connection to import Google BigQuery metadata, create data objects, preview data, and run mappings. When you create a Google BigQuery connection, you define the connection attributes that the Developer tool uses to connect to the Google BigQuery dataset.

Use the Developer tool, Administrator tool, or infacmd to create a Google BigQuery connection.

## Connection Modes

You can configure a Google BigQuery connection to use one of the following connection modes:

### Simple mode

If you use simple mode, PowerExchange for Google BigQuery flattens each field within the Record data type field as a separate field in the Google BigQuery data object.

### Hybrid mode

If you use hybrid mode, PowerExchange for Google BigQuery displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the Google BigQuery data object.

### Complex mode

If you use complex mode, PowerExchange for Google BigQuery displays all the columns in the Google BigQuery table as a single field of the String data type in the Google BigQuery data object.

## Connection Mode Example

PowerExchange for Google BigQuery reads and writes the Google BigQuery data based on the connection mode that you configure for the Google BigQuery connection.

You have a Customers table in Google BigQuery that contains primitive fields and the **Address** field of the Record data type. The Address field contains two primitive sub-fields, **City** and **State**, of the String data type.

The following image shows the schema of the Customers table in Google BigQuery:

<b>ID</b>	INTEGER	NULLABLE
<b>Name</b>	STRING	NULLABLE
<b>Address</b>	RECORD	NULLABLE
<b>Address.City</b>	STRING	NULLABLE
<b>Address.State</b>	STRING	NULLABLE
<b>Mobile</b>	STRING	REPEATED
<b>Totalpayments</b>	FLOAT	NULLABLE
<b>age</b>	INTEGER	REPEATED

The following table shows the Customers table data in Google BigQuery:

ID	Name	Address.City	Address.State	Mobile	Totalpayments
14	John	LOS ANGELES	CALIFORNIA	+1-9744884744	18433.90
				+1-8267389993	
29	Jane	BOSTON	MANHATTAN	+1-8789390309	28397.33
				+1-9876553784	
				+1-8456437848	

## Simple Mode

If you use simple connection mode, PowerExchange for Google BigQuery flattens each field within the Record data type field as a separate field in the **Columns** tab of the Google BigQuery data object.

The following table shows two separate fields, Address\_City and Address\_State, for the respective sub-fields within the Address Record field in the Customers table:

ID	Name	Address_City	Address_State	Mobile	Totalpayments
14	John	LOS ANGELES	CALIFORNIA	+1-9744884744	18433.90
14	John	LOS ANGELES	CALIFORNIA	+1-8267389993	18433.90
29	Jane	BOSTON	MANHATTAN	+1-8789390309	28397.33
29	Jane	BOSTON	MANHATTAN	+1-9876553784	28397.33
29	Jane	BOSTON	MANHATTAN	+1-8456437848	28397.33

The following image shows the Address\_State and Address\_City fields in the **Columns** tab of the Google BigQuery data object:

Object Details:

General Columns Advanced									
	Name	Native Name	Type	Precision	Scale	Repeated	isRecord	Access Type	Description
1	ID	ID	INTEGER	19	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
2	Name	Name	STRING	255	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
3	Address_State	Address.State	STRING	255	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
4	Address_City	Address.City	STRING	255	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
5	Mobile	Mobile	STRING	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read and Write	
6	TotalPayments	TotalPayments	FLOAT	16	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
7	age	age	INTEGER	19	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read and Write	

## Hybrid Mode

If you use hybrid connection mode, PowerExchange for Google BigQuery displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the **Columns** tab of the Google BigQuery data object.

The following image shows the Address field in the **Columns** tab of the Google BigQuery data object:

Object Details:

General Columns Advanced									
	Name	Native Name	Type	Precision	Scale	Repeated	isRecord	Access Type	Description
1	ID	ID	INTEGER	19	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
2	Name	Name	STRING	255	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
3	Address	Address	STRING	65535	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Read and Write	
4	Mobile	Mobile	STRING	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read and Write	
5	TotalPayments	TotalPayments	FLOAT	16	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	
6	age	age	STRING	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Read and Write	

## Complex Mode

If you use complex connection mode, PowerExchange for Google BigQuery displays all the columns in the Google BigQuery table as a single field of the String data type in the **Columns** tab of the Google BigQuery data object.

The following image shows the STRING\_DATA field in the **Columns** tab of the Google BigQuery data object:

Object Details:

General Columns Advanced									
	Name	Native Name	Type	Precision	Scale	Repeated	isRecord	Access Type	Description
1	STRING_DATA	STRING DATA	STRING	65535	0	<input type="checkbox"/>	<input type="checkbox"/>	Read and Write	

## Rules and Guidelines for Google BigQuery Connection Modes

### Simple Mode

Consider the following rules and guidelines when you configure a Google BigQuery connection to use simple connection mode:

- If the Google BigQuery data object contains repeated columns, you cannot configure filters for these columns in a data object read operation.
- If the Google BigQuery table contains more than one repeated column, you cannot preview data.
- If the Google BigQuery data object contains columns of the Record data type and repeated columns, you cannot configure update, upsert, and delete operations for these columns in a data object write operation.
- If the Google BigQuery data object contains columns that are marked as required in Google BigQuery, you cannot configure upsert operation for these columns in a data object write operation.
- You can configure upsert operation on columns of primitive data types in the Google BigQuery table.
- When you configure an upsert operation, you cannot perform update operation on multiple rows in the Google BigQuery target.
- When you read data from a Google BigQuery source, you must not map more than one repeated column in a single mapping. You must create multiple mappings for each repeated column.

### Hybrid Mode

Consider the following rules and guidelines when you configure a Google BigQuery connection to use hybrid connection mode:

- If the Google BigQuery data object contains columns of the Record data type and repeated columns, you cannot configure filters for these columns in a data object read operation.
- You cannot configure update, upsert, and delete operations for columns of the Record data type and repeated columns.
- You must select JSON (Newline Delimited) format as the data format of the staging file under the data object write operation advanced properties. You can use CSV format as the data format of the staging file only when the Google BigQuery table does not contain columns of the Record data type or repeated columns.
- The following CSV formatting options in the data object write operation advanced properties are not applicable:
  - Allow Quoted Newlines

- Field Delimiter
- Allow Jagged Rows

### Complex Mode

Consider the following rules and guidelines when you configure a Google BigQuery connection to use complex connection mode:

- When you configure a Google BigQuery connection to use complex connection mode, you cannot configure filters in a data object read operation.
- You cannot configure update, upsert, and delete operations.
- You must select JSON (Newline Delimited) format as the data format of the staging file under the data object write operation advanced properties.
- You cannot use CSV format as the data format of the staging file. The following CSV formatting options in the data object write operation advanced properties are not applicable:
  - Allow Quoted Newlines
  - Field Delimiter
  - Allow Jagged Rows

## Google BigQuery Connection Properties

When you set up a Google BigQuery connection, you must configure the connection properties.

**Note:** The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Google BigQuery connection properties:

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account in Google BigQuery.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account in Google BigQuery.
Connection mode	<p>The mode that you want to use to read data from or write data to Google BigQuery.</p> <p>Select one of the following connection modes:</p> <ul style="list-style-type: none"> <li>- Simple. Flattens each field within the Record data type field as a separate field in the mapping.</li> <li>- Hybrid. Displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the mapping.</li> <li>- Complex. Displays all the columns in the Google BigQuery table as a single field of the String data type in the mapping.</li> </ul> <p>Default is Simple.</p>



Property	Description
Schema Definition File Path	<p>Specifies a directory on the client machine where the Data Integration Service must create a JSON file with the sample schema of the Google BigQuery table. The JSON file name is the same as the Google BigQuery table name.</p> <p>Alternatively, you can specify a storage path in Google Cloud Storage where the Data Integration Service must create a JSON file with the sample schema of the Google BigQuery table. You can download the JSON file from the specified storage path in Google Cloud Storage to a local machine.</p>
Project ID	<p>Specifies the <code>project_id</code> value present in the JSON file that you download after you create a service account in Google BigQuery.</p> <p>If you have created multiple projects with the same service account, enter the ID of the project that contains the dataset that you want to connect to.</p>
Storage Path	<p>This property applies when you read or write large volumes of data.</p> <p>Path in Google Cloud Storage where the Data Integration Service creates a local stage file to store the data temporarily.</p> <p>You can either enter the bucket name or the bucket name and folder name.</p> <p>For example, enter <code>gs://&lt;bucket_name&gt;</code> or <code>gs://&lt;bucket_name&gt;/&lt;folder_name&gt;</code></p>
Dataset ID	Not applicable for PowerExchange for Google BigQuery.
Use Legacy SQL For Custom Query	Not applicable for PowerExchange for Google BigQuery.
Dataset Name for Custom Query	Not applicable for PowerExchange for Google BigQuery.
Staging Dataset	The Google BigQuery dataset name where you want to create the staging table to stage the data. You can define a Google BigQuery dataset that is different from the source or target dataset.
Region ID	<p>The region name where the Google BigQuery dataset resides.</p> <p>For example, if you want to connect to a Google BigQuery dataset that resides in Las Vegas region, specify <b>us-west4</b> as the <b>Region ID</b>.</p> <p><b>Note:</b> In the <b>Storage Path</b> connection property, ensure that you specify a bucket name or the bucket name and folder name that resides in the same region as the dataset in Google BigQuery.</p> <p>For more information about the regions supported by Google BigQuery, see the following Google BigQuery documentation: <a href="https://cloud.google.com/bigquery/docs/locations">https://cloud.google.com/bigquery/docs/locations</a></p>
Optional Properties	<p>Specifies whether you can configure certain source and target functionalities through custom properties.</p> <p>You can select one of the following options:</p> <ul style="list-style-type: none"> <li>- None. Select if you do not want to configure any custom properties.</li> <li>- Required. If you want to specify custom properties to configure the source and target functionalities.</li> </ul> <p>Default is None.</p>
Provide Optional Properties	<p>Comma-separated key-value pairs of custom properties to enable certain source and target functionalities.</p> <p>Appears only when you select <b>Required</b> in the Optional Properties.</p> <p>For more information about the list of custom properties that you can specify, see the Informatica Knowledge Base article: <a href="https://kb.informatica.com/faq/7/Pages/26/632722.aspx">https://kb.informatica.com/faq/7/Pages/26/632722.aspx</a></p>

# Creating a Google BigQuery Connection

Create a Google BigQuery connection before you create a Google BigQuery data object.

1. In the Developer tool, click **Window > Preferences**.
2. Select **Informatica > Connections**.
3. Expand the domain in the **Available Connections**.
4. Select the connection type **Enterprise Application > Google Big Query**, and click **Add**.
5. Enter a connection name and an optional description.
6. Select **Google Big Query** as the connection type.
7. Click **Next**.
8. Configure the connection properties.
9. Click **Test Connection** to verify the connection to Google BigQuery.
10. Click **Finish**.

## CHAPTER 5

# PowerExchange for Google BigQuery Data Objects

This chapter includes the following topics:

- [Google BigQuery Data Object Overview, 19](#)
- [Google BigQuery Data Object Properties, 20](#)
- [Read Modes, 20](#)
- [Write Modes, 21](#)
- [Pre SQL and Post SQL Commands, 21](#)
- [Using Merge Query for Update, Upsert, and Delete Operations, 22](#)
- [Google BigQuery Data Object Read Operation, 23](#)
- [Google BigQuery Data Object Write Operation, 26](#)
- [Creating an Google BigQuery Data Object, 30](#)
- [Creating a Google BigQuery Data Object Operation, 31](#)
- [Creating a Google BigQuery Target, 31](#)
- [Rules and Guidelines for PowerExchange for Google BigQuery Data Object Operations, 32](#)

## Google BigQuery Data Object Overview

A Google BigQuery data object is a physical data object that uses Google BigQuery as a source or target. A Google BigQuery data object is a physical data object that represents data based on a Google BigQuery resource.

You can configure the data object read and write operation properties that determine how you can read data from Google BigQuery sources and load data to Google BigQuery targets.

Create a Google BigQuery data object from the Developer tool. You can configure the data object read, and write operation properties that determine how the Data Integration Service reads data from Google BigQuery sources and loads data to Google BigQuery targets.

To read data from the Google BigQuery, create a data object read operation based on the Google BigQuery data object. Configure the read operation properties to determine how the Data Integration Service must read data from the Google BigQuery table. Add the read operation as a source in a mapping.

To write data to the Google BigQuery, create a data object write operation based on the Google BigQuery data object. Configure the write operation properties to determine how the Data Integration Service must write data to the Google BigQuery. Add the write operation as a Write transformation in a mapping.

You can use a Google BigQuery data object read operation to look up data in a Google BigQuery table. You can add the data object read operation to a mapping as a lookup transformation. You can look up data from a Google BigQuery table in a mapping based on a lookup condition. You can configure a cached lookup operation to cache the lookup table on the Spark engine and an uncached lookup operation in the native environment.

## Google BigQuery Data Object Properties

Specify the data object properties when you create the data object.

The following table describes the properties that you configure for the Google BigQuery data objects:

Property	Description
Name	Name of the Google BigQuery data object.
Location	The project or folder in the Model Repository Service where you want to store the Google BigQuery data object.
Connection	Name of the Google BigQuery connection.

## Read Modes

When you use PowerExchange for Google BigQuery, you can read data by using direct mode or staging mode. Before you choose a mode, see the Google documentation to understand the cost implications and trade-offs for each mode.

You can read data from a Google BigQuery source by using one of the following modes:

### Direct Mode

Use direct mode when the volume of data that you want to read is small. In direct mode, PowerExchange for Google BigQuery directly reads data from a Google BigQuery source. You can configure the number of rows that you want PowerExchange for Google BigQuery to read in one request.

### Staging Mode

Use staging mode when you want to read large volumes of data in a cost-efficient manner.

In staging mode, PowerExchange for Google BigQuery first exports the data from the Google BigQuery source into Google Cloud Storage. After the export is complete, PowerExchange for Google BigQuery downloads the data from Google Cloud Storage into a local stage file. You can configure the local stage file directory in the advanced source properties. PowerExchange for Google BigQuery then reads the data from the local stage file.

When you enable staging file compression, PowerExchange for Google BigQuery compresses the size of the staging file in Google Cloud Storage. PowerExchange for Google BigQuery then downloads the staging file and decompresses the staging file before it reads the file. To improve the performance and download data in parallel, you can configure the number of threads for downloading the staging file.

# Write Modes

When you use PowerExchange for Google BigQuery, you can write data by using bulk mode or streaming mode. Before you choose a mode, see the Google documentation to understand the cost implications and trade-offs for each mode.

You can write data to a Google BigQuery target by using one of the following modes:

## Bulk mode

Use bulk mode when you want to write large volumes of data in a cost-efficient manner.

In bulk mode, PowerExchange for Google BigQuery first writes the data to a staging file in Google Cloud Storage. When the staging file contains all the data, PowerExchange for Google BigQuery loads the data from the staging file to the Google BigQuery target.

When you enable staging file compression, PowerExchange for Google BigQuery compresses the size of the staging file before it writes data to Google Cloud Storage. PowerExchange for Google BigQuery writes the compressed file to Google Cloud Storage and then submits a load job to the Google BigQuery target.

**Note:** Enabling compression reduces the time that PowerExchange for Google BigQuery takes to write data to Google Cloud Storage. However, there will be a performance degradation when PowerExchange for Google BigQuery writes data from Google Cloud Storage to the Google BigQuery target.

PowerExchange for Google BigQuery deletes the staging file unless you configure the task or mapping to persist the staging file. You can choose to persist the staging file if you want to archive the data for future reference.

## Streaming mode

Use streaming mode when you want the Google BigQuery target data to be immediately available for querying and real-time analysis. Evaluate Google's streaming quota policies and billing policies before you use streaming mode.

In streaming mode, PowerExchange for Google BigQuery directly writes data to the Google BigQuery target. PowerExchange for Google BigQuery appends the data into the Google BigQuery target.

You can configure the number of rows that you want PowerExchange for Google BigQuery to stream in one request. If you want to stream a larger number of rows than the maximum permissible limit prescribed by Google, you can write the data to multiple smaller target tables instead of one large target table. You can create a template table based on which Google BigQuery must create multiple tables. You can define a unique suffix for each table. Google BigQuery creates each table based on the template table and adds the suffix to uniquely identify each table.

# Pre SQL and Post SQL Commands

You can specify **pre SQL** and **post SQL** data object read and write operation properties for Google BigQuery data objects. When you create a data object read or write operation in the Developer tool, you can specify SQL commands on the **Advanced** view.

You can perform the following operations by using pre SQL and post SQL commands:

- SELECT
- UPDATE

- DELETE

You can configure the options in Google BigQuery with a pre SQL or post SQL statement in the **pre SQL Configuration** or **post SQL Configuration** data object read and write operation properties for Google BigQuery data objects.

You must use the following format to specify a pre SQL configuration or a post SQL configuration:

```
<Option1:Value1,Option2:Value2,...OptionN:ValueN>
```

The following table shows the configuration options and supported values that you can specify in a pre SQL configuration or post SQL configuration:

Options	Supported Values
DestinationDataset	Dataset ID in Google BigQuery
DestinationTable	Table name in Google BigQuery
FlattenResults	True and False
UseLegacySQL	True and False
WriteDisposition	WRITE_TRUNCATE, WRITE_APPEND, and WRITE_EMPTY

## Using Merge Query for Update, Upsert, and Delete Operations

You can implement the Merge query to perform update, upsert, and delete operations on a Google BigQuery target in a single SQL statement.

To implement the Merge query, select the **Enable Merge** option in the advanced properties for the data object write operation.

### Rules and Guidelines

Consider the following rules and guidelines when you use the Merge query:

- When you configure a Google BigQuery connection to use the simple connection mode and select CSV as the staging file format, you must not mark any columns of the Record data type for update or upsert operation.
- When you configure a Google BigQuery connection to use the simple connection mode, the Google BigQuery target table must not contain repeated columns.
- When you configure a Google BigQuery connection to use hybrid connection mode, the Google BigQuery target table must not contain repeated columns as the key field.
- When you perform an update or upsert operation on a column of record data type and the column is set to nullable in Google BigQuery, ensure that there are no required fields in the record.
- When you perform upsert operation on a Google BigQuery target table, ensure that a column does not contains NULL values. Otherwise, the mapping fails.

# Google BigQuery Data Object Read Operation

Create a mapping with a Google BigQuery data object read operation to read data from Google BigQuery.

## Output Properties of the Data Object Read Operation

The output properties represent data that the Data Integration Service passes into the mapping pipeline. Select the output properties to configure advanced properties of the data object read operation.

The output properties of the data object read operation include general properties that apply to the data object operation. The output properties also include source, query, run-time, and advanced properties that apply to the Google BigQuery data object.

You can view and change the output properties of the data object read operation from the **General**, **Sources**, **Query**, **Run-time**, and **Advanced** tabs.

### General Properties

The general properties display the name and description of the data object read operation.

### Sources Properties

The sources properties list the Google BigQuery objects used in the data object read operation. You cannot join data from multiple sources of the Google BigQuery data object in a read operation.

### Query Properties

Use the **Query** tab to select specific records from a Google BigQuery table.

The following table describes the query properties that you configure for a data object read operation:

Property	Description
Query	Filter value in a read operation. The filter specifies the where clause of select statement. Use a filter to reduce the number of rows that the Data Integration Service reads from the source. When you enter a source filter, the Developer tool adds a WHERE clause to the default query. You can use the Native or Platform expression to select specific records.

### Run-time Properties

The run-time properties displays the name of the connection that the Data Integration Service uses to read data from the Google BigQuery table.

You can define the partition type as key range partitioning to read data from Google BigQuery data object that use simple or hybrid connection mode. When you configure key range partitioning, the Data Integration Service distributes rows of data based on a port or set of ports that you define as the partition key. You can define a range of values for each port. The Data Integration Service uses the key and ranges to send rows to the appropriate partition.

### Advanced Properties

The Data Integration Service reads data from Google BigQuery based on the data object read operation.

The Developer tool displays advanced properties for the Google BigQuery data object operation in the Advanced view.

The following table describes the advanced properties for a Google BigQuery data object read operation:

Property	Description
Source Dataset ID	Optional. Overrides the Google BigQuery dataset name specified in the Google BigQuery data object read operation.
Source Table Name	Overrides the Google BigQuery table name specified in the Google BigQuery data object read operation.
Number of Rows to Read	Specifies the number of rows to read from the Google BigQuery source. Enter a number greater than 0.
Spark Mode	Applies when you run a mapping on the Spark engine. Specifies the mode of the Spark engine to read data from the Google BigQuery source. You can select <b>Generic</b> or <b>Optimized</b> . Default is Generic mode. <b>Note:</b> Select Optimized mode to increase the performance to read data from the Google BigQuery source. For more information about the list of advanced properties that you can configure with Optimized Spark mode, see the Informatica How-To Library, "Configuring Optimized Spark Mode for PowerExchange for Google BigQuery Data Objects": <a href="https://docs.informatica.com/data-integration/powerexchange-adapters-for-informatica/h2l/1384-configuring-optimized-spark-mode-for-powerexchange-for-goog/abstract.html">https://docs.informatica.com/data-integration/powerexchange-adapters-for-informatica/h2l/1384-configuring-optimized-spark-mode-for-powerexchange-for-goog/abstract.html</a>
Allow Large Results	Determines whether PowerExchange for Google BigQuery must produce arbitrarily large result tables to query large source tables. If you select this option, you must specify a destination table to store the query results.
Query Results Table Name	Required if you select the <b>Allow Large Results</b> option. Specifies the destination table name to store the query results. If the table is not present in the dataset, PowerExchange for Google BigQuery creates the destination table with the name that you specify.
Job Poll Interval in Seconds	The number of seconds after which PowerExchange for Google BigQuery polls the status of the read job operation. Enter a number greater than 0. Default is 10.
Read Mode	Specifies the read mode to read data from the Google BigQuery source. Select one the following read modes: - Direct. In direct mode, PowerExchange for Google BigQuery reads data directly from the Google BigQuery source table. <b>Note:</b> When you use hybrid and complex connection mode, you cannot use direct mode to read data from the Google BigQuery source. - Staging. In staging mode, PowerExchange for Google BigQuery exports the data from the Google BigQuery source table into Google Cloud Storage. After the download is complete, PowerExchange for Google BigQuery downloads the data from Google Cloud Storage into the local stage file that you specify into the local stage file and then reads data from the local stage file. Default is Direct mode.
Number of Threads for Downloading Staging Files	Specifies the number of files that PowerExchange for Google BigQuery downloads at a time to enable parallel download. Enter a number greater than 0. This property applies to staging mode.



Property	Description
Data Format of the staging file	<p>Specifies the data format of the staging file.</p> <p>Select one of the following data formats:</p> <ul style="list-style-type: none"> <li>- JSON (Newline Delimited). Supports flat and record data with nested and repeated fields.</li> <li>- CSV. Supports flat data. If you use hybrid and complex connection mode, you cannot use CSV format as the data format of the staging file.</li> </ul>
Local Stage File Directory	<p>Specifies the directory on your local machine where PowerExchange for Google BigQuery stores Google BigQuery source data temporarily before it reads the data.</p> <p>This property applies to staging mode.</p>
Staging File Name	<p>Name of the staging file where data from the Google BigQuery source table is exported to Google Cloud Storage.</p> <p>This property applies to staging mode.</p>
Enable Staging File Compression	<p>Indicates whether to compress the size of the staging file in Google Cloud Storage before PowerExchange for Google BigQuery reads data from the staging file.</p> <p>You can enable staging file compression to reduce cost and transfer time.</p> <p>This property applies to staging mode.</p>
Persist Destination Table	<p>Indicates whether PowerExchange for Google BigQuery must persist the query results table after it reads data from the query results table.</p> <p>By default, PowerExchange for Google BigQuery deletes the query results table.</p>
pre SQL	<p>SQL statement that you want to run before reading data from the source.</p> <p>For example, if you want to select records in the database before you read the records from the table, specify the following pre SQL statement:</p> <pre>SELECT * FROM [api-project-80697026669:EMPLOYEE.DEPARTMENT] LIMIT 1000;</pre>
pre SQL Configuration	<p>Specify a pre SQL configuration.</p> <p>For example,</p> <pre>DestinationTable:PRESQL_SRC, DestinationDataset:EMPLOYEE, FlattenResults:False, WriteDisposition:WRITE_TRUNCATE, UseLegacySql:False</pre>
post SQL	<p>SQL statement that you want to run after reading data from the source.</p> <p>For example, if you want to update records in a table after you read the records from a source table, specify the following post SQL statement:</p> <pre>UPDATE [api-project-80697026669.EMPLOYEE.PERSONS_TGT_DEL] SET phoneNumber.number=1000011, phoneNumber.areaCode=100 where fullname='John Doe' SET phoneNumber.number =1000011, phoneNumber.areaCode=100 where fullname='John Doe'</pre>
post SQL Configuration	<p>Specify a post SQL configuration.</p> <p>For example,</p> <pre>DestinationTable:POSTSQL_SRC, DestinationDataset:EMPLOYEE, FlattenResults:True, WriteDisposition:WRITE_TRUNCATE, UseLegacySql:False</pre>
SQL Override Query	<p>Overrides the default SQL query used to extract data from the Google BigQuery source.</p> <p><b>Note:</b> When you specify a SQL override query, you must specify a dataset name in the SQL query. Ensure that the list of selected columns, data types, and the order of the columns that appear in the query matches the columns, data types, and order in which they appear in the source object.</p> <p><b>Note:</b> Ensure that you only map all the columns in the SQL override query to the target.</p>

Property	Description
Use Legacy SQL for SQL Override	Uses legacy SQL override query. You can clear this option to define a standard SQL override query.
Partitions Per Core	Specifies the number of partitions to split the data. Default is 1. This property applies to Optimized spark mode.

## Google BigQuery Data Object Write Operation

Create a mapping to write data to Google BigQuery. Use the Google BigQuery connection, and define the write operation properties to write data to Google BigQuery.

You can perform insert, update, delete, and upsert operations on a Google BigQuery target.

To perform a update, delete, or upsert operation, you must add an Update Strategy transformation to a mapping and flag the records in an update strategy expression.

**Note:** When you run a mapping on the Spark engine, you cannot add an Update Strategy transformation. Hence, you cannot perform update, delete, or upsert operations on a Google BigQuery target when you run a mapping on the Spark engine.

For more information about configuring an Update Strategy transformation, see "Update Strategy Transformation" in the *Informatica Developer Transformation Guide*.

## Input Properties of the Data Object Write Operation

Input properties represent data that the Data Integration Service writes to a Google BigQuery table. Select the input properties to configure run-time properties of the data object write operation. You can also specify advanced data object write operation properties to write data to Google BigQuery table.

The input properties of the data object write operation include general properties that apply to the data object write operation. Input properties also include source, run-time, and advanced properties that apply to the data object write operation.

You can view and change the input properties of the data object write operation from the **General**, **Sources**, **Run-time**, and **Advanced** tabs.

### General Properties

The general properties list the name and description of the data object write operation.

### Target Properties

The target properties list the Google BigQuery table in the data object write operation.

## Run-time Properties

The run-time properties displays the name of the connection that the Data Integration Service uses to write data to the Google BigQuery table.

You can also use the **Run-time** tab to configure partitioning. You can configure dynamic partitioning for Google BigQuery data object write operation.

## Advanced Properties

Google BigQuery data object write operation properties include advanced properties that apply to the Google BigQuery data object.

The Developer tool displays advanced properties for the Google BigQuery data object operation in the **Advanced** tab.

You can configure the following advanced properties in the data object write operation:

Property	Description
UpdateMode	<p>Determines the mode that the Data Integration Service uses to update rows in the Google BigQuery target.</p> <p>You can select one of the following modes:</p> <ul style="list-style-type: none"><li>- Update As Update. The Data Integration Service updates all rows flagged for update in the Update Strategy transformation if the entries exist.</li><li>- Update Else Insert. The Data Integration Service first updates all rows flagged for update in the Update Strategy transformation if the entries exist in the target. If the entries do not exist, the Data Integration Service inserts the entries.</li></ul> <p>Default is Update As Update.</p>
Target Dataset ID	<p>Optional. Overrides the Google BigQuery dataset name specified in the Google BigQuery data object write operation.</p>
Target Table Name	<p>Optional. Overrides the Google BigQuery target table name specified in the Google BigQuery data object write operation.</p>
Create Disposition	<p>Specifies whether PowerExchange for Google BigQuery must create the target table if it does not exist.</p> <p>You can select one of the following values:</p> <ul style="list-style-type: none"><li>- Create if needed. If the table does not exist, PowerExchange for Google BigQuery creates the table.</li><li>- Create never. If the table does not exist, PowerExchange for Google BigQuery does not create the table.</li></ul>
Write Disposition	<p>Specifies how PowerExchange for Google BigQuery must write data in bulk mode if the target table already exists.</p> <p>You can select one of the following values:</p> <ul style="list-style-type: none"><li>- Write append. If the target table exists, PowerExchange for Google BigQuery appends the data to the existing data in the table.</li><li>- Write truncate. If the target table exists, PowerExchange for Google BigQuery overwrites the existing data in the table.</li><li>- Write empty. If the target table exists and contains data, PowerExchange for Google BigQuery displays an error and does not write the data to the target.</li></ul> <p>PowerExchange for Google BigQuery writes the data to the target only if the target table does not contain any data.</p> <p><b>Note:</b> Write disposition is applicable for bulk mode. Write disposition is applicable only when you perform an insert operation on a Google BigQuery target.</p>

Property	Description
Spark Mode	<p>Applies when you run a mapping on the Spark engine.</p> <p>Specifies the mode of the Spark engine to write data into the Google bigQuery target.</p> <p>You can select <b>Generic</b> or <b>Optimized</b>.</p> <p>Default is Generic mode.</p> <p><b>Note:</b> Select Optimized mode to increase the performance to write data into the Google BigQuery target.</p> <p>For more information about the list of advanced properties that you can configure with Optimized Spark mode, see the Informatica How-To Library, "Configuring Optimized Spark Mode for PowerExchange for Google BigQuery Data Objects":</p> <p><a href="https://docs.informatica.com/data-integration/powerexchange-adapters-for-informatica/h2l/configuring-optimized-spark-mode-for-powerexchange-for-google-bi/abstract.html">https://docs.informatica.com/data-integration/powerexchange-adapters-for-informatica/h2l/configuring-optimized-spark-mode-for-powerexchange-for-google-bi/abstract.html</a></p>
Write Mode	<p>Specifies the mode to write data to the Google BigQuery target.</p> <p>You can select one of the following modes:</p> <ul style="list-style-type: none"> <li>- Bulk. In bulk mode, PowerExchange for Google BigQuery first writes the data to a staging file in Google Cloud Storage.</li> </ul> <p>When the staging file contains all the data, PowerExchange for Google BigQuery loads the data from the staging file to the BigQuery target. Google BigQuery then deletes the staging file unless you configure the mapping to persist the staging file.</p> <ul style="list-style-type: none"> <li>- Streaming. In streaming mode, PowerExchange for Google BigQuery directly writes data to the BigQuery target. PowerExchange for Google BigQuery writes the data into the target row by row.</li> </ul> <p>Default is Bulk mode.</p>
Streaming Template Table Suffix	<p>Specifies the suffix that PowerExchange for Google BigQuery adds to the individual target tables that it creates based on the template target table.</p> <p>This property applies to streaming mode.</p>
Rows per Streaming Request	<p>Specifies the number of rows that PowerExchange for Google BigQuery streams to the BigQuery target for each request.</p> <p>Enter a number greater than 0. Default is 500 rows.</p> <p>The maximum row size that PowerExchange for Google BigQuery can stream to the BigQuery target for each request is 10 MB.</p> <p>This property applies to streaming mode.</p>
Staging File Name	<p>Name of the staging file that PowerExchange for Google BigQuery creates in the Google Cloud Storage before it loads the data to the Google BigQuery target.</p> <p>This property applies to bulk mode.</p>
Data Format of the staging file	<p>Specifies the data format of the staging file.</p> <p>You can select one of the following data formats:</p> <ul style="list-style-type: none"> <li>- JSON (Newline Delimited). Supports flat and record data with nested and repeated fields.</li> <li>- CSV. Supports flat data.</li> </ul> <p><b>Note:</b> If you use hybrid and complex connection mode, you cannot use CSV format as the data format of the staging file.</p>
Persist Staging File After Loading	<p>Indicates whether PowerExchange for Google BigQuery must persist the staging file in the Google Cloud Storage after it writes the data to the Google BigQuery target. You can persist the staging file if you want to archive the data for future reference.</p> <p>By default, PowerExchange for Google BigQuery deletes the staging file in Google Cloud Storage.</p> <p>This property applies to bulk mode.</p>

Property	Description
Enable Staging File Compression	<p>Select this option to compress the size of the staging file before PowerExchange for Google BigQuery writes the data to the Google Cloud Storage and decompress the staging file before it loads the data to the Google BigQuery target.</p> <p>You can enable staging file compression to reduce cost and transfer time.</p>
Job Poll Interval in Seconds	<p>The number of seconds after which PowerExchange for Google BigQuery polls the status of the write job operation.</p> <p>Enter a number greater than 0. Default is 10.</p>
Number of Threads for Uploading Staging file	<p>The number of files that PowerExchange for Google BigQuery must upload to Google Cloud Storage in bulk mode.</p> <p>Enter a number greater than 0.</p>
Local Stage File Directory	<p>Specifies the directory on your local machine where PowerExchange for Google BigQuery stores the files temporarily before writing the data to the staging file in Google Cloud Storage.</p> <p>This property applies to bulk mode.</p> <p><b>Note:</b> This property is not applicable if you run mappings with the Spark engine in the Hadoop environment.</p>
Allow Quoted Newlines	Indicates whether PowerExchange for Google BigQuery must allow the quoted data sections with newline character in a .csv file.
Field Delimiter	Delimiter character for the fields in a .csv file.
Allow Jagged Rows	Indicates whether PowerExchange for Google BigQuery must accept the rows without trailing columns in a .csv file.
Pre SQL	<p>SQL statement that you want to run before writing data to the target.</p> <p>For example, if you want to select records from the database before you write the records into the table, specify the following pre SQL statement:</p> <pre>SELECT * FROM `api-project-80697026669.EMPLOYEE.RegionNation` LIMIT 1000</pre>
Pre SQL Configuration	<p>Specify a pre SQL configuration.</p> <p>For example,</p> <pre>DestinationTable:PRESQL_TGT2, DestinationDataset:EMPLOYEE, FlattenResults:False, WriteDisposition:WRITE_TRUNCATE, UseLegacySql:False</pre>
Post SQL	<p>SQL statement that you want to run after writing the data into the target.</p> <p>For example, if you want to update records in a table after you write the records into the target table, specify the following post SQL statement:</p> <pre>UPDATE [api-project-80697026669.EMPLOYEE.PERSONS_TGT_DEL] SET phoneNumber.number =1000011, phoneNumber.areaCode=100 where fullname='John Doe'</pre>
Post SQL Configuration	<p>Specify a post SQL configuration.</p> <p>For example,</p> <pre>DestinationTable:POSTSQL_SRC, DestinationDataset:EMPLOYEE, FlattenResults:True, UseLegacySQL:False</pre>

Property	Description
Target Schema Strategy	Not applicable for PowerExchange for Google BigQuery. Leave the default value of <b>RETAIN</b> unchanged.
Enable Merge	Implements Merge query to perform update, upsert, or delete operations on the Google BigQuery target table. Default is not selected.
Treat Empty String as Null	Determines whether PowerExchange for Google BigQuery must treat an empty string in the source as null in the Google BigQuery target.
Staging Dataset	Optional. Overrides the Google BigQuery staging dataset name that you specified in the connection and the Target Dataset ID target advanced property. Provide only the dataset name. For example, if the dataset name in a Google BigQuery project is <i>TEMP_STAGING</i> , then enter just the dataset name. Ensure that the staging dataset is in the same region of target dataset.

## Creating an Google BigQuery Data Object

Create a Google BigQuery data object to add to a mapping.

1. Select a project or folder in the **Object Explorer** view.
2. Click **File > New > Data Object**.
3. Select **Bigquery Data Object** and click **Next**.  
The **Bigquery Data Object** dialog box appears.
4. Enter a name for the data object.
5. Click **Browse** next to the **Location** option and select the target project or folder.
6. Click **Browse** next to the **Connection** option and select the Google BigQuery connection from which you want to import the Google BigQuery object.
7. To add a resource, click **Add** next to the **Selected Resources** option.  
The **Add Resource** dialog box appears.
8. Select the checkbox next to the Google BigQuery object you want to add and click **OK**.  
**Note:** While creating a data object, you must not select multiple Google BigQuery objects. After you create the Google BigQuery data object, click the **New (Insert)** button to add multiple objects.
9. Click **Finish**.  
The data object appears under Physical Data Objects in the project or folder in the **Object Explorer** view.

# Creating a Google BigQuery Data Object Operation

You can create the data object read, write, or lookup operation for Google BigQuery data objects. You can then add the Google BigQuery data object operation to a mapping.

1. Select the data object in the **Object Explorer** view.
2. Right-click and select **New > Data Object Operation**.  
The **Data Object Operation** dialog box appears.
3. Enter a name for the data object operation.
4. Select the type of data object operation. You can choose to create a read or write operation.
5. Click **Add**.  
The **Select Resources** dialog box appears.
6. Select the Google BigQuery data object for which you want to create the data object operation and click **OK**.
7. Click **Finish**.

The Developer tool creates the data object operation for the selected data object.

# Creating a Google BigQuery Target

You can create a Google BigQuery target using the **Create Target** option. PowerExchange for Google BigQuery creates the data object write operation for the Google BigQuery data object.

1. Select a project or folder in the **Object Explorer** view.
2. Select a source or a transformation in the mapping.
3. Delete the FileName port from the **Ports** properties of the data object read operation and save.  
**Note:** If you do not want to delete the FileName port, you must add a transformation and map the fields to the Source transformation. Then, right-click on the transformation and select **Create Target** option to create a Google BigQuery target.
4. Right-click on the Source transformation or the transformation and select **Create Target**.  
The **Create Target** dialog box appears.
5. Select **Others** and then select **Google BigQuery** from the list in the **Data Object Type** section.
6. Select **Link ports by name** as the **Link Type**.
7. Click **OK**.  
The **New Google BigQuery Data Object** dialog box appears.
8. Enter a name for the data object.

9. On the **Advanced** tab, set the following properties for the Google BigQuery data object write operation:

Property	Value
Dataset ID	Provide a Google BigQuery dataset name.
Create Disposition	Select <b>Create If Needed</b> .
Target Schema Strategy	Select <b>Retain</b>

10. Click **Finish**.

The new target appears under the **Physical Data Objects** category in the project or folder in the **Object Explorer** view.

**Note:** You must specify a connection for the newly created Google BigQuery target in the **Connection** field to run a mapping.

## Rules and Guidelines for PowerExchange for Google BigQuery Data Object Operations

Use the following rules and guidelines when you configure a PowerExchange for Google BigQuery data object read and write operation:

- You must define a primary key in the source table and target table for update and delete operations. If you do not define a primary key in the target table, the mapping fails to update records in or delete records from the target table.
- You cannot read hierarchical data from a Google BigQuery source, convert the hierarchical input to relational output, and write data to relational targets.
- You cannot perform more than one operation with a pre SQL or post SQL command.
- If you perform an UPDATE or DELETE operation with a pre SQL or post SQL command, you must specify the following parameter in the pre SQL configuration or post SQL configuration:  
`UseLegacySQL:False`
- To perform an upsert operation on a column, you must ensure that the column in the target table is set to Required in Google BigQuery.
- When you perform an upsert operation on a Google BigQuery target table, you cannot perform an update operation on multiple rows.
- When you configure a Google BigQuery connection in simple mode, you can perform upsert operation on columns of primitive data types only.
- When you import a Google BigQuery partitioned table for the data object read operation or write operation, you must use a Google BigQuery connection in hybrid mode.
- When you configure an uncached lookup operation to look up data from a Google BigQuery source, ensure that you do not select **Direct** for the Read Mode property. Otherwise, the mapping fails.
- When you configure an uncached lookup operation to look up data from a Google BigQuery source, ensure that you do not set the **Number of Rows to Read** property to a non-zero value. Otherwise, the mapping fails.



- When you configure an uncached lookup operation to look up data from a Google BigQuery source, ensure that you do not set the **On multiple matches** property to **Return Any**. Otherwise, the mapping fails.
- When you configure an uncached lookup operation to look up data from a Google BigQuery source, ensure that you do not configure the **SQL Override Query** property. Otherwise, the mapping fails.

## CHAPTER 6

# PowerExchange for Google BigQuery Mappings

This chapter includes the following topics:

- [PowerExchange for Google BigQuery Mappings Overview, 34](#)
- [Mapping Validation and Run-time Environments, 34](#)

## PowerExchange for Google BigQuery Mappings Overview

After you create a Google BigQuery data object read or write operation, you can create a mapping to extract data from a Google BigQuery source or load data to a Google BigQuery target.

You can define properties in an operation to determine how the Data Integration Service must extract data from a Google BigQuery source or load data to a Google BigQuery target. You can extract data from one or more Google BigQuery sources, and load data to one or more Google BigQuery targets. When the Data Integration Service extracts data from the source or loads data to the target, it converts the data based on the data types associated with the source or the target.

## Mapping Validation and Run-time Environments

You can validate and run mappings in the native environment or with the Spark engine in the Hadoop environment.

The Data Integration Service validates whether the mapping can run in the selected environment. You must validate the mapping for an environment before you run the mapping in that environment.

### Native environment

You can configure the mappings to run in the native environment. When you run mappings in the native environment, the Data Integration Service processes the mapping and runs the mapping from the Developer tool.

## Spark Engine

When you run mappings on the Spark engine, the Data Integration Service pushes the mapping to a Hadoop cluster and processes the mapping on a Spark engine. The Data Integration Service generates an execution plan to run mappings on the Spark engine.

You can view the plan in the Developer tool before you run the mapping and in the Administrator tool after you run the mapping.

For more information about the Hadoop environment and Spark engines, see the *Informatica Data Engineering Integration User Guide*.

## CHAPTER 7

# Google BigQuery Lookup

This chapter includes the following topics:

- [Google BigQuery Lookup Overview, 36](#)
- [General Properties, 37](#)
- [Ports Properties, 37](#)
- [Run-time Properties, 38](#)
- [Lookup Properties, 38](#)
- [Adding a Google BigQuery Data Object Read Operation as a Lookup in a Mapping, 39](#)
- [Rules and Guidelines for Google BigQuery Lookup Operation, 39](#)

## Google BigQuery Lookup Overview

You can use a Google BigQuery data object read operation to look up data in a Google BigQuery table.

You can add a Google BigQuery data object read operation as a lookup in a mapping. You can then configure a lookup condition to look up data from the Google BigQuery table. You can configure the following types of lookups on a Google BigQuery table:

### **Cached Lookup**

You can configure a connected cached lookup operation to cache the lookup data on the Spark engine or in the native environment.

### **Uncached Lookup**

You can configure an connected uncached lookup operation in the native environment when you do not want the Data Integration Service to cache the lookup data.

**Note:** When you configure an uncached lookup operation, the Data Integration Service does not consider the Google BigQuery data object read advanced properties. However, you must provide the Google BigQuery dataset name in the Google BigQuery data object read advanced properties to validate the read operation.

For more information about cached lookup, see *"Lookup Transformation" in the Developer Transformation Guide*.

# General Properties

The general properties display the name and description of the Google BigQuery lookup.

The following table describes the general properties that you can view and edit for a Google BigQuery lookup:

Property	Description
Name	Name of the Google BigQuery lookup.
Description	Description of the Google BigQuery lookup.
Physical Data Object	Name of the Google BigQuery data object read operation.
On multiple matches	Determines which row the Google BigQuery lookup returns when it finds multiple rows that match the lookup condition. You can select one of the following options: <ul style="list-style-type: none"><li>- Return any row</li><li>- Return all rows</li><li>- Report error</li></ul> <b>Note:</b> The <b>Return first row</b> or <b>Return last row</b> options are not applicable for PowerExchange for Google BigQuery.

# Ports Properties

The ports properties display the input ports from the source in the mapping to the Google BigQuery lookup. You can specify the ports to be available as output ports from the Google BigQuery lookup. The ports properties display the data types, precision, and scale of the source port.

The following table describes the ports properties:

Property	Description
Name	Name of the source port.
Type	Data type of the source port.
Precision	Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale.
Scale	Maximum number of digits after the decimal point of numeric values.
Output	Specify the ports that must be available as output ports from the Google BigQuery lookup.
Description	Description of the port.
Input Rules	A set of rules that filter the ports to include or exclude in the transformation based on port names or data type. Configure input rules when you define dynamic ports.

## Run-time Properties

Set the run-time properties to configure a cached lookup in a mapping.

When you enable lookup caching, the Data Integration Service caches the lookup values. The Data Integration Service queries the lookup source once, caches the values, and looks up values in the cache. Caching the lookup values can increase performance on large lookup tables. By default, the **Lookup caching enabled** check box is selected.

When you disable caching, the Data Integration Service does not cache the lookup values. The Data Integration Service queries the lookup source instead of building and querying the lookup cache. Each time a row passes, the Data Integration Service issues a SELECT statement to the lookup source for lookup values. Do not select the **Lookup caching enabled** check box to enable the uncached lookup.

## Lookup Properties

Specify the lookup properties to look up a Google BigQuery table. You can configure a lookup condition to look up data from the Google BigQuery table.

To configure a lookup condition, you can select one of the following options in the **Specify by** property:

- **Value:** Select this option if you want to configure a lookup condition using the column name.
- **Parameter:** Select this option if you want to parameterize the lookup condition.

The following table describes the lookup properties that you can specify for a Google BigQuery lookup if you select the **Value** option:

Property	Description
Lookup Column	The name of the columns that you want to look up.
Operator	Operators that you can use to filter records. You can select one of the following operators: =, !=, <=, >=, and
Input Port	The input source port.

The following table describes the lookup properties that you can specify for a Google BigQuery lookup if you select the **Parameter** option:

Property	Description
Parameter	The name of the parameter that you want to use to look up. You can also create a new parameter. Click <b>New</b> to create a new parameter. Enter the parameter name and specify an expression in the <b>New Parameter</b> dialog box. Click <b>Validate</b> to check if the expression that you specified is valid or not.

# Adding a Google BigQuery Data Object Read Operation as a Lookup in a Mapping

You can add a Google BigQuery data object read operation as a lookup to look up data in a Google BigQuery table.

1. Open a mapping from the **Object Explorer** view.
2. From the **Object Explorer** view, drag a Google BigQuery data object read operation to the mapping editor. The **Add to Mapping** dialog box appears.
3. Select **Lookup** to add the data object read operation as an operation to the mapping.
4. Select the Google BigQuery data object read operation and connect the lookup input ports and the lookup output ports.
5. In the **Properties** view, configure the following parameters:
  - a. On the **General** tab, select the option that you want the Data Integration Service to return when it finds multiple rows that match the lookup condition.
  - b. On the **Ports** tab, configure the output ports and input rules.
  - c. On the **Run-time** tab, ensure that lookup caching is enabled.
  - d. On the **Lookup** tab, configure the lookup condition.
6. When the mapping is valid, click **File > Save** to save the mapping to the Model repository.

## Rules and Guidelines for Google BigQuery Lookup Operation

Consider the following rules and guidelines when you configure a PowerExchange for Google BigQuery data object lookup operation:

- Informatica recommends to use a cached lookup transformation for better performance. If you use an uncached lookup, Informatica recommends that you use a Google BigQuery connection in Simple mode and select **Direct** as the **Read Mode** for the Google BigQuery data object lookup operation.
- You cannot configure an unconnected lookup transformation in a Google BigQuery data object lookup operation.
- You cannot apply the Lookup transformation on a column of Date/Time, Date, Time, or Timestamp data type.
- You cannot apply the Lookup transformation on a column of String data type that contains a single quote character.
- When you use a Google BigQuery connection in simple mode in an uncached Lookup transformation, you cannot configure a lookup condition on a column of Bytes, Boolean, Record, or Repeated data type.
- When you use a Google BigQuery connection in simple mode in a Lookup transformation, you cannot configure a lookup condition for fields of numeric, date, or datetime data types.

## CHAPTER 8

# Google BigQuery Run-Time Processing

This chapter includes the following topics:

- [Google BigQuery Run-Time Processing Overview, 40](#)
- [Filter Expression, 40](#)
- [Partitioning, 41](#)
- [Parameterization for Google BigQuery Sources, 42](#)
- [Parameterization for Google BigQuery Targets, 43](#)

## Google BigQuery Run-Time Processing Overview

When you create a Google BigQuery data object read or write operation, you define properties that determine how the Data Integration Service reads data from or writes data to a Google BigQuery database.

You can configure lookup caching, partitioning, and parameterization in the run-time properties.

## Filter Expression

To read specific data from a Google BigQuery table, you can configure a filter condition to query the Google BigQuery tables. You can use the Native or Platform expression to query specific columns in a Google BigQuery table.

### Native Expression

You can specify a native expression that use AND, OR, or nested conditions. The expression that you enter becomes the WHERE clause in the query used to retrieve records from the source.

An Google BigQuery filter consists of one or more Boolean expressions. The Boolean expressions uses the following format:

```
<Tablename.columnname><Operator><Value>
```

If you use logical operators, add the operators as a prefix to the expression list. Default is blank.



To filter records from an Google BigQuery source, set the native expression in the data object read operation.

**Note:** When you configure a Native expression and select a column that you want to filter, you must delete the `<Tablename.>` value in the native expression.

## Platform Expression

You can use the platform expression to select specific records from a Google BigQuery table based on the filter condition you specify.

The following table describes the properties you specify when you filter records from a Google BigQuery table when you use the platform expression filter:

Property	Description
Expression Type	The type of filter expression that you want to use to filter records.
Left Field	The Google BigQuery column on which you want to apply the filter condition. You cannot specify a platform expression for columns of the following data types: <ul style="list-style-type: none"><li>- Date</li><li>- Time</li><li>- Date/Time</li><li>- Timestamp</li></ul>
Operator	Simple operators you can use to filter records. You can select one of the following operators:  <code>=</code> , <code>!=</code> , <code>&lt;</code> , <code>&lt;=</code> , <code>&gt;</code> , and <code>&gt;=</code>
Right Field	The value you specify to filter Google BigQuery columns.

**Note:** When you create native or platform expression to query columns of Numeric data type in a Google BigQuery table, you must ensure that you use a Google BigQuery connection in hybrid connection mode.

## Partitioning

When you read data from or write data to Google BigQuery, you can configure partitioning to optimize the mapping performance at run time. You can configure partitioning for Google BigQuery mappings that you run in the native or Spark engine. The partition type controls how the Data Integration Service distributes data among partitions at partition points. You can configure a partition key for a Google BigQuery data object that uses a simple or hybrid connection mode.

You can define the partition type as key range partitioning. To configure key range partitioning, open the Google BigQuery data object read operation, and select the **Key Range** partition type option on the **Run-time** tab.

When you configure key range partitioning, the Data Integration Service distributes rows of data based on a port or set of ports that you define as the partition key. You can define a range of values for each port. The Data Integration Service uses the key and ranges to send rows to the appropriate partition.

When you use a simple connection mode in a Google BigQuery connection, you can configure a partition key for fields of the following data types:

- Integer
- Float

When you use a hybrid connection mode in a Google BigQuery connection, you can configure a partition key for fields of the following data types:

- Integer
- Float
- Numeric
- Timestamp

**Note:** You cannot configure a partition key for Record data type columns and repeated columns.

You can configure dynamic partitioning for Google BigQuery data object write operation. To configure dynamic partitioning, open the Google BigQuery data object write operation, and select the **Dynamic** partition type option on the **Runtime** tab.

When you configure dynamic partitioning, the Data Integration Service determines the number of partitions that it must create at run time. It scales the number of partitions based on factors such as the maximum parallelism value defined for the Data Integration Service and the mapping, and the number of CPUs available on the nodes where the mappings run.

When you configure the advanced properties in the data object write operation, you cannot use the **Write Disposition > Write truncate** option for multiple partitions. In such case, set the Maximum Parallelism property for the mapping to 1.

**Note:** When you configure key range partitioning, you cannot use the create target option.

## Parameterization for Google BigQuery Sources

You can parameterize the Google BigQuery connection and data object read operation properties to override the mapping properties at run time.

You can parameterize the following read operation properties for a Google BigQuery source:

- Source Dataset ID
- Source Table name
- Number of Rows to Read
- Query Results Table Name
- Job Poll Interval In Seconds
- Read Mode
- Number of Threads for Downloading Staging Files
- Data format of the staging file
- Local Stage File Directory
- Staging File Name
- pre SQL
- post SQL

- pre SQL Configuration
- post SQL Configuration
- SQL Override Query

## Parameterization for Google BigQuery Targets

You can parameterize the Google BigQuery connection and data object write operation properties to override the mapping properties at run time.

You can parameterize the following write operation properties for a Google BigQuery target:

- Target Dataset ID
- Target Table name
- Create Disposition
- Write Disposition
- Write Mode
- Streaming Template Table Suffix
- Rows per Streaming Request
- Staging File Name
- Number of Threads for Uploading Staging Files
- Data format of the staging file
- Job Poll Interval In Seconds
- Local Stage File Directory
- Staging File Name
- Field Delimiter
- pre SQL
- post SQL
- pre SQL Configuration
- post SQL Configuration

## APPENDIX A

# Google BigQuery Data Type Reference

This appendix includes the following topics:

- [Data Type Reference Overview, 44](#)
- [Google BigQuery and Transformation Data Types, 44](#)

## Data Type Reference Overview

Developer Tool uses the following data types in Google BigQuery mappings:

- Google BigQuery native data types. Google BigQuery data types appear in Google BigQuery definitions in a mapping.
- Transformation data types. Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Data Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

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

## Google BigQuery and Transformation Data Types

The following table lists the Google BigQuery data types that Developer Tool supports and the corresponding transformation data types:

Google BigQuery Data Type	Transformation Data Type	Range and Description for the Transformation Data Type
BOOLEAN	String	1 to 104,857,600 characters.
BYTE	Binary	1 to 104,857,600 bytes

Google BigQuery Data Type	Transformation Data Type	Range and Description for the Transformation Data Type
DATE	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
DATETIME	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
FLOAT	Double	Precision 15.
INTEGER	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
RECORD	String	1 to 104,857,600 characters Precision 65535, scale 0
NUMERIC	Decimal	Default precision 38, scale 9 PowerExchange for Google BigQuery supports precision upto 38 and scale upto 9.
STRING	String	1 to 104,857,600 characters. Precision 255, scale 0
TIME	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
TIMESTAMP	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)

# INDEX

## A

advanced properties  
input [27](#)

## B

bulk mode  
staging file [21](#)

## C

create  
    data object operation  
        create [31](#)  
    Google BigQuery data object [30](#)  
create target  
    Google BigQuery [31](#)  
creating  
    Google BigQuery connection [18](#)

## D

data types [44](#)

## G

general properties  
    input [26](#)  
Google BigQuery  
    data object properties [20](#)  
    data object read operation [23](#)  
    data object write operation [26](#)  
    introduction [7](#)  
    lookup overview [36](#)  
Google BigQuery connection  
    properties [16](#)  
Google BigQuery connections  
    connection modes [12](#)  
    connection modes example [13](#)  
    creating [18](#)  
    overview [12](#)  
Google BigQuery data object  
    create [30](#)  
    overview [19](#)  
Google BigQuery data types [44](#)  
Google BigQuery lookup  
    creating [39](#)  
    general properties [37](#)  
    lookup properties [38](#)  
    ports properties [37](#)

Google BigQuery parameterization  
    for sources [42](#)  
    for targets [43](#)  
Google BigQuery run-time processing  
    parameterization [42](#), [43](#)  
    partitioning [41](#)

## I

input properties [26](#)

## N

native environment  
    mappings [34](#)

## P

post SQL commands  
    entering [21](#)  
PowerExchange for Google BigQuery  
    configuration [8](#)  
    example [7](#)  
    overview [7](#)  
PowerExchange for Google BigQuery mappings  
    overview [34](#)  
pre SQL commands  
    entering [21](#)  
pre SQL configuration and post SQL configuration  
    entering [21](#)

## Q

query  
    native expression [40](#)  
    platform expression [40](#), [41](#)

## S

Spark engine  
    mappings [34](#)  
streaming mode  
    creating template table [21](#)

## T

transformation data types [44](#)

## U

upsert [22](#)

## W

write modes

bulk [21](#)

streaming [21](#)