# 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 Site .......... 6
  - Informatica Global Customer Support ........................ 6

**Chapter 1: Introduction to Google BigQuery Connector** ............... 7

  - Google BigQuery Connector Overview .......................... 7
  - Data Integration Hosted Agent .................................. 7
  - Google BigQuery Supported Task Types and Object Types ... 8
  - Google BigQuery Example ....................................... 8
  - Administration of Google BigQuery Connector .............. 8

**Chapter 2: Google BigQuery Connections** ................................ 11

  - Google BigQuery Connections Overview ...................... 11
  - Connection Modes .................................................. 11
    - Connection Mode Example .................................... 11
    - Rules and Guidelines for Google BigQuery Connection Modes .... 15
  - Google BigQuery connection properties ........................ 16

**Chapter 3: Google BigQuery Pushdown Optimization** .................. 18

  - Pushdown Optimization ........................................... 18
  - Pushdown Optimization Functions and Operators ............. 18
  - Pushdown Optimization Transformations and Data Types .... 20
  - Configuring a Google BigQuery ODBC Connection ............ 21
    - Configuring a Google BigQuery ODBC Connection on Windows ... 21
    - Configuring a Google BigQuery ODBC Connection on Linux .... 26
  - Create an ODBC Connection ..................................... 27
  - Configuring the Secure Agent for Pushdown Optimization .... 29

**Chapter 4: Synchronization Tasks with Google BigQuery Connector** .... 30

  - Pre SQL and Post SQL Commands ............................... 30
  - Google BigQuery Sources in Synchronization Tasks .......... 31
  - Read Modes ......................................................... 31
  - Advanced Properties for Google BigQuery Sources .......... 32
Preface

The Data Integration Google BigQuery Connector Guide contains information about how to set up and use Google BigQuery Connector. The guide explains how you can use Google BigQuery Connector to read data from and write data to Google BigQuery.

Informatica Resources

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email 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 Data Integration editions and applications as well as information about other 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:

To find resources on using Application Integration (the Informatica Cloud Real Time service), access the community at:
https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-application-integration/content

Developers can learn more and share tips at the Cloud Developer community:
Informatica Intelligent Cloud Services Marketplace

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


Data Integration Connector Documentation

You can access documentation for Data Integration Connectors at the Informatica Cloud Community:
https://network.informatica.com/cloud/index.htm

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.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 Site

Subscribe to the Informatica trust site for upgrade, maintenance, and incident notifications.

Status.Informatica.com displays the production status of all the Informatica cloud products. All maintenance updates are posted to this status 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 a single component, a single incident, or the site as a whole. Subscribing to the site as a whole is the best way to be certain you never miss an update. To subscribe, go to http://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 Google BigQuery Connector

This chapter includes the following topics:

- Google BigQuery Connector Overview, 7
- Data Integration Hosted Agent, 7
- Google BigQuery Supported Task Types and Object Types, 8
- Google BigQuery Example, 8
- Administration of Google BigQuery Connector, 8

Google BigQuery Connector Overview

You can use Google BigQuery Connector to connect to Google BigQuery from Data Integration. Use Google BigQuery Connector to read data from and write data to Google BigQuery. You can use a Google BigQuery object as a source and as a target in synchronization tasks, mapping tasks, and mappings.

When you run a task or mapping, the Secure Agent uses the JAVA client libraries of the Google APIs to integrate with Google BigQuery.

Data Integration Hosted Agent

You can use the Data Integration Hosted Agent (Hosted Agent) as a runtime environment for a Google BigQuery connection if you have the Cloud Runtime license.

Data Integration Secure Agents are installed locally. As an alternative to installing a Secure Agent, you can use a Hosted Agent. Hosted Agents are hosted at Data Integration hosting facility. The Data Integration hosting facility manages the Hosted Agent runtime environment and the agents that run in it. You cannot add, delete, or configure a Hosted Agent runtime environment. Because you do not install a Hosted Agent, you do not have access to files normally stored in the Secure Agent directory, such as configuration, success, and reject files.
Google BigQuery Supported Task Types and Object Types

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

The following table lists the Google BigQuery Connector object types that you can include in Data Integration tasks:

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mapping</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Google BigQuery 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 Google BigQuery Connector 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.

Administration of Google BigQuery Connector

Google BigQuery is a RESTful web service that the Google Cloud Platform provides.

Before you use Google BigQuery Connector, you must complete the following prerequisite tasks:

1. Create a Google account to access Google BigQuery.
2. On the Credentials page, navigate to the APIs and auth section, and create a service account. After you create the service account, you can download a JSON file that contains the client_email, project_id, and private_key values. You will need to enter these details when you create a Google BigQuery connection in Data Integration.
3. On the **Dashboards** page of the Google API Console, [https://console.developers.google.com/](https://console.developers.google.com/), enable the **BigQuery API** and **Google Cloud Storage JSON API**. Google BigQuery Connector uses the Google APIs to integrate with Google BigQuery and Google Cloud Storage. The following image shows the **Dashboard** page where you can enable the APIs:

4. Create a project and dataset in Google BigQuery. Verify that the dataset contains the source table and the target table. You will need to enter the project ID, dataset ID, source table name, and target table name when you create tasks and mappings in Data Integration. The following image shows a project:

5. Verify that you have read and write access to the Google BigQuery dataset that contains the source table and target table.

6. If you use bulk mode, verify that you have write access to the Google Cloud Storage path where the Secure Agent creates the staging file.
7. If you use staging mode, verify that you have read access to the Google Cloud Storage path where the Secure Agent creates the staging file to store the data from the Google BigQuery source.
This chapter includes the following topics:

- **Google BigQuery Connections Overview, 11**
- **Connection Modes, 11**
- **Google BigQuery connection properties, 16**

### Google BigQuery Connections Overview

Create a Google BigQuery connection to read data from a Google BigQuery source and write data to a Google BigQuery target. You must create a connection for each dataset that you want to connect to. You can use Google BigQuery connections in synchronization tasks, mapping tasks, and mappings. When you create a Google BigQuery connection, you can configure a connection mode based on how you want to read and write the data.

### Connection Modes

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

**Simple mode**

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

**Hybrid mode**

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

**Complex mode**

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

### Connection Mode Example

Google BigQuery Connector 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:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Address.City</th>
<th>Address.State</th>
<th>Mobile</th>
<th>Totalpayments</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>John</td>
<td>LOS ANGELES</td>
<td>CALIFORNIA</td>
<td>+1-9744884744</td>
<td>18433.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+1-8267389993</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Jane</td>
<td>BOSTON</td>
<td>MANHATTAN</td>
<td>+1-8789390309</td>
<td>28397.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+1-9876553784</td>
<td></td>
</tr>
</tbody>
</table>

**Simple Mode**

If you use simple connection mode, Google BigQuery Connector flattens each field within the Record data type field as a separate field in the **Field Mapping** tab.
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:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Address_City</th>
<th>Address_State</th>
<th>Mobile</th>
<th>Totalpayments</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>John</td>
<td>LOS ANGELES</td>
<td>CALIFORNIA</td>
<td>+1-9744884744</td>
<td>18433.90</td>
</tr>
<tr>
<td>14</td>
<td>John</td>
<td>LOS ANGELES</td>
<td>CALIFORNIA</td>
<td>+1-8267389993</td>
<td>18433.90</td>
</tr>
<tr>
<td>29</td>
<td>Jane</td>
<td>BOSTON</td>
<td>MANHATTAN</td>
<td>+1-8789390309</td>
<td>28397.33</td>
</tr>
<tr>
<td>29</td>
<td>Jane</td>
<td>BOSTON</td>
<td>MANHATTAN</td>
<td>+1-9876553784</td>
<td>28397.33</td>
</tr>
<tr>
<td>29</td>
<td>Jane</td>
<td>BOSTON</td>
<td>MANHATTAN</td>
<td>+1-8456437848</td>
<td>28397.33</td>
</tr>
</tbody>
</table>

The following image shows the fields in the Field Mapping tab of a synchronization task:

Hybrid Mode

If you use hybrid connection mode, Google BigQuery Connector displays all the top-level fields in the Google BigQuery table including Record data type fields. Google BigQuery Connector displays the top-level Record data type field as a single field of the String data type in the Field Mapping tab.

The following image shows the Field Mapping tab of a synchronization task:
Complex Mode

If you use complex connection mode, Google BigQuery Connector displays all the columns in the Google BigQuery table as a single field of the String data type in the Field Mapping tab.

The following image shows the STRING_DATA field in the Field Mapping tab of a synchronization task:
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:

- You cannot create a Google BigQuery target table that contains repeated columns using the Create Target option.
- If the Google BigQuery source table contains repeated columns, you cannot configure data filters for these columns.
- If the Google BigQuery table contains more than one repeated column, you cannot preview data.
- If the Google BigQuery target table contains repeated columns, you cannot configure update and delete operations for these columns.
- You cannot configure upsert operations for columns of the Record data type and repeated columns.
- 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:

- You cannot preview data.
- You cannot create a Google BigQuery target table using the Create Target option.
- If the Google BigQuery source table contains columns of the Record data type and repeated columns, you cannot configure data filters for these columns.
- 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 advanced target properties.
- You cannot use CSV format as the data format of the staging file. The following CSV formatting options in the advanced target 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:

- You cannot preview data.
- You cannot create a Google BigQuery target table using the Create Target option.
- When you configure a Google BigQuery source connection to use complex connection mode, you cannot configure data filters for the source.
- 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 advanced target properties.
• You cannot use CSV format as the data format of the staging file. The following CSV formatting options in the advanced target properties are not applicable:
  - Allow Quoted Newlines
  - Field Delimiter
  - Allow Jagged Rows
• You cannot use key range partitioning for Google BigQuery sources.

### Google BigQuery connection properties

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

The following table describes the Google BigQuery connection properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Name</td>
<td>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: `~ ! $ % ^ &amp; * ( ) - + = {[]}&quot;:;&quot;&lt;,&gt;,. , /</td>
</tr>
<tr>
<td>Description</td>
<td>Optional. The description of the connection. The description cannot exceed 4,000 characters.</td>
</tr>
<tr>
<td>Type</td>
<td>The Google BigQuery connection type.</td>
</tr>
<tr>
<td>Runtime Environment</td>
<td>Name of the runtime environment where you want to run the tasks.</td>
</tr>
<tr>
<td>Service Account ID</td>
<td>Specifies the client_email value present in the JSON file that you download after you create a service account.</td>
</tr>
<tr>
<td>Service Account Key</td>
<td>Specifies the private_key value present in the JSON file that you download after you create a service account.</td>
</tr>
</tbody>
</table>
| Connection mode     | The mode that you want to use to read data from or write data to Google BigQuery. Select one of the following connection modes:  
  - Simple. Flattens each field within the Record data type field as a separate field in the mapping.  
  - Hybrid. Displays all the top-level fields in the Google BigQuery table including Record data type fields. Google BigQuery Connector displays the top-level Record data type field as a single field of the String data type in the mapping.  
  - Complex. Displays all the columns in the Google BigQuery table as a single field of the String data type in the mapping.  
  Default is Simple. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema Definition File Path</td>
<td>Specifies a directory on the Secure Agent machine where the Secure Agent 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. Alternatively, you can specify a storage path in Google Cloud Storage where the Secure Agent 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. The schema definition file is required if you configure complex connection mode in the following scenarios: - You add a Hierarchy Builder transformation in a mapping to read data from relational sources and write data to a Google BigQuery target. - You add a Hierarchy Parser transformation in a mapping to read data from a Google BigQuery source and write data to relational targets.</td>
</tr>
<tr>
<td>Project ID</td>
<td>Specifies the project_id value present in the JSON file that you download after you create a service account. 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.</td>
</tr>
<tr>
<td>Dataset ID</td>
<td>Name of the dataset that contains the source table and target table that you want to connect to.</td>
</tr>
<tr>
<td>Storage Path</td>
<td>This property applies when you read or write large volumes of data. Path in Google Cloud Storage where the Secure Agent creates a local stage file to store the data temporarily. You can either enter the bucket name or the bucket name and folder name. For example, enter gs://&lt;bucket_name&gt; or gs://&lt;bucket_name&gt;/&lt;folder_name&gt;</td>
</tr>
</tbody>
</table>
CHAPTER 3

Google BigQuery Pushdown Optimization

This chapter includes the following topics:

- Pushdown Optimization, 18
- Pushdown Optimization Functions and Operators, 18
- Pushdown Optimization Transformations and Data Types, 20
- Configuring a Google BigQuery ODBC Connection, 21
- Create an ODBC Connection, 27
- Configuring the Secure Agent for Pushdown Optimization, 29

Pushdown Optimization

You can use pushdown optimization to push transformation logic to source databases or target databases. Use pushdown optimization when you use database resources to improve the performance of the task.

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.

You can configure Full and Source pushdown optimization for the ODBC connection type that uses Google BigQuery ODBC drivers for mapping.

**Note:** When you use pushdown optimization to write data to a Google BigQuery target, you cannot perform update, upsert, or delete operation on a Google BigQuery target.

Pushdown Optimization Functions and Operators

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Pushdown</th>
<th>Function</th>
<th>Pushdown</th>
<th>Function</th>
<th>Pushdown</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT()</td>
<td>-</td>
<td>INITCAP()</td>
<td>-</td>
<td>REG_MATCH()</td>
<td>-</td>
</tr>
<tr>
<td>ABS()</td>
<td>X</td>
<td>INSTR()</td>
<td>-</td>
<td>REG_REPLACE</td>
<td>-</td>
</tr>
<tr>
<td>ADD_TO_DATE()</td>
<td>-</td>
<td>IS_DATE()</td>
<td>-</td>
<td>REPLACECHR()</td>
<td>X</td>
</tr>
<tr>
<td>AES_DECRYPT()</td>
<td>-</td>
<td>IS_NUMBER()</td>
<td>-</td>
<td>REPLACESTR()</td>
<td>X</td>
</tr>
<tr>
<td>AES_ENCRYPT()</td>
<td>-</td>
<td>IS_SPACES()</td>
<td>-</td>
<td>REVERSE()</td>
<td>-</td>
</tr>
<tr>
<td>ASCII()</td>
<td>-</td>
<td>ISNULL()</td>
<td>X</td>
<td>ROUND()</td>
<td>-</td>
</tr>
<tr>
<td>AVG()</td>
<td>X</td>
<td>LAST()</td>
<td>-</td>
<td>ROUND()</td>
<td>X</td>
</tr>
<tr>
<td>CEIL()</td>
<td>X</td>
<td>LAST_DAY()</td>
<td>-</td>
<td>RPAD()</td>
<td>-</td>
</tr>
<tr>
<td>CHOOSE()</td>
<td>-</td>
<td>LEAST()</td>
<td>-</td>
<td>RTRIM()</td>
<td>X</td>
</tr>
<tr>
<td>CHR()</td>
<td>-</td>
<td>LENGTH()</td>
<td>X</td>
<td>SET_DATE_PART()</td>
<td>-</td>
</tr>
<tr>
<td>CHRCODE()</td>
<td>-</td>
<td>LN()</td>
<td>-</td>
<td>SIGN()</td>
<td>-</td>
</tr>
<tr>
<td>COMPRESS()</td>
<td>-</td>
<td>LOG()</td>
<td>-</td>
<td>SIN()</td>
<td>X</td>
</tr>
<tr>
<td>CONCAT()</td>
<td>X</td>
<td>LOOKUP</td>
<td>X</td>
<td>SINH()</td>
<td>-</td>
</tr>
<tr>
<td>COS()</td>
<td>X</td>
<td>LOWER()</td>
<td>X</td>
<td>SOUNDEX()</td>
<td>-</td>
</tr>
<tr>
<td>COSH()</td>
<td>-</td>
<td>LPAD()</td>
<td>-</td>
<td>SQRT()</td>
<td>X</td>
</tr>
<tr>
<td>COUNT()</td>
<td>X</td>
<td>LTRIM()</td>
<td>X</td>
<td>STDDEV()</td>
<td>-</td>
</tr>
<tr>
<td>CRC32()</td>
<td>-</td>
<td>MAKE_DATE_TIME()</td>
<td>-</td>
<td>SUBSTR()</td>
<td>X</td>
</tr>
<tr>
<td>CUME()</td>
<td>-</td>
<td>MAX()</td>
<td>X</td>
<td>SUM()</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Supports Number data type</td>
<td></td>
</tr>
<tr>
<td>DATE_COMPARE()</td>
<td>X</td>
<td>MD5()</td>
<td>-</td>
<td>SYSDATE()</td>
<td>-</td>
</tr>
<tr>
<td>DATE_DIFF()</td>
<td>-</td>
<td>MEDIAN()</td>
<td>-</td>
<td>SYSTIMESTAMP()</td>
<td>-</td>
</tr>
<tr>
<td>DECODE()</td>
<td>-</td>
<td>METAPHONE()</td>
<td>-</td>
<td>TAN()</td>
<td>X</td>
</tr>
<tr>
<td>DECODE_BASE64()</td>
<td>-</td>
<td>MIN()</td>
<td>X</td>
<td>TANH()</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Supports Date, Number, and String data type</td>
<td></td>
</tr>
<tr>
<td>DECOMPRESS()</td>
<td>-</td>
<td>MOD()</td>
<td>X</td>
<td>TO_BIGINT</td>
<td>X</td>
</tr>
</tbody>
</table>
The following list summarizes the availability of pushdown operators in a Google BigQuery database:

+, *, /, %, ||, >, =, >=, <=, !=, AND, OR, NOT, ^=.

### Pushdown Optimization Transformations and Data Types

When you use pushdown optimization, the Secure Agent converts the expression in the transformation by determining equivalent operators and functions in the database.

The following table lists the transformation logic that the Secure Agent can push to a Google BigQuery source or target:

<table>
<thead>
<tr>
<th>Transformations Supported</th>
<th>Pushdown Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregator</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Expression</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Filter</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Joiner</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Lookup</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Sorter</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Union</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Transformations Supported</td>
<td>Pushdown Type</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Router</td>
<td>Source, Full</td>
</tr>
<tr>
<td>Update Strategy</td>
<td>Source, Full</td>
</tr>
</tbody>
</table>

The following table lists the Google BigQuery data types that can be used for pushdown optimization:

<table>
<thead>
<tr>
<th>Google BigQuery Data Type</th>
<th>Transformation Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>Date.</td>
</tr>
<tr>
<td></td>
<td>Change the Google BigQuery data type to varchar. Set precision as 10.</td>
</tr>
<tr>
<td>DATETIME</td>
<td>DateTime</td>
</tr>
<tr>
<td></td>
<td>Change the Google BigQuery data type to varchar. Set precision as 30.</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Double</td>
</tr>
<tr>
<td>INT</td>
<td>BigInt</td>
</tr>
<tr>
<td></td>
<td>Change the Google BigQuery data type to Integer.</td>
</tr>
<tr>
<td>STRING</td>
<td>String</td>
</tr>
</tbody>
</table>

**Configuring a Google BigQuery ODBC Connection**

You can set the pushdown optimization for the ODBC connection type that uses Google BigQuery ODBC driver to enhance the mapping performance. To use an ODBC connection to connect to Google BigQuery, you must configure the ODBC connection.

After you create a Google BigQuery ODBC connection, add the **Pushdown Optimization** property under **Advanced Session Properties** tab when you create a mapping task and specify **Full** or **Source** in the **Session Property Value** field. You cannot configure target-side pushdown optimization by using Google BigQuery ODBC driver. To verify that the pushdown optimization has taken place, you can check the session log for the job. In **Monitor**, view the log for jobs.

Google BigQuery supports Google BigQuery ODBC drivers on Windows and Linux systems. You must install the Simba ODBC Driver for Google BigQuery 64-bit driver based on your system requirement.

**Configuring a Google BigQuery ODBC Connection on Windows**

To establish an ODBC connection to connect to Google BigQuery on Windows, you must download the 64-bit Google BigQuery ODBC driver on the machine where Secure Agent runs.

For more information about installing the Simba ODBC Driver for Google BigQuery on Windows, see the **Simba ODBC Driver for Google BigQuery Install and Configuration Guide** located in the following directory:

`<Driver installation directory>\Simba ODBC Driver for Google BigQuery`

Perform the following steps to configure an ODBC connection on Windows:

1. Click **Start** > **Control Panel**.
2. Click **Administrative Tools**.

3. Click **Data Sources (ODBC)**.
   
   The **ODBC Data Source Administrator** dialog box appears.

4. Click the **System DSN** tab.
   
   The **System DSN** tab appears. The following image shows the System DSN tab on the ODBC Data Source Administrator (64-bit) dialog box:

   ![ODBC Data Source Administrator (64-bit) dialog box](image)

5. Click **Add**.
The Create New Data Source dialog appears. The following image shows the Create New Data Source dialog where you can select the Google BigQuery data source:

6. Select the Simba ODBC Driver for Google BigQuery to set up the data source.

7. Click Finish.

The Simba ODBC Driver for Google BigQuery DSN Setup dialog box appears.
The following image shows the connection properties in the Simba ODBC Driver for Google BigQuery DSN Setup dialog box:
8. Specify the following connection properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source Name</td>
<td>Enter a name for the data source. The ODBC Driver for Google BigQuery uses the DSN to connect to the Google BigQuery.</td>
</tr>
<tr>
<td>Description</td>
<td>Enter a description.</td>
</tr>
<tr>
<td>OAuth Mechanism</td>
<td>The OAuth 2.0 authentication mechanism used to authenticate the driver. Select the Service Authentication option to authenticate the driver through a Google service account. If you select the User Authentication option, you will need to sign in to your Google service account. Click on Sign In and in the browser that opens, enter your credentials and then click Sign In. Click Accept to allow the Google BigQuery Client Tools to access Google BigQuery objects.</td>
</tr>
<tr>
<td>Confirmation Code</td>
<td>Code that Google provides when you click Accept. Copy and paste the code in the Confirmation Code field.</td>
</tr>
<tr>
<td>Refresh Token</td>
<td>The OAuth Mechanism populates the Refresh Token field when you paste the confirmation code.</td>
</tr>
<tr>
<td>Email</td>
<td>Specify the Google service account email id. This field is needed to authenticate the service account.</td>
</tr>
<tr>
<td>Key File Path</td>
<td>Enter the path to the .p12 or JSON key file that is used to authenticate the Google service account.</td>
</tr>
<tr>
<td>Request Google Drive scope access</td>
<td>Allows the driver to access Google Drive so that the driver can support federated tables that combine Google BigQuery data with data from Google Drive.</td>
</tr>
</tbody>
</table>
| Trusted Certificates      | Path of the .pem file. Use the trusted CA certificates from a specific .pem file or use the trusted CA certificates .pem file that is installed with the driver.  
**Note:** If you specify the Trusted Certificates .pem file path, you do not need to select the Use System Trust Store option. |
| Use System Trust Store    | If you select the Use System Trust Store option, you do not need to specify the Trusted Certificates .pem file path.                         |
| Catalog (Project)         | Name of the Google BigQuery project associated with your billing account that the Simba ODBC Driver for Google BigQuery queries against. |

9. Click **Proxy Options** to configure the connection to a data source through a proxy server.

10. Click **Advanced Options** to configure the Simba ODBC Driver for Google BigQuery advanced properties. To perform pushdown optimization, select **Standard SQL** from the Language Dialect option in the Advanced Options dialog box.

11. To enable logging, click **Logging Options**.

12. Click **Test** to verify the connection to Google BigQuery.

13. Click **OK** to close the Simba ODBC Driver for Google BigQuery DSN Setup dialog box.

14. Click **OK** to close the ODBC Data Source Administrator dialog box.

The Google BigQuery ODBC connection is configured successfully on Windows.
After you configure the Google BigQuery ODBC connection, you must create an ODBC connection to connect to Google BigQuery. For more information about how to create an ODBC connection to connect to Google BigQuery, see "Create an ODBC Connection" on page 27.

Configuring a Google BigQuery ODBC Connection on Linux

To establish an ODBC connection to connect to Google BigQuery on Linux, install the Google BigQuery ODBC (64-bit) driver on the Linux machine where the Secure Agent runs and configure the ODBC connection.

For more information about installing the Simba ODBC Driver for Google BigQuery on Linux, see the Simba ODBC Driver for Google BigQuery Install and Configuration Guide located in the following directory:

   <Driver installation directory>\simba\googlebigqueryodbc

Perform the following steps to configure an ODBC connection on Linux:

1. Create the odbc.ini file and add the following properties:

   [ODBC Data Sources]
   Sample DSN=Simba ODBC Driver for Google BigQuery 64-bit
   Description=<DSN Description>
   [Sample DSN]
   Driver=<Driver installation directory>/simba/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
   Catalog=<project_id>
   DefaultDataset=<Dataset name>
   SQLDialect=<1>
   OAuthMechanism=0
   Email=<Client Email ID of the Google service account>
   KeyFilePath=<.p12 or JSON key file path>

   For example,

   [ODBC Data Sources]
   GBQ_ODBC=Simba ODBC Driver for Google BigQuery 64-bit
   [GBQ_ODBC]
   Driver=/export/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
   Catalog=api-project=1243343
   DefaultDataset=QATEST
   SQLDialect=1
   OAuthMechanism=0
   Email=simbaccount@api-project-1243343.iam.gserviceaccount.com
   KeyFilePath=/export/Simba_GBQ_ODBC/API Project=c993e990af5.json

2. Specify the following properties in the odbc.ini file:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODBC Data Sources</td>
<td>Name of the data source.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the data source.</td>
</tr>
<tr>
<td>Driver</td>
<td>Full path of the Simba ODBC Driver for Google BigQuery library file.</td>
</tr>
<tr>
<td>Catalog</td>
<td>Name of the Google BigQuery project associated with your billing account that</td>
</tr>
</tbody>
</table>
<pre><code>                 | the Simba ODBC Driver for Google BigQuery queries against.                  |
</code></pre>
<p>| SQLDialect       | The SQL dialect used to run queries against the Google BigQuery tables using |
| the DSN. To perform pushdown optimization, specify the value of SQLDialect   |
| property as 1.                                                               |
| DefaultDataset   | Specify a dataset name in Google BigQuery that the Simba ODBC driver queries |</p>
**Property** | **Description**
--- | ---
OAuthMechanism | The OAuth 2.0 authentication mechanism used to authenticate the driver. To authenticate the driver through a Google service account, specify the value of OAuthMechanism property as 0. Specify the Email To authenticate the driver through a Google user account, specify the value of OAuthMechanism property as 1. Obtain a Refresh Token based on your Google user account and set the following property in the DSN: Auth_RefreshToken=<Refresh token value> If you already have your refresh token, then you can set the following property in the DSN: RefreshToken=<Refresh token value>
Email | Applicable when you set the value of OAuthMechanism property as 0. Specify the value of the Email property to the Google service account email ID.
KeyFilePath | Applicable when you set the value of OAuthMechanism property as 0. Enter the path to the .p12 or JSON key file that is used to authenticate the Google service account.
RefreshToken | Applicable when you set the value of OAuthMechanism property as 1. Specify the refresh token associated with the Google user account.

3. Run the following command to set the environment variable ODBCINI:

   ```bash
   Export ODBCINI=/odbci.ini file path>/odbc.ini
   ```

4. Restart the Secure Agent.

   The Google BigQuery ODBC connection on Linux is configured successfully.

   After you configure the Google BigQuery ODBC connection, you must create an ODBC connection to connect to Google BigQuery.

   For more information about how to create an ODBC connection to connect to Google BigQuery, see "Create an ODBC Connection" on page 27.

---

**Create an ODBC Connection**

You must create an ODBC connection to connect to Google BigQuery after you configure the ODBC connection.

Perform the following steps to create a Google BigQuery ODBC connection on the Connections page:

1. In Administrator, click Connections.
   The Connections page appears.
2. Click New Connection.
The New Connection page appears. The following image shows the New Connection page:

### Connection Details
- **Connection Name**: Test
- **Description**: 
- **Type**: ODBC

### ODBC Connection Properties
- **Runtime Environment**: Select...
- **User Name**: dummy
- **Password**: ● ● ● ● ●
- **Data Source Name**: dummy
- **Schema**: 
- **Code Page**: UTF-8
- **ODBC Subtype**: Google BigQuery
- **Driver Manager for Linux**: Data Direct

3. Configure the following connection details in the **Connection Details** section:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Name</td>
<td>Name of the ODBC connection. For example, sf_odbc.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the connection.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the connection. Select the type of the connection as <strong>ODBC</strong>.</td>
</tr>
</tbody>
</table>

4. Configure the following connection details in the **ODBC Connection Properties** section:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime Environment</td>
<td>Runtime environment that contains the Secure Agent you can use to access the system.</td>
</tr>
<tr>
<td>User Name</td>
<td>Username to log in to the Google BigQuery database.</td>
</tr>
<tr>
<td>Password</td>
<td>Password to log in to the Google BigQuery database.</td>
</tr>
<tr>
<td>Data Source Name</td>
<td>Enter the name of the ODBC data source name that you created for the Google BigQuery database.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Schema</td>
<td>Name of the Google BigQuery schema.</td>
</tr>
<tr>
<td>Code Page</td>
<td>The code page of the database server or flat file defined in the connection.</td>
</tr>
<tr>
<td>ODBC Subtype</td>
<td>Select Google BigQuery.</td>
</tr>
<tr>
<td>Driver Manager for Linux</td>
<td>The driver that the Google BigQuery ODBC driver manager sends database calls to. Select Data Direct to connect to Google BigQuery.</td>
</tr>
</tbody>
</table>

The Google BigQuery ODBC connection is created successfully.

Configuring the Secure Agent for Pushdown Optimization

Before you configure pushdown optimization, you must add the AutoCommitOff flag in the Custom Configuration Details for the Secure Agent ans set the value to Yes.

To add the AutoCommitOff flag, perform the following steps:

1. In Administrator, select Runtime Environments.
   The Runtime Environments page appears.
2. Select the Secure Agent for which you want to set the AutoCommitOff property from the list of available Secure Agents.
3. In the upper-right corner, click Edit.
4. In the Custom Configuration Details section, select the Service as Data Integration Server and Type as DTM.
5. Add AutoCommitOff in the Name field and set the Value as Yes.

The following image shows the Custom Configuration Details section:
Chapter 4

Synchronization Tasks with Google BigQuery Connector

This chapter includes the following topics:

- Pre SQL and Post SQL Commands, 30
- Google BigQuery Sources in Synchronization Tasks, 31
- Read Modes, 31
- Advanced Properties for Google BigQuery Sources, 32
- Data Filters, 34
- Google BigQuery Targets in Synchronization Tasks, 35
- Write Modes, 35
- Advanced Synchronization Task Options for Google BigQuery Targets, 36
- Advanced Properties for Google BigQuery Targets, 36
- Upsert Task Operation, 39

Pre SQL and Post SQL Commands

You can specify pre SQL and post SQL advanced properties for Google BigQuery sources and targets. When you create a task in Data Integration, you can specify SQL commands in the advanced properties for a source and target.

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

- SELECT
- UPDATE
- DELETE

Note: You cannot perform more than one operation with a pre SQL or post SQL command.

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 advanced properties for Google BigQuery sources and targets.

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:

<table>
<thead>
<tr>
<th>Options</th>
<th>Supported Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DestinationDataset</td>
<td>Dataset ID in Google BigQuery</td>
</tr>
<tr>
<td>DestinationTable</td>
<td>Table name in Google BigQuery</td>
</tr>
<tr>
<td>FlattenResults</td>
<td>True and False</td>
</tr>
<tr>
<td>UseLegacySQL</td>
<td>True and False</td>
</tr>
<tr>
<td>WriteDisposition</td>
<td>WRITE_TRUNCATE, WRITE_APPEND, and WRITE_EMPTY</td>
</tr>
</tbody>
</table>

**Note:** 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`

### Google BigQuery Sources in Synchronization Tasks

You can use a single object in a synchronization task.

You can configure the Google BigQuery source properties on the **Source** page of the Synchronization Task wizard.

The following table describes the Google BigQuery source properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Name of the active Google BigQuery source connection.</td>
</tr>
<tr>
<td>Source Type</td>
<td>Type of the Google BigQuery source objects available. You can read data from a single Google BigQuery source object.</td>
</tr>
<tr>
<td>Source Object</td>
<td>Name of the Google BigQuery source object.</td>
</tr>
<tr>
<td>Display technical names instead of labels</td>
<td>This property is not applicable for Google BigQuery Connector because both the technical names and labels are the same for Google.</td>
</tr>
<tr>
<td>Display source fields in alphabetical order</td>
<td>Displays source fields in alphabetical order. By default, fields appear in the order returned by the source system.</td>
</tr>
</tbody>
</table>

### Read Modes

When you use Google BigQuery Connector, 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, Google
BigQuery Connector directly reads data from a Google BigQuery source. You can configure the number
of rows that you want Google BigQuery Connector 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, Google BigQuery Connector first exports the data from the Google BigQuery source into
Google Cloud Storage. After the export is complete, Google BigQuery Connector 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. Google BigQuery Connector then reads the data from the local stage file.

When you enable staging file compression, Google BigQuery Connector compresses the size of the
staging file in Google Cloud Storage. Google BigQuery Connector 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.

Advanced Properties for Google BigQuery Sources

You can configure advanced source properties on the Schedule page of the Synchronization Task wizard.

The following table describes the advanced properties that you can configure for a Google BigQuery source:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Dataset ID</td>
<td>Optional. Overrides the Google BigQuery dataset name that you specified in the connection.</td>
</tr>
<tr>
<td>Number of Rows to Read</td>
<td>Specifies the number of rows to read from the Google BigQuery source table.</td>
</tr>
<tr>
<td>Allow Large Results</td>
<td>Determines whether Google BigQuery Connector 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.</td>
</tr>
<tr>
<td>Query Results Table Name</td>
<td>Required if you select the Allow Large Results option. Specifies the destination table name to store the query results. If the table is not present in the dataset, Google BigQuery Connector creates the destination table with the name that you specify.</td>
</tr>
<tr>
<td>Job Poll Interval in Seconds</td>
<td>The number of seconds after which Google BigQuery Connector polls the status of the read job operation. Default is 10.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Read Mode** | Specifies the read mode to read data from the Google BigQuery source. You can select one the following read modes:  
  - Direct. In direct mode, Google BigQuery Connector reads data directly from the Google BigQuery source table.  
  - Staging. In staging mode, Google BigQuery Connector exports data from the Google BigQuery source into Google Cloud Storage. After the export is complete, Google BigQuery Connector downloads the data from Google Cloud Storage into the local stage file and then reads data from the local stage file.  
  **Note:** When you use hybrid and complex connection mode, you cannot use direct mode to read data from the Google BigQuery source. Default is Direct mode. |
| **Number of Threads for Downloading Staging Files** | Specifies the number of files that Google BigQuery Connector downloads at a time to enable parallel download. This property applies to staging mode. |
| **Local Stage File Directory** | Specifies the directory on your local machine where Google BigQuery Connector stores the Google BigQuery source data temporarily before it reads the data. This property applies to staging mode. |
| **Staging File Name** | Name of the staging file where data from the Google BigQuery source table is exported to Google Cloud Storage. This property applies to staging mode. |
| **Enable Staging File Compression** | Indicates whether to compress the size of the staging file in Google Cloud Storage before Google BigQuery Connector reads data from the staging file. You can enable staging file compression to reduce cost and transfer time. This property applies to staging mode. |
| **Persist Destination Table** | Indicates whether Google BigQuery Connector must persist the query results table after it reads data from the query results table. By default, Google BigQuery Connector deletes the query results table. |
| **pre SQL** | SQL statement that you want to run before reading data from the source. 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:  
  ```sql  
  SELECT * FROM [api-project-80697026669:EMPLOYEE.DEPARTMENT] LIMIT 1000;  
  ```  
| **pre SQL Configuration** | Specify a pre SQL configuration. For example,  
  ```java  
  DestinationTable:PRESQL_SRC,DestinationDataset:EMPLOYEE, FlattenResults:False,WriteDisposition:WRITE_TRUNCATE,UseLegacySql:False  
  ```  

Advanced Properties for Google BigQuery Sources   33
**Data Filters**

You can create simple or advanced data filters. You can also create a set of data filters for each object included in a synchronization task. Each set of data filters act independently of the other sets.

You can create simple or advanced data filters for the following data types:

- Integer
- Float
- String
- Timestamp

### Simple Data Filters

You can create one or more simple data filters. When you create multiple simple data filters, the associated task creates an AND operator between the filters and loads rows that apply to all simple data filters.

### Advanced Data Filters

You can create an advanced data filter to create complex expressions 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.

**Note:** Google BigQuery Connector does not support the $LastRunDate variable under advanced data filters.
Google BigQuery Targets in Synchronization Tasks

You can use a single Google BigQuery object as a target in a synchronization task.

The following table describes the Google BigQuery target properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Name of the active Google BigQuery target connection that is associated with a dataset.</td>
</tr>
<tr>
<td>Target Object</td>
<td>You can select an existing object from the list or create a target at run time.</td>
</tr>
<tr>
<td>Child Object</td>
<td>This property is not applicable for Google BigQuery Connector.</td>
</tr>
<tr>
<td>Display technical names instead of labels</td>
<td>This property is not applicable for Google BigQuery Connector because both the technical names and labels are the same for Google.</td>
</tr>
<tr>
<td>Display target fields in alphabetical order</td>
<td>Displays target fields in alphabetical order. By default, fields appear in the order returned by the target system.</td>
</tr>
</tbody>
</table>

Write Modes

When you use Google BigQuery Connector, 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, Google BigQuery Connector first writes the data to a staging file in Google Cloud Storage. When the staging file contains all the data, Google BigQuery Connector loads the data from the staging file to the BigQuery target.

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

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

Google BigQuery Connector 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, Google BigQuery Connector directly writes data to the BigQuery target. Google BigQuery Connector appends the data into the BigQuery target.

You can configure the number of rows that you want Google BigQuery Connector 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.

### Advanced Synchronization Task Options for Google BigQuery Targets

You can configure advanced task options for a Google BigQuery target on the Schedule page of the Synchronization Task wizard.

The following table describes the advanced task options that you can configure for a Google BigQuery target:

<table>
<thead>
<tr>
<th>Advanced Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter File Name</td>
<td>Name of the file that contains the definitions and values of user-defined parameters used in the task.</td>
</tr>
<tr>
<td>Maximum Number of Log Files</td>
<td>Number of session log files, error log files, and import log files to retain. By default, Data Integration stores each type of log file for 10 runs before it overwrites the log files for new runs.</td>
</tr>
<tr>
<td>Update Columns</td>
<td>Specifies the temporary primary key columns to update, upsert, or delete target data. If the Google BigQuery target does not include a primary key column, and the task performs an update, upsert, or delete task operation, click Add to add a temporary key. You can select multiple columns. By default, no columns are specified.</td>
</tr>
</tbody>
</table>

### Advanced Properties for Google BigQuery Targets

You can configure advanced target properties on the Schedule page of the Synchronization Task wizard.

The following table describes the advanced properties that you can configure for a Google BigQuery target:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Dataset ID</td>
<td>Optional. Overrides the Google BigQuery dataset name that you specified in the connection.</td>
</tr>
<tr>
<td>Target Table Name</td>
<td>Optional. Overrides the Google BigQuery target table name that you specified in the Target page of the synchronization task.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Create Disposition            | Specifies whether Google BigQuery Connector must create the target table if it does not exist. You can select one of the following values:  
- Create if needed. If the table does not exist, Google BigQuery Connector creates the table.  
- Create never. If the table does not exist, Google BigQuery Connector does not create the table and displays an error message.                                                                                                                                                                      |
| Write Disposition             | Specifies how Google BigQuery Connector must write data in bulk mode if the target table already exists. You can select one of the following values:  
- Write append. If the target table exists, Google BigQuery Connector appends the data to the existing data in the table.  
- Write truncate. If the target table exists, Google BigQuery Connector overwrites the existing data in the table.  
- Write empty. If the target table exists and contains data, Google BigQuery Connector displays an error and does not write the data to the target. Google BigQuery Connector writes the data to the target only if the target table does not contain any data.  
**Note:** Write disposition is applicable for bulk mode.  
**Note:** Write disposition is applicable only when you perform an insert operation on a Google BigQuery target.                                                                                                                                                          |
| Write Mode                    | Specifies the mode to write data to the Google BigQuery target. You can select one of the following modes:  
- Bulk. In bulk mode, Google BigQuery Connector first writes the data to a staging file in Google Cloud Storage. When the staging file contains all the data, Google BigQuery Connector loads the data from the staging file to the BigQuery target. Google BigQuery Connector then deletes the staging file unless you configure the task to persist the staging file.  
- Streaming. In streaming mode, Google BigQuery Connector directly writes data to the BigQuery target. Google BigQuery Connector writes the data into the target row by row.  
**Default is Bulk mode.**                                                                                                                                                                                                                                                   |
| Streaming Template Table Suffix | Specify the suffix to add to the individual target tables that Google BigQuery Connector creates based on the template target table.  
**This property applies to streaming mode.**                                                                                                                                                                                                                               |
| Rows per Streaming Request    | Specifies the number of rows that Google BigQuery Connector streams to the BigQuery target for each request.  
**Default is 500 rows.**  
The maximum row size that Google BigQuery Connector can stream to the Google BigQuery target for each request is 10 MB.  
**This property applies to streaming mode.**                                                                                                                                                                                                                           |
| Staging file name             | Name of the staging file that Google BigQuery Connector creates in the Google Cloud Storage before it loads the data to the Google BigQuery target.  
**This property applies to bulk mode.**                                                                                                                                                                                                                                     |
| Data Format of the staging file | Specifies the data format of the staging file. You can select one of the following data formats:  
- JSON (Newline Delimited). Supports flat and record data with nested and repeated fields.  
- CSV. Supports flat data.  
**Note:** In a .csv file, columns of the Timestamp data type are represented as floating point numbers that cause the milliseconds value to differ.                                                                                                                                                                                                 |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persist Staging File After Loading</td>
<td>Indicates whether Google BigQuery Connector 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. By default, Google BigQuery Connector deletes the staging file in Google Cloud Storage. This property applies to bulk mode.</td>
</tr>
<tr>
<td>Enable Staging File Compression</td>
<td>Select this option to compress the size of the staging file before 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. You can enable staging file compression to reduce cost and transfer time.</td>
</tr>
<tr>
<td>Job Poll Interval in Seconds</td>
<td>The number of seconds after which Google BigQuery Connector polls the status of the write job operation. Default is 10.</td>
</tr>
<tr>
<td>Number of Threads for Uploading Staging file</td>
<td>The number of files that Google BigQuery Connector must create to upload the staging file in bulk mode.</td>
</tr>
<tr>
<td>Local Stage File Directory</td>
<td>Specifies the directory on your local machine where Google BigQuery Connector stores the files temporarily before writing the data to the staging file in Google Cloud Storage. This property applies to bulk mode.</td>
</tr>
<tr>
<td>Allow Quoted Newlines</td>
<td>Indicates whether Google BigQuery Connector must allow the quoted data sections with newline character in a .csv file.</td>
</tr>
<tr>
<td>Field Delimiter</td>
<td>Delimiter character for the fields in a .csv file.</td>
</tr>
<tr>
<td>Allow Jagged Rows</td>
<td>Indicates whether Google BigQuery Connector must accept the rows without trailing columns in a .csv file.</td>
</tr>
<tr>
<td>Pre SQL</td>
<td>SQL statement that you want to run before writing data to the target. 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: <code>SELECT * FROM </code>api-project-80697026669.EMPLOYEE.RegionNation<code> LIMIT 1000</code></td>
</tr>
<tr>
<td>Pre SQL Configuration</td>
<td>Specify a pre SQL configuration. For example, <code>DestinationTable:PRESQL_TGT2, DestinationDataset:EMPLOYEE, FlattenResults:False,WriteDisposition:WRITE_TRUNCATE,UseLegacySql:False</code></td>
</tr>
<tr>
<td>Post SQL</td>
<td>SQL statement that you want to run after writing the data into the target. 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: <code>UPDATE [api-project-80697026669.EMPLOYEE.PERSONS_TGT_DEL] SET phoneNumber.number =1000011, phoneNumber.areaCode=100 where fullName='John Doe'</code></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Post SQL Configuration</td>
<td>Specify a post SQL configuration. For example, DestinationTable:POSTSQL_SRC,DestinationDataset:EMPLOYEE, FlattenResults:True,UseLegacySQL:False</td>
</tr>
<tr>
<td>Success File Directory</td>
<td>Not applicable for Google BigQuery Connector.</td>
</tr>
<tr>
<td>Error File Directory</td>
<td>Not applicable for Google BigQuery Connector.</td>
</tr>
</tbody>
</table>

**Upsert Task Operation**

When you perform an upsert operation on a Google BigQuery target, you must configure the upsert fields for the target table. You can use an ID field for standard objects. Ensure that you include the upsert field in the field mappings for the task.

**Rules and Guidelines**

Consider the following rules and guidelines when you perform an upsert operation on a Google BigQuery target:

- You cannot use streaming mode to write data to a Google BigQuery target.
- When you configure a Google BigQuery connection to use simple or hybrid connection mode, you cannot configure upsert operations for columns of the Record data type and repeated columns.
- When you configure a Google BigQuery connection to use complex connection mode, you cannot configure an upsert operation.
This chapter includes the following topics:

- **Pre SQL and Post SQL Commands, 40**
- **Google BigQuery Sources in Mappings, 41**
- **Google BigQuery Targets in Mappings, 44**
- **Upsert Task Operation, 47**
- **Partitioning, 47**
- **Hierarchy Parser Transformation in Mappings, 49**
- **Hierarchy Builder Transformation in Mappings, 49**
- **Rules and Guidelines for Mapping and Mapping Tasks, 49**

### Pre SQL and Post SQL Commands

You can specify **pre SQL** and **post SQL** advanced properties for Google BigQuery sources and targets. When you create a task in Data Integration, you can specify SQL commands in the advanced properties for a source and target.

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

- **SELECT**
- **UPDATE**
- **DELETE**

**Note:** You cannot perform more than one operation with a pre SQL or post SQL command.

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** advanced properties for Google BigQuery sources and targets.

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:

<table>
<thead>
<tr>
<th>Options</th>
<th>Supported Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DestinationDataset</td>
<td>Dataset ID in Google BigQuery</td>
</tr>
<tr>
<td>DestinationTable</td>
<td>Table name in Google BigQuery</td>
</tr>
<tr>
<td>FlattenResults</td>
<td>True and False</td>
</tr>
<tr>
<td>UseLegacySQL</td>
<td>True and False</td>
</tr>
<tr>
<td>WriteDisposition</td>
<td>WRITE_TRUNCATE, WRITE_APPEND, and WRITE_EMPTY</td>
</tr>
</tbody>
</table>

**Note:** 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

---

**Google BigQuery Sources in Mappings**

To read data from Google BigQuery, configure a Google BigQuery object as the Source transformation in a mapping.

Specify the name and description of the Google BigQuery source. Configure the source, query options, and advanced properties for the source object.

The following table describes the source properties that you can configure for a Google BigQuery source:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Name of the active Google BigQuery source connection.</td>
</tr>
<tr>
<td>Source Type</td>
<td>Type of the Google BigQuery source objects available. You can read data from a single Google BigQuery source object or parameterize the object. You cannot read data from multiple objects.</td>
</tr>
<tr>
<td>Object</td>
<td>Name of the Google BigQuery source object based on the source type selected.</td>
</tr>
<tr>
<td>Filter</td>
<td>Configure a simple filter or an advanced filter to remove rows at the source. You can improve efficiency by filtering early in the data flow. A simple filter includes a field name, operator, and value. Use an advanced filter to define a more complex filter condition, which can include multiple conditions using the AND or OR logical operators.</td>
</tr>
</tbody>
</table>
The following table describes the advanced properties that you can configure for a Google BigQuery source:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Dataset ID</td>
<td>Optional. Overrides the Google BigQuery dataset name that you specified in the connection.</td>
</tr>
<tr>
<td>Number of Rows to Read</td>
<td>Specifies the number of rows to read from the Google BigQuery source table.</td>
</tr>
<tr>
<td>Allow Large Results</td>
<td>Determines whether Google BigQuery Connector 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.</td>
</tr>
<tr>
<td>Query Results Table Name</td>
<td>Required if you select the <strong>Allow Large Results</strong> option. Specifies the destination table name to store the query results. If the table is not present in the dataset, Google BigQuery Connector creates the destination table with the name that you specify.</td>
</tr>
<tr>
<td>Job Poll Interval in Seconds</td>
<td>The number of seconds after which Google BigQuery Connector polls the status of the read job operation. Default is 10.</td>
</tr>
<tr>
<td>Read Mode</td>
<td>Specifies the read mode to read data from the Google BigQuery source. You can select one of the following read modes:</td>
</tr>
<tr>
<td></td>
<td>- Direct. In direct mode, Google BigQuery Connector reads data directly from the Google BigQuery source table.</td>
</tr>
<tr>
<td></td>
<td>- Staging. In staging mode, Google BigQuery Connector exports data from the Google BigQuery source into Google Cloud Storage. After the export is complete, Google BigQuery Connector downloads the data from Google Cloud Storage into the local stage file and then reads data from the local stage file. Default is Direct mode.</td>
</tr>
<tr>
<td>Number of Threads for Downloading Staging Files</td>
<td>Specifies the number of files that Google BigQuery Connector downloads at a time to enable parallel download. This property applies to staging mode.</td>
</tr>
<tr>
<td>Local Stage File Directory</td>
<td>Specifies the directory on your local machine where Google BigQuery Connector stores the Google BigQuery source data temporarily before it reads the data. This property applies to staging mode.</td>
</tr>
<tr>
<td>Staging File Name</td>
<td>Name of the staging file where data from the Google BigQuery source table is exported to Google Cloud Storage. This property applies to staging mode.</td>
</tr>
<tr>
<td>Enable Staging File Compression</td>
<td>Indicates whether to compress the size of the staging file in Google Cloud Storage before Google BigQuery Connector reads data from the staging file. You can enable staging file compression to reduce cost and transfer time. This property applies to staging mode.</td>
</tr>
<tr>
<td>Persist Destination Table</td>
<td>Indicates whether Google BigQuery Connector must persist the query results table after it reads data from the query results table. By default, Google BigQuery Connector deletes the query results table.</td>
</tr>
</tbody>
</table>
You can set the tracing level in the advanced properties session to determine the amount of details that logs contain.

The following table describes the tracing levels that you can configure:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>terse</td>
<td>The Secure Agent logs initialization information, error messages, and notification of rejected data.</td>
</tr>
<tr>
<td>normal</td>
<td>The Secure Agent logs initialization and status information, errors encountered, and skipped rows due to transformation row errors. Summarizes session results, but not at the level of individual rows.</td>
</tr>
<tr>
<td>verbose init</td>
<td>In addition to normal tracing, the Secure Agent logs additional initialization details, names of index and data files used, and detailed transformation statistics.</td>
</tr>
<tr>
<td>verbose data</td>
<td>In addition to verbose initialization tracing, the Secure Agent logs each row that passes into the mapping. Also notes where the Secure Agent truncates string data to fit the precision of a column and provides detailed transformation statistics. When you configure the tracing level to verbose data, the Secure Agent writes row data for all rows in a block when it processes a transformation.</td>
</tr>
</tbody>
</table>
Google BigQuery Targets in Mappings

To write data to a Google BigQuery target, configure a Google BigQuery object as the Target transformation in a mapping.

Specify the name and description of Google BigQuery target. Configure the target and advanced properties for the target object.

The following table describes the target properties that you can configure for a Google BigQuery target:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Name of the active Google BigQuery connection that is associated with a dataset.</td>
</tr>
<tr>
<td>Target Type</td>
<td>Type of the Google BigQuery target objects available. You can write data to a single Google BigQuery target object or parameterize the object. You cannot write data to multiple objects.</td>
</tr>
<tr>
<td>Object</td>
<td>Name of the Google BigQuery target object based on the target type selected.</td>
</tr>
<tr>
<td>Create New at Runtime</td>
<td>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. The target name can contain alphanumeric characters. You can use the following special characters in the file name: ., _, @, $, % Google BigQuery Connector creates a new target table in the Dataset ID specified in the Google BigQuery connection.</td>
</tr>
<tr>
<td>Operation</td>
<td>You can select one the following operations: Insert, Update, Upsert (Update or Insert), Delete, Data Driven Note: If you use complex connection mode, you cannot configure update, upsert, and delete operations.</td>
</tr>
<tr>
<td>Update Columns</td>
<td>Specifies the temporary primary key columns to update, upsert or delete target data. If the Google BigQuery target does not include a primary key column, and the mapping performs an update, upsert, or delete task operation, click Add to add a temporary key. You can select multiple columns. By default, no columns are specified.</td>
</tr>
</tbody>
</table>

The following table describes the advanced properties that you can configure for a Google BigQuery target:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Dataset ID</td>
<td>Optional. Overrides the Google BigQuery dataset name that you specified in the connection.</td>
</tr>
<tr>
<td>Target Table Name</td>
<td>Optional. Overrides the Google BigQuery target table name that you specified in the Target page of the synchronization task.</td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| Create Disposition | Specifies whether Google BigQuery Connector must create the target table if it does not exist. You can select one of the following values:  
- Create if needed. If the table does not exist, Google BigQuery Connector creates the table.  
- Create never. If the table does not exist, Google BigQuery Connector does not create the table and displays an error message. |
| Write Disposition | Specifies how Google BigQuery Connector must write data in bulk mode if the target table already exists. You can select one of the following values:  
- Write append. If the target table exists, Google BigQuery Connector appends the data to the existing data in the table.  
- Write truncate. If the target table exists, Google BigQuery Connector overwrites the existing data in the table.  
- Write empty. If the target table exists and contains data, Google BigQuery Connector displays an error and does not write the data to the target. Google BigQuery Connector writes the data to the target only if the target table does not contain any data.  
**Note:** Write disposition is applicable for bulk mode.  
**Note:** Write disposition is applicable only when you perform an insert operation on a Google BigQuery target. |
| Write Mode | Specifies the mode to write data to the Google BigQuery target. You can select one of the following modes:  
- Bulk. In bulk mode, Google BigQuery Connector first writes the data to a staging file in Google Cloud Storage. When the staging file contains all the data, Google BigQuery Connector loads the data from the staging file to the BigQuery target. Google BigQuery Connector then deletes the staging file unless you configure the task to persist the staging file.  
- Streaming. In streaming mode, Google BigQuery Connector directly writes data to the BigQuery target. Google BigQuery Connector writes the data into the target row by row.  
Default is Bulk mode. |
| Streaming Template Table Suffix | Specify the suffix to add to the individual target tables that Google BigQuery Connector creates based on the template target table.  
This property applies to streaming mode. |
| Rows per Streaming Request | Specifies the number of rows that Google BigQuery Connector streams to the BigQuery target for each request. Default is 500 rows.  
The maximum row size that Google BigQuery Connector can stream to the Google BigQuery target for each request is 10 MB.  
This property applies to streaming mode. |
| Staging file name | Name of the staging file that Google BigQuery Connector creates in the Google Cloud Storage before it loads the data to the Google BigQuery target.  
This property applies to bulk mode. |
| Data Format of the staging file | Specifies the data format of the staging file. You can select one of the following data formats:  
- JSON (Newline Delimited). Supports flat and record data with nested and repeated fields.  
- CSV. Supports flat data.  
**Note:** In a .csv file, columns of the Timestamp data type are represented as floating point numbers that cause the milliseconds value to differ. |
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persist Staging File After Loading</td>
<td>Indicates whether Google BigQuery Connector 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. By default, Google BigQuery Connector deletes the staging file in Google Cloud Storage. This property applies to bulk mode.</td>
</tr>
<tr>
<td>Enable Staging File Compression</td>
<td>Select this option to compress the size of the staging file before 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. You can enable staging file compression to reduce cost and transfer time.</td>
</tr>
<tr>
<td>Job Poll Interval in Seconds</td>
<td>The number of seconds after which Google BigQuery Connector polls the status of the write job operation. Default is 10.</td>
</tr>
<tr>
<td>Number of Threads for Uploading Staging file</td>
<td>The number of files that Google BigQuery Connector must create to upload the staging file in bulk mode.</td>
</tr>
<tr>
<td>Local Stage File Directory</td>
<td>Specifies the directory on your local machine where Google BigQuery Connector stores the files temporarily before writing the data to the staging file in Google Cloud Storage. This property applies to bulk mode.</td>
</tr>
<tr>
<td>Allow Quoted Newlines</td>
<td>Indicates whether Google BigQuery Connector must allow the quoted data sections with newline character in a .csv file.</td>
</tr>
<tr>
<td>Field Delimiter</td>
<td>Indicates whether Google BigQuery Connector must allow field separators for the fields in a .csv file.</td>
</tr>
<tr>
<td>Allow Jagged Rows</td>
<td>Indicates whether Google BigQuery Connector must accept the rows without trailing columns in a .csv file.</td>
</tr>
</tbody>
</table>
| Pre SQL                                      | SQL statement that you want to run before writing data to the target. 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:  
`SELECT * FROM `api-project-80697026669.EMPLOYEE.RegionNation` LIMIT 1000` |
| Pre SQL Configuration                        | Specify a pre SQL configuration. For example,  
`DestinationTable:PRESQL_TGT2,DestinationDataset:EMPLOYEE,FlattenResults:False,WriteDisposition:WRITE_TRUNCATE,UseLegacySql:False` |
| Post SQL                                     | SQL statement that you want to run after writing the data into the target. 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:  
`UPDATE [api-project-80697026669.EMPLOYEE.PERSONS_TGT_DEL]`  
`SET phoneNumber.number =1000011, phoneNumber.areaCode=100 where`  
`fullname='John Doe'` |
### Upsert Task Operation

When you perform an upsert operation on a Google BigQuery target, you must configure the upsert fields for the target table. You can use an ID field for standard objects. Ensure that you include the upsert field in the field mappings for the task.

#### Rules and Guidelines

Consider the following rules and guidelines when you perform an upsert operation on a Google BigQuery target:

- You cannot use streaming mode to write data to a Google BigQuery target.
- You cannot configure key range partitioning with more than one partition key.
- When you configure a Google BigQuery connection to use simple or hybrid connection mode, you cannot configure upsert operations for columns of the Record data type and repeated columns.
- When you configure a Google BigQuery connection to use complex connection mode, you cannot configure an upsert operation.

### Partitioning

When you read data from a Google BigQuery source and use simple or hybrid connection mode, you can configure key range partitioning to optimize the mapping performance at run time.

#### Key Range Partitioning for Google BigQuery Sources

You can configure key range partitioning when you use a mapping task to read data from Google BigQuery sources and use simple or hybrid connection mode. With key range partitioning, the Secure Agent distributes...
rows of source data based on the fields that you define as partition keys. The Secure Agent compares the field value to the range values for each partition and sends rows to the appropriate partitions.

Use key range partitioning for columns that have an even distribution of data values. Otherwise, the partitions might have unequal size. For example, a column might have 10 rows between key values 1 and 1000 and the column might have 999 rows between key values 1001 and 2000. If the mapping includes multiple sources, use the same number of key ranges for each source.

When you define key range partitioning for a column, the Secure Agent reads the rows that are within the specified partition range. For example, if you configure two partitions for a column with the ranges as 10 through 20 and 30 through 40, the Secure Agent does not read the rows 20 through 30 because these rows are not within the specified partition range.

You can configure a partition key for fields of the following data types:

- String
- Integer
- Timestamp. Use the following format: YYYY-MM-DD HH24:MI:SS

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

You cannot use key range partitions when a mapping includes any of the following transformations:

- Web Services
- JSON to Relational

### Configuring Key Range Partitioning

Perform the following steps to configure key range partitioning for Google BigQuery sources:

1. In the Source Properties, click the **Partitions** tab.
2. Select the required partition key from the list.
3. Click **Add New Key Range** to define the number of partitions and the key ranges based on which the Secure Agent must partition data.

   Use a blank value for the start range to indicate the minimum value. Use a blank value for the end range to indicate the maximum value.

   The following image displays the **Partitions** tab:
Hierarchy Parser Transformation in Mappings

To preserve the hierarchical structure when you read data from Google BigQuery and write data to relational targets, you must use a Hierarchy Parser transformation.

The transformation processes JSON input from the source transformation and provides relational output to the target transformation. The Hierarchy Parser transformation converts hierarchical input based on the sample schema of the Google BigQuery table that you associate with the transformation and the way that you map the data.

For more information on using the Hierarchy Parser transformation with Google BigQuery Connector, see the Informatica How-To Library article, "Using a Hierarchical Parser Transformation with Google BigQuery Connector":


Hierarchy Builder Transformation in Mappings

When you read data from relational sources and write data to a Google BigQuery target, you must use a Hierarchy Builder transformation.

The transformation processes relational input from the upstream transformation and provides JSON output to the downstream transformation. The Hierarchy Builder transformation produces JSON output based on the sample schema of the Google BigQuery table that you associate with the transformation and the way that you map the data.

For more information on using the Hierarchy Builder transformation with Google BigQuery Connector, see the Informatica How-To Library article, "Using a Hierarchical Builder Transformation with Google BigQuery Connector":


Rules and Guidelines for Mapping and Mapping Tasks

Consider the following rules and guidelines for mapping and mapping tasks:

- When you write large datasets to a Google BigQuery target, increase the Java heap size in the JVM options for type DTM. Set JVMOption3 to -Xms1024m and JVMOption4 to -Xmx4096m in the System Configuration Details section of the Secure Agent and restart the Secure Agent.

- When you use the Hosted Agent as the runtime environment in a mapping task and use a Hierarchy Builder or Hierarchy Parser transformation in a mapping, you must specify a storage path in Google Cloud Storage in the Schema Definition File Path field under the connection properties. You can then download the sample schema definition file for the Google BigQuery table from the specified storage path in Google Cloud Storage to a local machine.
When you read JSON data from a MongoDB source table and write data to a column of Record data type in a Google BigQuery target table, you must specify an explicit value for columns that contain _id in the column name. Otherwise, the task fails with the following error:

```plaintext
[ERROR] The [LOAD] job failed with the error - [JSON parsing error in row starting at position 0:
```
Data Type Reference

This appendix includes the following topics:

- Data Type Reference Overview, 51
- Google BigQuery and Transformation Data Types, 52

Data Type Reference Overview

Data Integration uses the following data types in mappings, synchronization tasks, and mapping tasks with Google BigQuery:

**Google BigQuery native data types**

Google BigQuery 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 transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

When Data Integration reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When Data Integration 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 describes the data types that Data Integration supports for Google BigQuery sources and targets:

<table>
<thead>
<tr>
<th>Google BigQuery Data Type</th>
<th>Transformation Data Type</th>
<th>Range and Description for the Transformation Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOLEAN</td>
<td>String</td>
<td>1 to 104,857,600 characters</td>
</tr>
</tbody>
</table>
| DATE                     | Date/Time                | Date values. Google BigQuery Connector uses the following format: DD-MM-YYYY  
Minimum value: 1/1/1970  
Maximum value: 30/12/9999 |
| 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                            |
| STRING                   | String                   | 1 to 104,857,600 characters                            |
| BYTE                     | Byte                     | 1 to 104,857,600 bytes                                |
| TIME                     | Date/Time                | Time values.  
(precision to the nanosecond) |
| TIMESTAMP                | Date/Time                | Jan 1, 0001 A.D. to Dec 31, 9999 A.D.  
(precision to the nanosecond) |
INDEX

B
bulk mode
   staging file 35

C
Cloud Application Integration community
   URL 5
Cloud Data Integration web site
   URL 5
Cloud Developer community
   URL 5
configuring key range
   partitioning 48
   connections
   Google BigQuery 16

D
data filters
   advanced 34
   simple 34
Data Integration community
   URL 5

G
Google BigQuery
   connection properties 16
Google BigQuery connections
   overview 11
Google BigQuery connector
   administration 9
   overview 7
   supported task and object types 8
Google BigQuery data types
   mapping to transformation data types 52
   overview 51
Google BigQuery ODBC connection
   configuration on linux 26
   configuration on windows 21
   odbc.ini file 26
   system DSN 21
Google BigQuery ODBC driver 21
Google BigQuery sources
   synchronization task 31
Google BigQuery targets
   advanced options for synchronization task 36
   write disposition 36

H
hosted agent 7

I
Informatica Global Customer Support
   contact information 6

M
maintenance outages 6
mapping
   Google BigQuery sources 41
   Google BigQuery targets 44

P
Partitioning
   key range 48
post SQL commands
   entering 30, 40
pre SQL and post SQL
   entering 30, 40
pre SQL commands
   entering 30, 40
pushdown optimization
   activity log 21
   advanced session properties 21
   create an ODBC connection 27
   full pushdown 18
   ODBC subtype 27
   source pushdown 18
   target pushdown 18

S
status
   Informatica cloud services 6
   streaming mode
      creating template table 35
      synchronization
         Google BigQuery targets 36
         system status 6

T
trust site
   description 6
U
upgrades 6
upsert 39, 47

W
write modes
bulk 35
streaming 35