Informatica® PowerExchange for Google Cloud Storage

10.5

User Guide

Informatica PowerExchange for Google Cloud Storage User Guide
10.5
March 2021

Publication Date: 2021-03-23

Table of Contents

Preface

Use the *Informatica PowerExchange for Google Cloud Storage User Guide* to learn how to read from and write to Google Cloud Storage by using the Developer tool. Learn to create a Google Cloud Storage 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.

Informatica Documentation

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

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica 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.

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 Cloud Storage

This chapter includes the following topics:

PowerExchange for Google Cloud Storage Overview

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

You can use Google Cloud Storage objects as sources and targets in mappings. When you use Google Cloud Storage objects in mappings, you must configure properties specific to Google Cloud Storage.

You can validate and run Google Cloud Storage mappings on the Spark engine in the Hadoop environment. When you run a task or mapping, the PowerExchange for Google Cloud Storage uses the Google Cloud Storage API to perform the specified operation and reads data from or writes data to Google Cloud Storage buckets.

Example

You run the IT department of a major bank and are responsible for storing huge volumes of transaction files in a relational database. You want to store the data in another database to avoid data loss if the relational database fails.

You can use PowerExchange for Google Cloud Storage to upload huge volumes of transaction files to Google Cloud storage from any location and at any time. You can back up data in Google Cloud Storage for disaster recovery purposes and retrieve the data later, if needed.

You can configure a mapping in Informatica Developer to write data to Google Cloud Storage.

Introduction to Google Cloud Storage

Google Cloud Storage is a web service that allows global storage and retrieval of large volumes of data at any time.

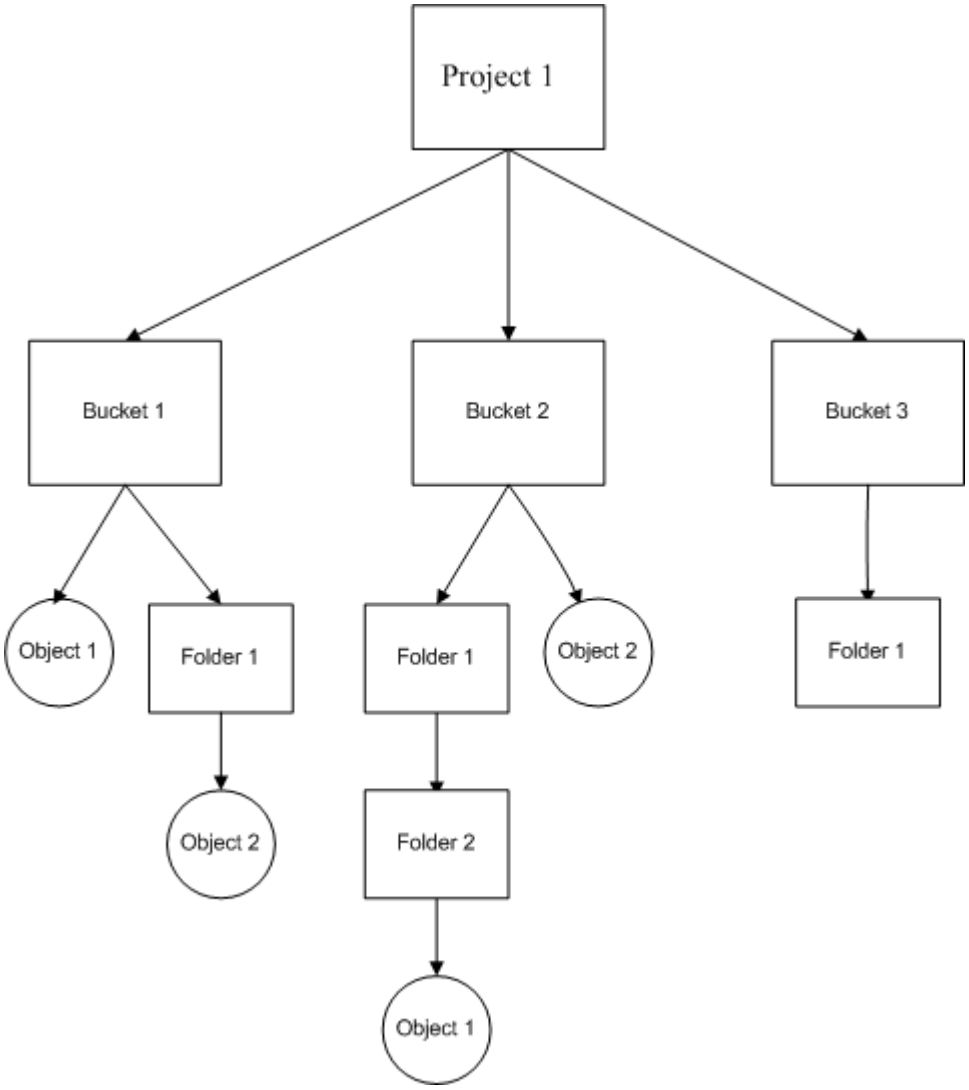
You can use Google Cloud Storage to stream multimedia, store custom data analytics pipelines, or distribute large data objects to users through direct download. You can write data to Google Cloud Storage for data backup. In the event of a database failure, you can read the data from Google Cloud Storage and restore it back to the database.

Google Cloud Storage offers different storage classes based on factors such as data availability, latency, and price.

Google Cloud Storage comprises the following components:

- Projects
- Buckets
- Objects

The following image shows how data can be organized in Google Cloud Storage:



You can use the following components to read data from and write to Google Cloud Storage:

Projects

In Google Cloud Storage, all resources are stored within a project. Project is a top-level container that stores billing details and user details. You can create multiple projects. A project has a unique project name, project ID, and project number.

Buckets

Each bucket acts like a container that stores data. You can use buckets to organize and access data. You can create more than one bucket but you cannot nest buckets.

You can create multiple folders within a bucket and you can also nest folders.

You can define access control lists to manage objects and buckets. An access control list consists of permission and scope entries. Permission defines the access to perform a read or write operation. Scope defines a user or a group who can perform the operation.

Objects

Objects comprise the data that you upload to Google Cloud Storage. You can create objects in a bucket. Objects consist of object data and object metadata components. The object data is a file that you store in Google Cloud Storage. The object metadata is a collection of name-value pairs that describe object qualities.

Google Cloud Storage File Formats

PowerExchange for Google Cloud Storage can read data from and write data to Google Cloud Storage in the following formats:

- Binary
- Flat
- JSON (newline-delimited)
- Avro
- Parquet

Note: When you run mappings on the Spark engine, you cannot read data from and write data to Google Cloud Storage in binary format.

CHAPTER 2

PowerExchange for Google Cloud Storage Configuration

This chapter includes the following topics:

PowerExchange for Google Cloud Storage Configuration Overview

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

Prerequisites

To use PowerExchange for Google Cloud Storage, perform the following steps:

1. Install and configure Informatica Services.
2. Install and configure the Developer tool. You can install the Developer tool when you install Informatica clients.
3. Create and configure a Model Repository Service, a Data Integration Service, and a Metadata Access Service in the Informatica domain.

   Note: Google Cloud Storage connection uses Metadata Access Service to import metadata from files in Google Cloud Storage.
4. Verify that you have write permissions on all the directories within the `<Informatica installation directory>` directory.
5. Ensure that the PowerExchange for Google Cloud Storage license is activated.
6. Ensure that you have a Google service account to access Google Cloud Storage.

7. Ensure that 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 Cloud Storage connection in the Developer tool.
8. Ensure that you have enabled the Google Cloud Storage JSON API for your service account. PowerExchange for Google Cloud Storage uses the Google API to integrate with Google Cloud Storage.
9. Ensure that you have a project, bucket, source file, and target file. You will need to enter the project_id, path, source file name, and target file name when you create data objects in the Developer tool.
10. Verify that you have read and write access to the Google Cloud Storage bucket that contains the source file and target file.

Configuring Environment Variables

After you install PowerExchange for Google Cloud Storage, configure environment variables for the Data Integration Service and the Metadata Access Service.

Configuring Environment Variables for the Metadata Access Service

To successfully import the metadata of the Avro or Parquet files, you must configure the INFA_PARSER_HOME environment variable for the Metadata Access Service in Informatica Administrator.

Perform the following steps to configure the INFA_PARSER_HOME property:

1. Log in to Informatica Administrator.
2. Click the Metadata Access Service and then click the **Processes** tab on the right pane.
3. Click **Edit** in the **Environment Variables** section.
4. Click **New** to add an environment variable.
5. Enter the name of the environment variable as INFA_PARSER_HOME.
6. Set the value of the INFA_PARSER_HOME environment variable to the following absolute path of the Hadoop distribution directory on the machine that runs the Metadata Access Service:
   `<Informatica installation directory/Informatica/services/shared/hadoop/<Hadoop distribution name>_<version>`

   For example, enter the following absolute path:

   `C:/Informatica/services/shared/hadoop/CDH_7.1`

Note: Verify that the Hadoop distribution version that you define in the INFA_PARSER_HOME property is the same as the version you defined in the cluster configuration.

On the machine where you have installed the Developer tool, verify that the Hadoop distribution version that you define in the -DINFA_HADOOP_DIST_DIR environment variable in the `developerCore.ini` file is the same as the version you defined in the INFA_PARSER_HOME property for the Metadata Access Service in the Administrator tool.

You can find `developerCore.ini` file in the following directory:

`<Informatica installation directory>\clients\DeveloperClient`

For example, enter the following path of the Hadoop distribution directory in the -DINFA_HADOOP_DIST_DIR environment variable in the `developerCore.ini` file:

```
-DINFA_HADOOP_DIST_DIR=hadoop\CDH_7.1
```

Configuring Environment Variables for the Data Integration Service

To successfully preview data from the Avro or Parquet files or run a mapping in Data Integration Service with the Avro or Parquet files, you must configure the INFA_PARSER_HOME environment variable for the Data Integration Service in Informatica Administrator.

Perform the following steps to configure the INFA_PARSER_HOME property:

1. Log in to Informatica Administrator.
2. Click the Data Integration Service and then click the **Processes** tab on the right pane.
3. Click **Edit** in the **Environment Variables** section.
4. Click **New** to add an environment variable.
5. Enter the name of the environment variable as INFA_PARSER_HOME.
6. Set the value of the INFA_PARSER_HOME environment variable to the following absolute path of the Hadoop distribution directory on the machine that runs the Data Integration Service:
   `<Informatica installation directory/Informatica/services/shared/hadoop/<Hadoop distribution name>_<version>`

   For example, enter the following absolute path:

   `C:/Informatica/services/shared/hadoop/CDH_6.1`

Note: Verify that the Hadoop distribution version that you define in the INFA_PARSER_HOME property is the same as the version you defined in the cluster configuration.

Configuring Java Heap Memory

When the Google Cloud Storage source or target file contains large amount of data, you must configure the memory for the Java heap size in the node that runs the Data Integration Service for the Google Cloud Storage mapping to run successfully.

1. In the Administrator tool, navigate to the Data Integration Service for which you want to change the Java heap size.
2. Click the **Processes** tab.
3. Edit the **Advanced Properties** section.
   The **Edit Advanced Properties** dialog box appears.
4. Specify the **Maximum Heap Size** limit based on the amount data that you want to process.
5. Click **OK**.
6. When you run a mapping in the native mode and specify a value greater than 1 for the **Maximum Parallelism** property, edit the **Custom Properties** section.
   The **Edit Custom Properties** dialog box appears.
7. Click **New** to add a new custom property.
8. Add the following custom properties and specify a value based on the amount of data that you want to process:

   ```
   ExecutionContextOptions.JVMMaxMemory = <size> MB
   ExecutionContextOptions.JVMMinMemory = <size> MB
   ```

   Where <size> is a valid heap size, such as 2048 MB.
9. Restart the Data Integration Service.

CHAPTER 3

Google Cloud Storage Connections

This chapter includes the following topics:

Google Cloud Storage Connections Overview

Use a Google Cloud Storage connection to access a Google Cloud Storage database.

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

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

Google Cloud Storage Connection Properties

When you set up a Google Cloud Storage 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 Cloud Storage connection properties:

Property	Description
Name	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:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select **Google Cloud Storage**.
Project ID	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 bucket that you want to connect to.
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.

Creating a Google Cloud Storage Connection

Create a Google Cloud Storage connection before you create a Google Cloud Storage 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 Cloud Storage**, and click **Add**.
5. Enter a connection name and an optional description.
6. Select **Google Cloud Storage** as the connection type.
7. Click **Next**.
8. Configure the connection properties.
9. Click **Test Connection** to verify the connection to Google Cloud Storage.
10. Click **Finish**.

CHAPTER 4

PowerExchange for Google Cloud Storage Data Objects

This chapter includes the following topics:

Google Cloud Storage Data Object Overview

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

Create a Google Cloud Storage data object in 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 Cloud Storage sources and loads data to Google Cloud Storage targets.

You can edit the advanced properties of the data object read or write operation and then add the operation to a mapping.

Google Cloud Storage 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 Cloud Storage data objects:

Property	Description
Name	Name of the Google Cloud Storage data object.
Location	The project or folder in the Model Repository Service where you want to store the Google Cloud Storage data object.
Resouce Format	You can create a Google Cloud Storage file data object from the following file formats: - Binary - Flat - Avro - Json - Parquet
Connection	Name of the Google Cloud Storage connection.

Google Cloud Storage Data Object Read Operation

Create a mapping to read data from Google Cloud Storage. Use the Google Cloud Storage connection, and define the read operation properties to read data from Google Cloud Storage.

Directory Source in Google Cloud Storage Sources

You can select the **Is Directory** option under the advanced properties for a Google Cloud Storage data object read operation to read all the files in a Google Cloud Storage folder.

PowerExchange for Google Cloud Storage provides the option to override the value of the **Google Cloud Storage Path** and **Source File Name** properties during run time. When you select the **Is Directory** option, the value of the **Source File Name** is not honored.

For read operation, if you provide the **Google Cloud Storage Path** value during run time, the Data Integration Service considers the value of the **Google Cloud Storage Path** from the data object read operation properties. If you do not provide the **Google Cloud Storage Path** value during run time, the Data Integration Service considers the value of the **Google Cloud Storage Path** that you specify during the data object creation.

Use the following rules and guidelines to configure the **Google Cloud Storage Path** property and **Is Directory** option:

- When you select the **Is Directory** option, PowerExchange for Google Cloud Storage does not read the files available in the sub-folder.
- You cannot specify wildcard characters in the **Google Cloud Storage Path** data object read operation property.
- All the source files in the directory must contain the same metadata.
- All the files must have data in the same format. For example, delimiters, header fields, and escape characters must be same.

Reading File Names for Google Cloud Storage Source Objects

You can use the data in the FileName port when you create a data object read operation.

When you import a Google Cloud Storage source object, the Data Integration Service creates a FileName port in the imported source transformation. The FileName port stores the name of the file from which the Data Integration Service reads the data at run-time. For example, a directory contains a number of files and each file contains multiple records that you want to read. You select the directory as source type in the Google Cloud Storage source mapping properties. When you run the mapping, the Data Integration Service reads each record and stores the respective file name in the FileName port.

When you run a mapping to read a Google Cloud Storage file using the FileName port, the Data Integration Service appends an underscore (_) to the directory name.

The FileName port is applicable to the following file formats:

- Flat file
- Avro (excluding hierarchical data types)
- Binary
- JSON (excluding hierarchical data types)
- Parquet (excluding hierarchical data types)

Rules and Guidelines for Using FileName Port

Use the following rules and guidelines when you run a mapping to read data using the FileName port:

- When you run a mapping to read a Google Cloud Storage file and if one of the values in the FileName port does not contain any value, the Data Integration Service creates the file in the following format:
  `<valueoftheNativeNamepropertyorFileNameDataObjectWriteOperation>_<fileextention>=<>`
  However, if you run a mapping to read the newly created file, the mapping fails with the following error message:

  ```
  java.lang.AssertionError: assertion failed: Empty partition column value in '< >'
  at scala.Predef$.assert(Predef.scala:170)
  ```

  You must ensure that all entries in the FileName port contains a value to read the newly created file successfully.
- Do not use a colon (:) and forward slash (/) character in the file name data of the FileName port of the source object to run a mapping.
- Do not connect FileName port to a FileName port because the FileName port in the source might contain colon (:) and forward slash (/) characters.

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 edit the port 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.

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

General Properties

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

Ports Properties

The output ports properties display the data types, precision, and scale of the data object read operation.

The following table describes the output ports properties that you configure in the data object read operation:

Property	Description
Name	Name of the port.
Type	Data type of the 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 for numeric values.
Description	Description of the port.

Sources Properties

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

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 Cloud Storage table.

Advanced Properties

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

The Developer tool displays advanced properties for the Google Cloud Storage data object operation in the Advanced view.
The following table describes the advanced properties for a Google Cloud Storage data object read operation:

Property	Description
Google Cloud Storage Path	Optional. Overrides the bucket name or folder path of the Google Cloud Storage file that you selected in the Google Cloud Storage data object. Use the following format: `gs://<bucket name>` or `gs://<bucket name>/<folder name>`
Source File Name	Optional. Overrides the Google Cloud Storage source file name specified in the Google Cloud Storage data object.
Is Directory	Select this property to read all the files available in the folder specified in the **Google Cloud Storage Path** property.

Google Cloud Storage Data Object Write Operation

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

You can perform insert operations on a Google Cloud Storage target. You cannot perform update, upsert, or delete operations on a Google Cloud Storage target.

Note: When you write data to a Google Storage file, you must create a sample file in Google Cloud Storage and you must ensure that the schema of the sample file matches the schema of the source file used in the mapping.

Input Properties of the Data Object Write Operation

Input properties represent the data that the Data Integration Service writes to a Google Cloud Storage target. Select the input properties to edit the port properties of the data object write operation. You can also specify advanced data object write operation properties to write data to the Google Cloud Storage target.

The input properties of the data object write operation include general properties that apply to the data object write operation. Input properties also include port, source, 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**, **Ports**, **Target**, **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 Cloud Storage file 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 Cloud Storage file.

Advanced Properties

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

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

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

Property	Description
Google Cloud Storage Path	Optional. Overrides the bucket name or folder path of the Google Cloud Storage file that you selected in the Google Cloud Storage data object. Use the following format: `gs://<bucket name>` or `gs://<bucket name>/<folder name>`
Target File Name	Optional. Overrides the Google Cloud Storage target file name specified in the Google Cloud Storage data object.
Target Schema Strategy	Not applicable for PowerExchange for Google Cloud Storage. Leave the default value of **RETAIN** unchanged.

Creating a Google Cloud Storage Data Object

Create a Google Cloud Storage 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 **GoogleStorage Data Object** and click **Next**.

   The **GoogleStorage 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. In the **Resource Format** list, select any of the following formats:
   - Binary. To read any type of resource.
   - Flat. To read a flat resource.
   - Avro. To read an Avro resource.
   - Json. To read a JSON resource.
   - Parquet. To read a Parquet resource.
7. Click **Browse** next to the **Connection** option and select the Google Storage connection from which you want to import the Google Cloud Storage file.
8. To add a resource, click **Add** next to the **Selected Resources** option.

   The **Add Resource** dialog box appears. Expand the Google Cloud Storage connection.

   Note: If a default Metadata Access Service is not set, a message appears to configure the Metadata Access Service. Click **OK** and set one Metadata Access Service as default. After you set a default Metadata Access Service, the **Add Resource** dialog box appears. If the Metadata Access Service does not exist, contact the Informatica administrator to create a new Metadata Access Service.
9. Expand a Google Cloud Storage bucket to list the folders inside the bucket and select the folder to list the files. Select the check box next to the Google Cloud Storage file that you want to add.

   Note: While creating a Google Cloud Storage data object, you must not select multiple Google Cloud Storage objects. After you create the Google Cloud Storage data object, click the **New (Insert)** button to add multiple objects.

10. Click **OK**.
11. If you selected **Binary**, **Avro**, **Json**, or **Parquet** as the **Resource Format**, click **Finish**.

    The data object appears under Physical Data Objects in the project or folder in the **Object Explorer** view.
12. If you selected **Flat** as the **Resource Format**, click **Next**.

    The **Column Projection** dialog box appears.

    a. Choose **Sample Metadata File**.

       You can click **Browse** and navigate to the directory that contains the file.

       Note: The **Delimited** and **Fixed-width** format properties are not applicable for PowerExchange for Google Cloud Storage.

    b. Click **Next**.

    c. Configure the following format properties:

Property	Description
Delimiters	Character used to separate columns of data. If you enter a delimiter that is the same as the escape character or the text qualifier, you might receive unexpected results. Google Cloud Storage reader and writer support Delimiters.
Text Qualifier	Quote character that defines the boundaries of text strings. If you select a quote character, the Developer tool ignores delimiters within pairs of quotes. Google Cloud Storage reader supports Text Qualifier.
Qualifier Mode	Qualifier behavior for the source object. You can select one of the following options: - **Minimal**. Default mode. Applies qualifier to data that have a delimiter value or a special character present in the data. Otherwise, the Data Integration Service does not apply the qualifier. - **All**. Applies qualifier to all data.
Row Delimiter	Specify a line break character. Select from the list or enter a character. Preface an octal code with a backslash (\). To use a single character, enter the character. The Data Integration Service uses only the first character when the entry is not preceded by a backslash. The character must be a single-byte character, and no other character in the code page can contain that byte. Default is line-feed, \012 LF (\n).
Header Line Number	Line number that you want to use as the header. You can also read a data from a file that does not have a header. To read data from a file with no header, specify the value of the Header Line Number field as 0.
First Data Row	Line number from where you want the Secure Agent to read data.
Escape Character	Character immediately preceding a column delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string. When you specify an escape character, the Data Integration Service reads the delimiter character as a regular character.

Property	Description
Maximum rows to preview	Specify the maximum number of rows that you want to display in the data preview.
Target Header	Indicates whether you want to write data with or without a header. **Note:** This property appears only when you configure a Google Cloud Storage data object write operation.

Note: The **Start import at line**, **Treat consecutive delimters as one**, and **Retain escape character in data** properties in the **Column Projection** dialog box are not applicable for PowerExchange for Google Cloud Storage.

d. Click **Next** to preview the flat file data object.

e. Click **Finish**.

The data object appears under Data Objects in the project or folder in the **Object Explorer** view.

Rules and Guidelines for Google Storage Data Objects

Use the following rules and guidelines when you configure a PowerExchange for Google Storage data objects:

- When you run a mapping on the Spark engine to read data from or write data to a Google Cloud Storage Avro file, the Google Cloud Storage data object must not contain a column of Null data type. Otherwise, the mapping fails.
- When you search for a Google Cloud Storage data object in the **Add Resource** dialog box, you must not use wild card characters in the search string.
- You can read and write data to Google Cloud Storage objects in JSON format in mappings that run in the native environment and on the Spark engine.

Creating a Google Cloud Storage Data Object Operation

You can create the data object read or write operation for Google Cloud Storage data objects. You can the add the Google Cloud Storage 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 Cloud Storage 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 Cloud Storage Target

You can create a Google Cloud Storage target using the **Create Target** option.

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 Cloud Storage 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 Cloud Storage** 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 Cloud Storage Data Object** dialog box appears.
8. Enter a name for the data object.
9. In the **Resource Format** list, select any of the following formats to create the target type:
   - Avro
   - Flat
   - JSON
   - Parquet
10. Click **Finish**.

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

Rules and Guidelines for Creating a new Google Cloud Storage Target

Use the following rules and guidelines when you create a new Google Cloud Storage target:

- You must specify a connection for the newly created Google Cloud Storage target in the **Connection** field to run a mapping.
- When you write data to a Google Cloud Storage Avro file using the **Create Target** option, you cannot provide a Null data type.
- For a newly created Google Cloud Storage target, provide a folder path and file name in the Google Cloud Storage data object write operation advanced properties.

- When you use a flat resource format to create a target, the Data Integration Service considers the following values for the formatting options:

Formatting Options	Values
Delimiters	Comma (,)
Text Qualifier	No quotes
Import Column Names From First Line	Generates header
Row Delimiter	Backslash with a character n (\n)
Escape Character	Empty

If you want to configure the formatting options, you must manually edit the projected columns.

CHAPTER 5

PowerExchange for Google Cloud Storage Mappings

This chapter includes the following topics:

PowerExchange for Google Cloud Storage Mappings Overview

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

You can define properties in an operation to determine how the Data Integration Service must extract data from a Google Cloud Storage source or load data to a Google Cloud Storage target. You can extract data from one or more Google Cloud Storage sources, and load data to one or more Google Cloud Storage 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 on 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 the 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.

When you run a mapping on the Spark engine to write data to an Avro, JSON, or Parquet file in Google Cloud Storage, PowerExchange for Google Cloud Storage creates a sub-folder within the original folder and uses the following logic to write data to Google Cloud Storage:

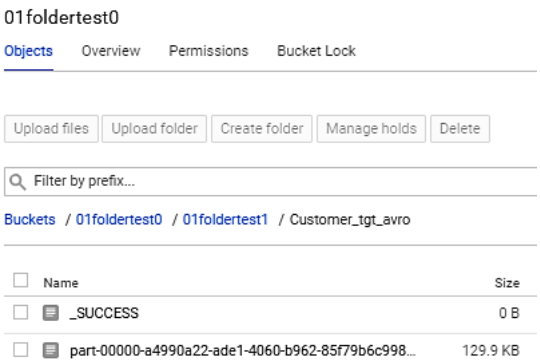
- The folder name is the native name of the target file.
- PowerExchange for Google Cloud Storage replaces period character (.) in the file name with underscore (_).
- PowerExchange for Google Cloud Storage creates a file with the following format:
  `part-0000-<unique identifier>.<file format>`
- PowerExchange for Google Cloud Storage writes data to the file in the sub-folder.

For example, select *Customer_tgt.avro* file in a Google Cloud Storage data object located in the following Google Cloud Storage path:

`/01foldertest0/01foldertest1`.

Use the write operation for the Google Cloud Storage data object in a mapping. When you run a mapping to write data to *Customer_tgt.avro* file, PowerExchange for Google Cloud Storage creates the **Customer_tgt_avro** sub-folder and **part-00000-a4990a22-ade1-4060-b962-85f79b6c9984-c000.avro** file.

The following image shows the Customer_tgt_avro sub-folder and part-00000-a4990a22-ade1-4060-b962-85f79b6c9984-c000.avro file in Google Cloud Storage:



When you run a mapping on the Spark engine to write data to a flat file in Google Cloud Storage, PowerExchange for Google Cloud Storage appends the file name with unique identifier information before it writes data to Google Cloud Storage in the following format:

`<original file name>-<unique identifier>`

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

APPENDIX A

Google Cloud Storage Data Type Reference

This appendix includes the following topics:

Data Type Reference Overview

Developer Tool uses the following data types in Google Cloud Storage mappings:

- Google Cloud Storage native data types. Google Cloud Storage data types appear in Google Cloud Storage 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.

Flat File and Transformation Data Types

Flat file data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table compares flat file data types to transformation data types:

Flat File Data type	Transformation Data type	Range
Bigint	Bigint	Precision of 19 digits, scale of 0
Number	Decimal	For transformations that support precision up to 38 digits, the precision is 1 to 38 digits, and the scale is 0 to 38. For transformations that support precision up to 28 digits, the precision is 1 to 28 digits, and the scale is 0 to 28. If you specify the precision greater than the maximum number of digits, the Data Integration Service converts decimal values to double in high precision mode. If the precision is greater than 15, the Data Integration Service converts decimal values to double in low-precision mode.
String	String	1 to 104,857,600 characters
Nstring	String	1 to 104,857,600 characters

Avro Data Types and Transformation Data Types

Avro data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table compares the Avro data types that the Data Integration Service supports and the corresponding transformation data types:

Avro Data Type	Transformation Data Type	Range
Array	Array	Unlimited number of characters.
Boolean	Integer	TRUE (1) or FALSE (0).
Bytes	Binary	Precision 4000.
Date	Date/Time	January 1, 0001 to December 31, 9999.

Avro Data Type	Transformation Data Type	Range
Decimal	Decimal	Decimal value with declared precision and scale. Scale must be less than or equal to precision. For transformations that support precision up to 38 digits, the precision is 1 to 38 digits, and the scale is 0 to 38. For transformations that support precision up to 28 digits, the precision is 1 to 28 digits, and the scale is 0 to 28. If you specify the precision greater than the maximum number of digits, the Data Integration Service converts decimal values to double in high precision mode.
Double	Double	Precision 15.
Fixed	Binary	1 to 104,857,600 bytes.
Float	Double	Precision 15.
Int	Integer	-2,147,483,648 to 2,147,483,647 Precision 10 and scale 0.
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Precision 19 and scale 0.
Map	Map	Unlimited number of characters.
Record	Struct	Unlimited number of characters.
String	String	1 to 104,857,600 characters.
Time	Date/Time	Time of the day. Precision to microsecond.
Timestamp	Date/Time	January 1, 0001 00:00:00 to December 31, 9999 23:59:59.997. Precision to microsecond.
Union	Corresponding data type in a union of ["primitive_type|complex_type", "null"] or ["null", "primitive_type|complex_type"].	Dependent on primitive or complex data type.

Avro Union Data Type

A union indicates that a field might have more than one data type. For example, a union might indicate that a field can be a string or a null. A union is represented as a JSON array containing the data types.
The Developer tool only interprets a union of ["primitive_type|complex_type", "null"] or ["null", "primitive_type|complex_type"]. The Avro data type converts to the corresponding transformation data type.

Avro Timestamp Data Type Support

The following table lists the Timestamp data type support for Avro file formats:

Timestamp Data type	Native	Spark
Timestamp_micros	Yes	Yes
Timestamp_millis	Yes	No
Time_millis	Yes	No
Time_micros	Yes	No

Unsupported Avro Data Types

The Developer tool does not support the following Avro data types:

- Enum
- Null
- Timestamp_tz

JSON Data Types and Transformation Data Types

JSON data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table compares the JSON data types that the Data Integration Service supports and the corresponding transformation data types:

JSON	Transformation	Range
Array	Array	Unlimited number of characters.
Double	Double	Precision of 15 digits.
Integer	Integer	-2,147,483,648 to 2,147,483,647. Precision of 10, scale of 0.
Object	Struct	Unlimited number of characters.
String	String	1 to 104,857,600 characters.

Unsupported JSON Data Types

The Developer tool does not support the following JSON data types:

- Date
- Decimal
- Timestamp
- Enum

- Union

Parquet Data Types and Transformation Data Types

Parquet data types map to transformation data types that the Data Integration Service uses to move data across platforms.

The following table compares the Parquet data types that the Data Integration Service supports and the corresponding transformation data types:

Parquet	Transformation	Range
Binary	Binary	1 to 104,857,600 bytes
Binary (UTF8)	String	1 to 104,857,600 characters
Boolean	Integer	-2,147,483,648 to 2,147,483,647 Precision of 10, scale of 0
Date	Date/Time	January 1, 0001 to December 31, 9999.
Decimal	Decimal	Decimal value with declared precision and scale. Scale must be less than or equal to precision. For transformations that support precision up to 38 digits, the precision is 1 to 38 digits, and the scale is 0 to 38. For transformations that support precision up to 28 digits, the precision is 1 to 28 digits, and the scale is 0 to 28. If you specify the precision greater than the maximum number of digits, the Data Integration Service converts decimal values to double in high precision mode.
Double	Double	Precision of 15 digits.
Float	Double	Precision of 15 digits.
Int32	Integer	-2,147,483,648 to 2,147,483,647 Precision of 10, scale of 0
Int64	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision of 19, scale of 0
Map	Map	Unlimited number of characters.
Struct	Struct	Unlimited number of characters.
Time	Date/Time	Time of the day. Precision to microsecond.

Parquet	Transformation	Range
Timestamp	Date/Time	January 1, 0001 00:00:00 to December 31, 9999 23:59:59.997. Precision to microsecond.
group (LIST)	Array	Unlimited number of characters.

The Parquet schema that you specify to read or write a Parquet file must be in smaller case. Parquet does not support case-sensitive schema.

Parquet Timestamp Data Type Support

The following table lists the Timestamp data type support for Parquet file formats:

Timestamp Data type	Native	Spark
Timestamp_micros	Yes	Yes
Timestamp_millis	Yes	No
Time_millis	Yes	No
Time_micros	Yes	No
int96	Yes	Yes

Unsupported Parquet Data Types

The Developer tool does not support the following Parquet data types:

- Timestamp_nanos
- Time_nanos
- Timestamp_tz

Rules and Guidelines for Data Types

Consider the following rules and guidelines for data types:

- Avro data types support:
  - Date, Decimal, and Timestamp data types are applicable when you run a mapping in the native environment or on the Spark engine in Cloudera CDH 6.3 distribution.
  - Time data type is applicable when you run a mapping in the native environment in Cloudera CDH 6.3 distribution.

- Parquet data types support:
  - When you set the -DINFA_HADOOP_DIST_DIR=hadoop\<Distro> option in the developerCore.ini file and import a Parquet file, the format of the imported metadata differs based on the distribution. For Cloudera CDP 7.1, the metadata is imported as string and for other supported distributions, the metadata is imported as UTF8.
  - Date, Time, and Timestamp data types till microseconds are applicable when you run a mapping in the native environment , Blaze, and Spark engine in the Hortonworks HDP 3.1, Azure HDInsight HDI 4.0, and Cloudera CDP 7.1 distributions.
  - Date, Time_Millis, and Timestamp_Millis data types are applicable when you run a mapping in the native environment or Spark engine in the Amazon Elastic MapReduce (EMR) 5.23, and MapR 6.1.
  - Decimal data types are applicable when you run a mapping in the native environment and Spark engine in Cloudera CDH 6.3, Hortonworks HDP 3.1, Amazon EMR 5.20, Amazon EMR 5.23, MapR 6.1, Google Dataproc 1.4, and Azure HDInsight HDI 4.0 distributions.
  - When you run a mapping and use Date data type that does not have a time value, the Data Integration Service adds the time value, based on the time zone, to the date in the target.
    For example, Date data type used in the source:

    ```
    1980-01-09
    ```

    Value generated in the target:

    ```
    1980-01-09 00:00:00
    ```
- When you run a mapping in the native environment and use Time data type in the source, the Data Integration Service writes incorrect date value to the target.
  For example, Time data type used in the source:

  ```
  1980-01-09 06:56:01.365235000
  ```

  Incorrect Date value is generated in the target:

  ```
  1899-12-31 06:56:01.365235000
  ```
- When you run a mapping in the native environment and use Date data type in the source, the Data Integration Service writes incorrect time value to the target.
  For example, Date data type used in the source:

  ```
  1980-01-09 00:00:00
  ```

  Incorrect Time value generated in the target:

  ```
  1980-01-09 05:30:00
  ```
- To run a mapping that reads and writes Date, Time, Timestamp, and Decimal data types, update the -DINFA_HADOOP_DIST_DIR option to the developerCore.ini file. The developerCore.ini file is located in the following directory:
  <Client installation directory>\clients\DeveloperClient\

  Add the following path to the developerCore.ini file:

  ```
  -DINFA_HADOOP_DIST_DIR=hadoop\<Hadoop distribution>_<version>
  ```

  For example: -DINFA_HADOOP_DIST_DIR=hadoop\CDH_6.3
- To use precision up to 38 digits for Decimal data type in the native environment, set the EnableSDKDecimal38 custom property to true for the Data Integration Service.

INDEX