

# Google Bigtable Connector Template and General Guidelines

---

## Table of Contents

<b>GOOGLE BIGTABLE CONNECTOR TEMPLATE AND GENERAL GUIDELINES .....</b>	<b>1</b>
TABLE OF CONTENTS.....	1
<b>INFORMATICA CLOUD GOOGLE BIGTABLE CONNECTOR USER GUIDE.....</b>	<b>2</b>
<b>CHAPTER 1 - INTRODUCTION TO GOOGLE BIGTABLE CONNECTOR.....</b>	<b>2</b>
GOOGLE BIGTABLE CONNECTOR OVERVIEW (REQUIRED) .....	2
GOOGLE BIGTABLE CONNECTOR TASK AND OBJECT TYPES (REQUIRED) .....	2
GOOGLE BIGTABLE OBJECTS (BASED ON CONTENT) .....	2
INTRODUCTION TO GOOGLE BIGTABLE .....	2
ADMINISTRATION OF GOOGLE BIGTABLE CONNECTOR (REQUIRED) .....	3
<b>CHAPTER 2 - GOOGLE BIGTABLE CONNECTIONS .....</b>	<b>6</b>
GOOGLE BIGTABLE CONNECTION OVERVIEW .....	6
GOOGLE BIGTABLE CONNECTION PROPERTIES .....	6
EXAMPLE GOOGLE BIGTABLE MAPPING TASK.....	7
<b>CHAPTER 3 - MAPPING AND MAPPING TASKS WITH GOOGLE BIGTABLE .....</b>	<b>7</b>
READ OPERATION MAPPING.....	7
WRITE OPERATION MAPPING .....	9
<b>APPENDIX: DATA TYPE REFERENCE.....</b>	<b>11</b>
DATA TYPE REFERENCE OVERVIEW .....	11
GOOGLE BIGTABLE AND TRANSFORMATION DATA TYPES .....	12
<b>GUIDELINES FOR INFORMATICA INTELLIGENT CLOUD SERVICES/ INFORMATICA CLOUD DATA INTEGRATION DOCUMENTATION .....</b>	<b>12</b>
<b>PRODUCT NAMING CONVENTIONS.....</b>	<b>12</b>
WHEN TO USE IICS AND CLOUD DATA INTEGRATION .....	12
<b>TASKS NAMING CONVENTIONS .....</b>	<b>12</b>
<b>CONTENT, PROCEDURES, EXAMPLES, AND SCREENSHOTS .....</b>	<b>13</b>
<b>STYLE COMMENTS .....</b>	<b>13</b>
<b>WORDING.....</b>	<b>13</b>
<b>RELEASE NOTES GUIDELINES .....</b>	<b>13</b>
<b>MESSAGE WRITING EXAMPLES .....</b>	<b>17</b>

## INFORMATICA CLOUD GOOGLE BIGTABLE CONNECTOR USER GUIDE

### Chapter 1 - Introduction to Google Bigtable Connector

#### Google Bigtable Connector Overview (Required)

You can use Google Bigtable Connector to read, insert and delete data from Google Bigtable. It can also be used to read the columns. It can be used to insert data to a Bigtable and it can also be used to delete row from a table by using rowkey.

#### Google Bigtable Connector Task and Object Types (Required)

This Connector is flat record based and it supports Mapping Task.

The following table lists the Google Bigtable object types that you can include in Informatica Cloud tasks:

Task Type	Source	Target
Mapping	Yes	Yes

#### Google Bigtable Objects (Based on content)

You can work with the following types of Google Bigtable objects in Data Integration:

##### Standard objects

The standard Objects supported by Google Bigtable are Read, insertData and delete.

Object Name	Read	Insert	Update	Delete
Read	Yes	NA	NA	NA
InsertData	NA	Yes	NA	NA
Delete	NA	NA	NA	Yes
Update	NA	NA	Yes	NA

#### Introduction to Google Bigtable

Google Cloud Bigtable is a sparsely populated table that can scale to billions of rows and thousands of columns, enabling you to store terabytes or even petabytes of data. A Bigtable is a sparse, distributed, persistent multidimensional sorted map.

Cloud Bigtable stores data in massively scalable tables, each of which is a sorted key/value map. The table is composed of *rows*, each of which typically describes a single entity, and *columns*, which contain individual values for each row. Each row is indexed by a single *row key*, and columns that are related to one another are typically grouped together into a *column family*. Each column is identified by a combination of the column family and a *column qualifier*, which is a unique name within the column family.

Each row/column intersection can contain multiple *cells*, or versions, at different timestamps, providing a record of how the stored data has been altered over time. Cloud Bigtable tables are sparse; if a cell does not contain any data, it does not take up any space.

### Administration of Google Bigtable Connector (required)

1. Create a Google cloud platform account to access Google Bigtable.
2. Create a Google could platform project.

**Google Cloud Platform**

New Project

**Warning:** You have 23 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)  
[MANAGE QUOTAS](#)

Project name \*  
 Infometry GoogleBigtable

Project ID \*  
 bigtable-256108

Project ID can have lowercase letters, digits, or hyphens. It must start with a lowercase letter and end with a letter or number.

Location \*  
 No organization [BROWSE](#)

Parent organization or folder

**CREATE** **CANCEL**

3. Enable the Cloud Bigtable and Cloud Bigtable Admin APIs.

**Google Cloud Platform** infometry GoogleBigtable

**APIs & Services** [+ ENABLE APIS AND SERVICES](#)

Dashboard  
 Library  
 Credentials  
 OAuth consent screen  
 Domain verification  
 Page usage agreements

Hide unused APIs

Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
BigQuery API				
BigQuery Storage API				
Cloud Bigtable Admin API				
Cloud Bigtable API				
Cloud Datastore API				

4. Open the Create Instance page in the Google Cloud Platform Console.
  - a. For instance name, enter infometry-bigtable.
  - b. For instance ID, enter infometry-bigtable.
  - c. For instance type, select Development.

- d. For Storage type, select SSD.
- e. For Cluster ID, enter infometry-bigtable-c1.
- f. For Region, select us-east1.
- g. For Zone, select us-east1-c.
- h. Click Create to create the instance.

☰ Google Cloud Platform
Infometry GoogleBigtable ▾

---

Bigtable

←
Create an instance

---

**Instance name**  
For display purposes only

**Instance ID**  
ID is permanent

**Instance type** ⓘ

Production (recommended)  
Minimum of 3 nodes. High availability. Cannot downgrade later.

Development  
Low-cost instance for development and testing. Does not provide high availability or replication. Can upgrade to Production later.

**Storage type** ⓘ  
Choice is permanent. Applies to all clusters. Affects cost.

SSD  
Lower latency and more rows read per second. Typically used for real-time serving use cases, such as ad serving and mobile app recommendations.

HDD  
Higher latency for random reads. Good performance on scans and typically used for batch analytics, such as machine learning or data mining.

**Clusters**

Cluster
↑

**Cluster ID**  
ID is permanent.

**Location**  
Choice is permanent. Determines where cluster data is stored. To reduce latency and increase throughput, store your data near the services that need it.

**Region**  **Zone**

**Nodes**  
Development instance performance is limited to the equivalent of a single node. Nodes increase data throughput and allow more rows to be read per second. Upgrade to Production for more nodes.

Done
Cancel

+ Add replicated cluster

Adding a second cluster automatically replicates all data between clusters. To add a second cluster, upgrade this instance to a production instance.

[Replication guidance](#)

Create
Cancel

**Monthly resource costs**  
Monthly costs reflect Bigtable resources only. Network traffic (replication and internet egress) costs are dependent on the location of your clusters and application request behavior. [Learn more](#)

Try another storage size (per cluster)

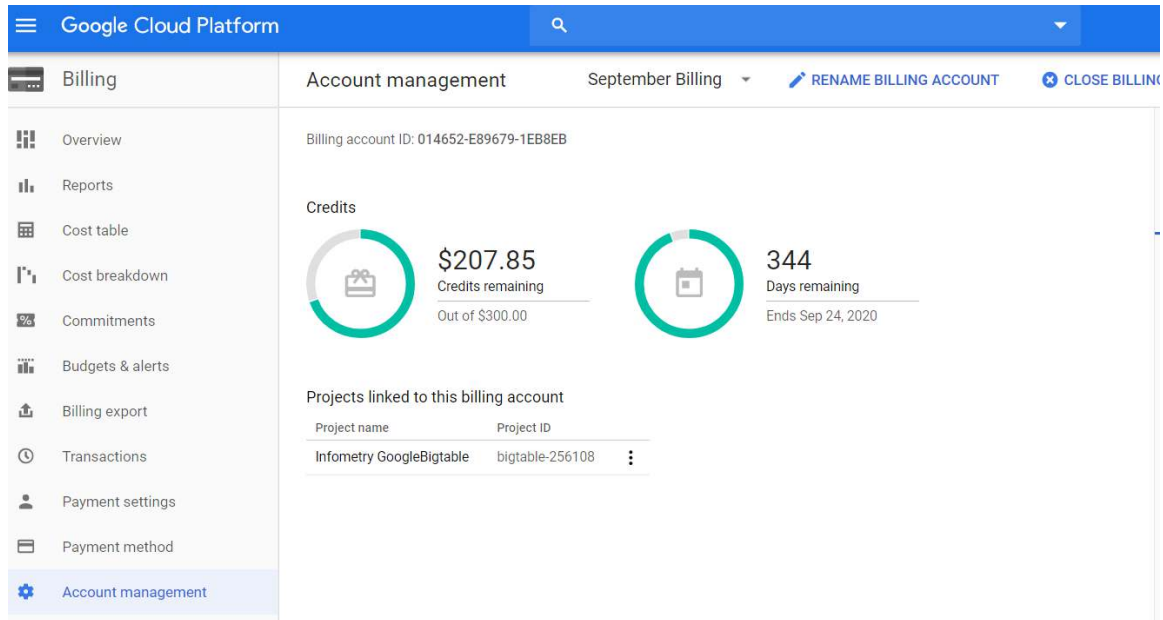
1	GB
---	----

Item	Estimated cost
▶ 1 cluster	\$474.50/month
▶ 1 GB SSD	\$0.17/month
<b>Total</b>	<b>\$474.67/month</b>

Node charges are for provisioned resources, regardless of node usage. The same node charges apply even if your instance is inactive. [Learn more](#)

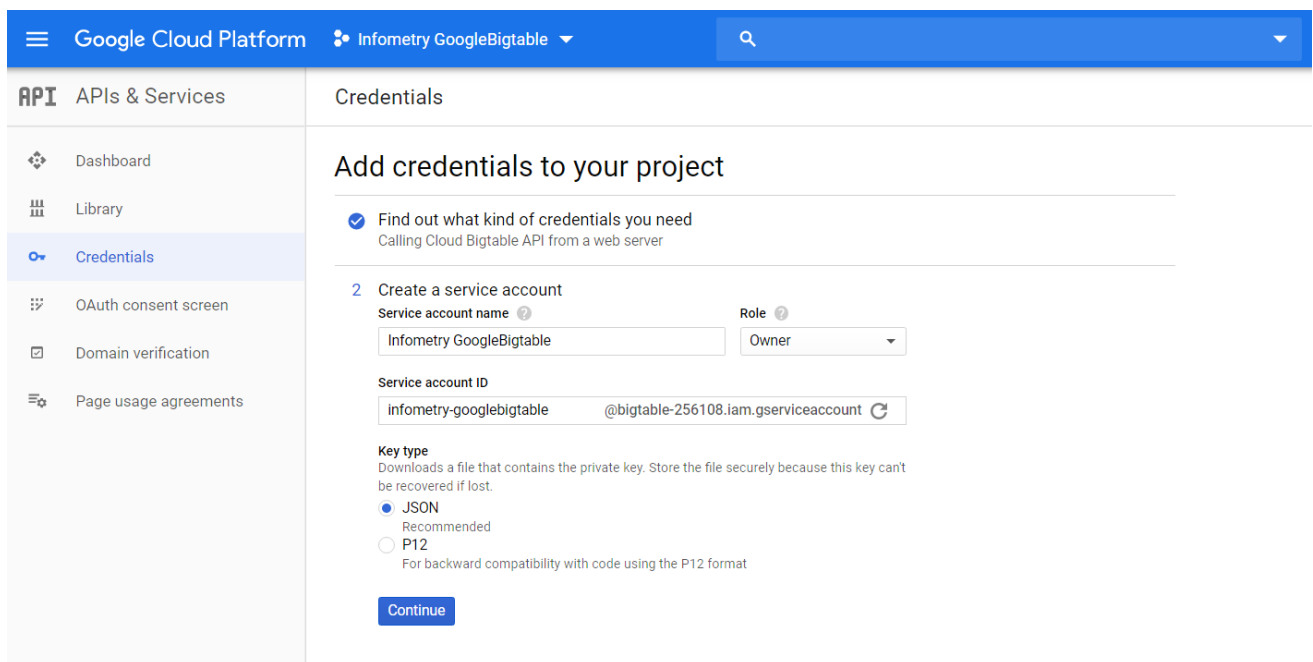
**Summary**  
Monthly charge: \$474.67 per month (1 GB data, development instance)  
Effective hourly rate: \$0.65

5. Make sure that billing is enabled for your Google Cloud Platform project.



6. 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 Bigtable connection in Informatica Cloud.

The following image shows the Credentials page where you can create the service account and key:



## Chapter 2 - Google Bigtable Connections

Create a Google Bigtable connection to read data from Google Bigtable source and write data to a Google Bigtable target. You must create a connection for Google Bigtable Object that you want to connect to. You can use Google Bigtable connections in Synchronization tasks and mapping tasks.

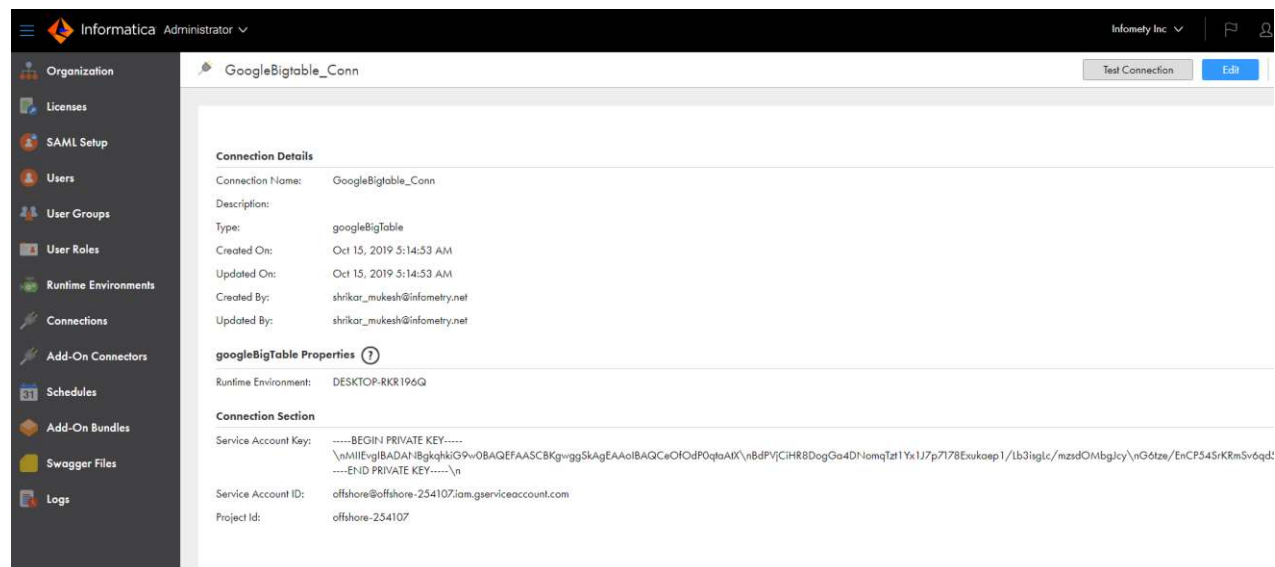
### Google Bigtable Connection Overview

Google Bigtable connector has 3 connection attributes and all of them are mandatory. Create a Google Bigtable connection to access Google Bigtable data from Data Integration. You can create a connection on the Connections page or when you create a task. After you create a connection, it becomes available to all users who have access to the organization.

### Google Bigtable Connection Properties

Connection Attributes	Description
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.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account.

The following image gives the connection information:



## Example Google Bigtable Mapping Task

### Chapter 3 - Mapping and Mapping Tasks with Google Bigtable

#### Read Operation Mapping

1. In Mapping for Google Bigtable Read operation, specify Task name as **GoogleBigtable\_Read\_Mapping**, take new Source, new Target and give Properties to Source and target.

Select Source connection which you have created for Google Bigtable.

Source properties: -

connection: GBT\_Windows\_Repro(googleBigtable)

source type: Single Object

Object: student

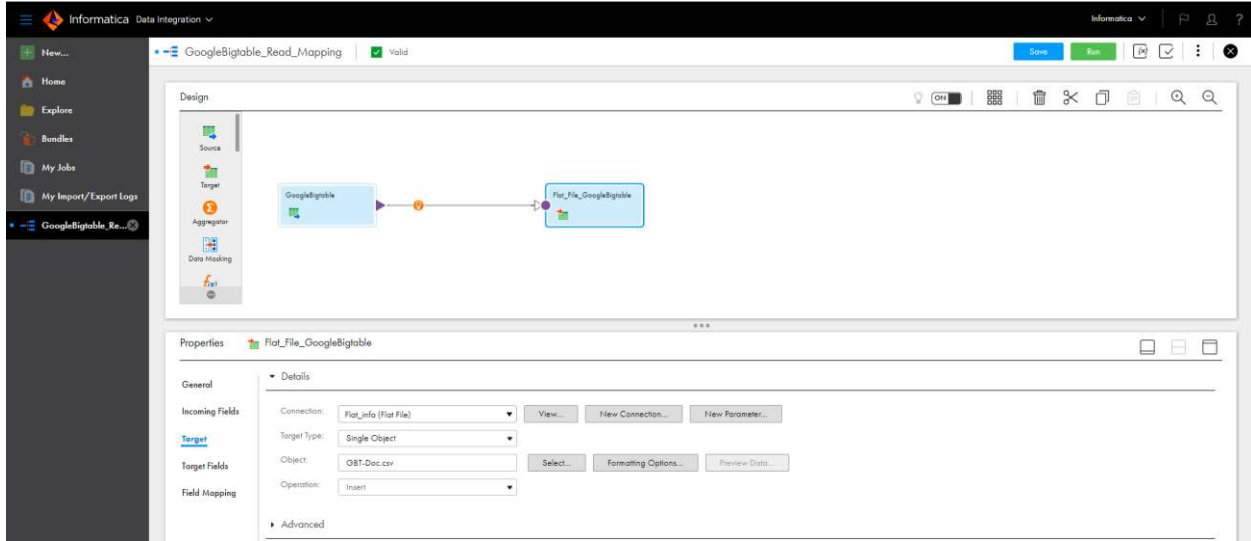
The screenshot displays the Informatica Data Integration software interface. The top navigation bar includes the Informatica logo and 'Data Integration' menu. The left sidebar contains navigation options: 'New...', 'Home', 'Explore', 'Bundles', 'My Jobs', 'My Import/Export Logs', and 'GoogleBigtable\_Re...'. The main workspace is titled 'GoogleBigtable\_Read\_Mapping' and shows a 'Design' view with a flow from a 'GoogleBigtable' source to a 'Flat\_File\_GoogleBigtable' target. Below the design view, the 'Properties' pane for the 'GoogleBigtable' source is visible, showing the following configuration:

- General**
- Source** (selected)
- Fields**
- Partitions**
- Details**
  - Connection: GBT\_Windows\_Repro (googleBigtable) [View... New Connection... New Parameter...]
  - Source Type: Single Object
  - Object: student [Select... Preview Data...]
- Query Options**

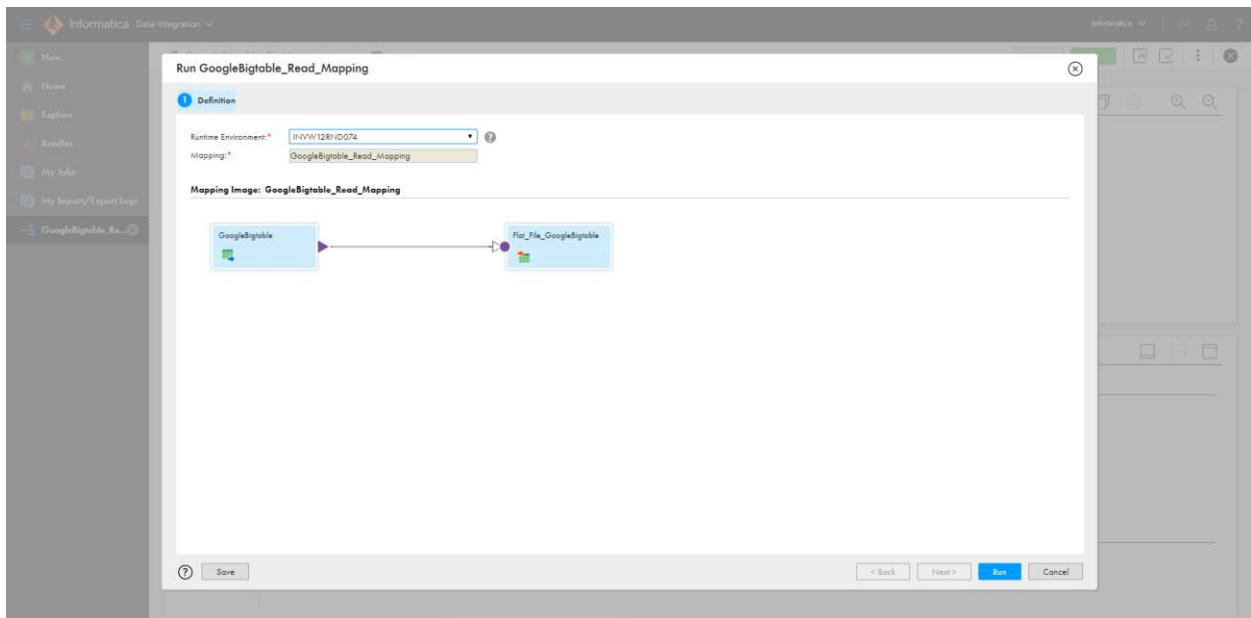
- In Mapping for Google Bigtable Read operation, Select target connection which you have created for Google Bigtable.

Target properties: -

- Connection: Flat infa(Flat File)
- Target type: Single Object
- Object: GBT-Doc.csv
- Operation: Insert

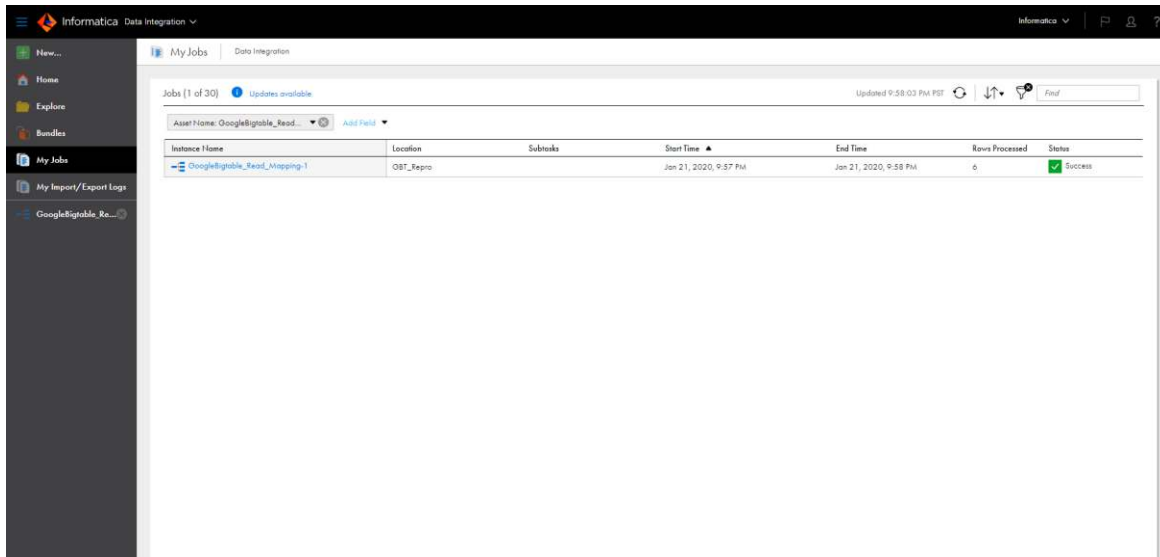


- In Mapping configuration task for Google Bigtable Read operation, specify Definition task name as **GoogleBigtable\_Read\_Mapping**, Select Runtime environment and run the task.



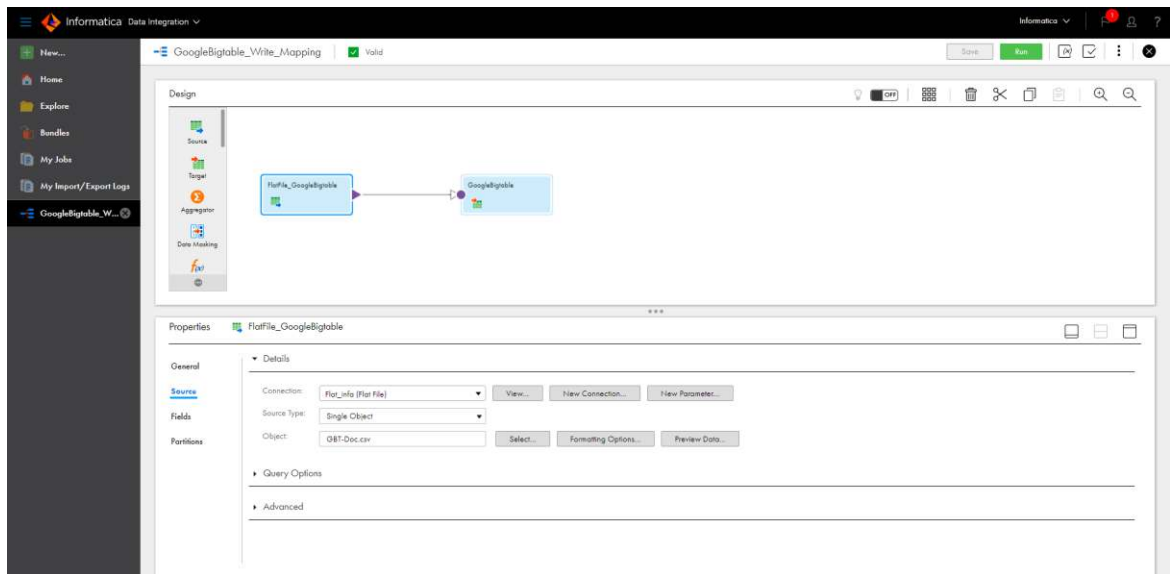


4. Click on My Jobs to see the result.



## Write Operation Mapping

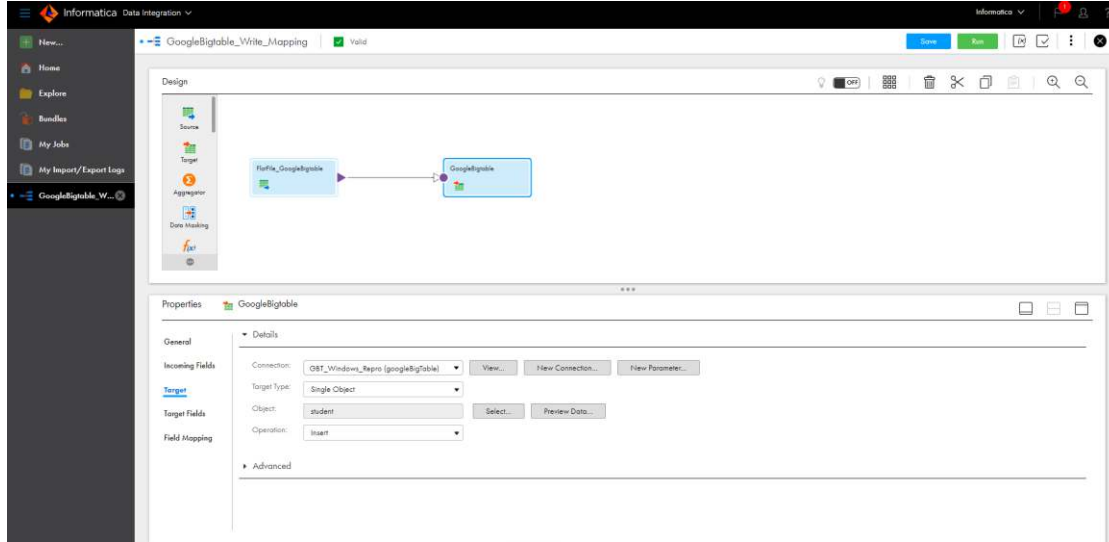
- In Mapping for Google Bigtable Write operation, specify Task name as **GoogleBigtable\_Write\_Mapping**, take new Source, new Target and give Properties to Source and target.  
 Select Source connection which you have created for Google Bigtable.  
 Source properties: -
  - connection: Flat info(Flat File)
  - source type: Single Object
  - Object: GBT-Doc.csv



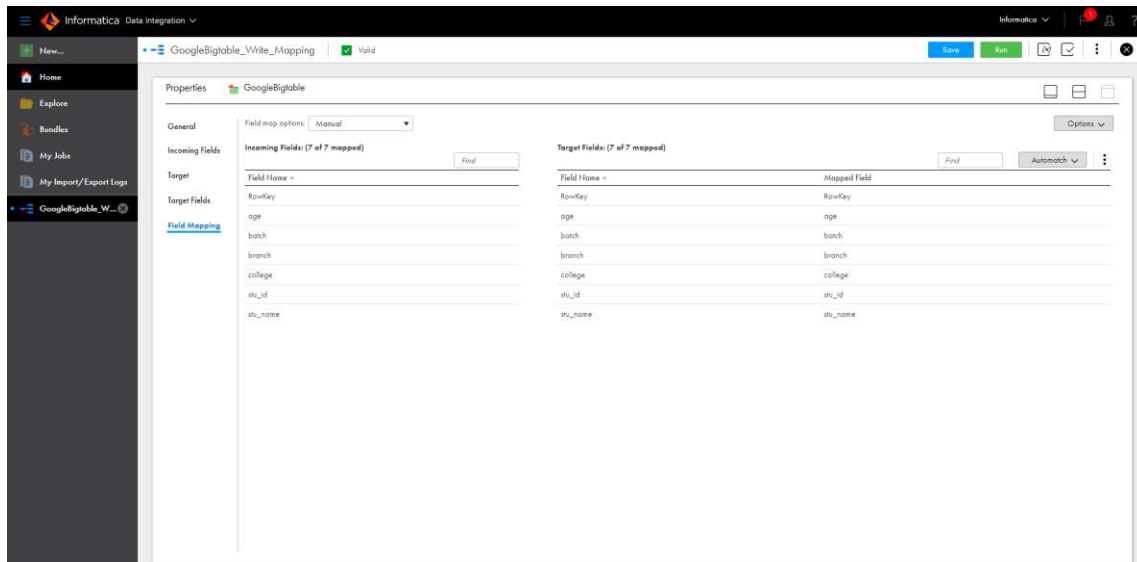
- In Mapping for Google Bigtable Write operation, Select target connection which you have created for Google Bigtable.

Target properties: -

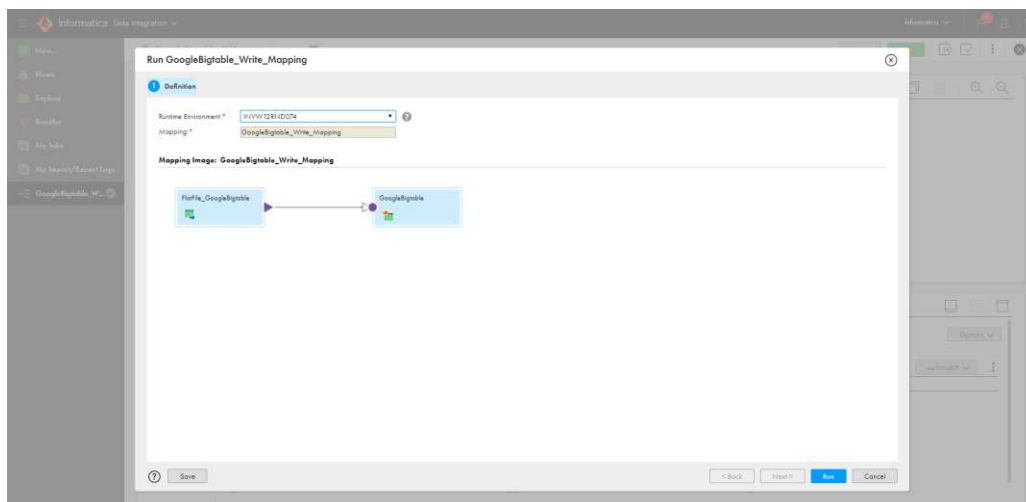
- Connection: GBT\_Windows\_Repo(googleBigtable)
- Target type: Single Object
- Object: student
- Operation: Insert



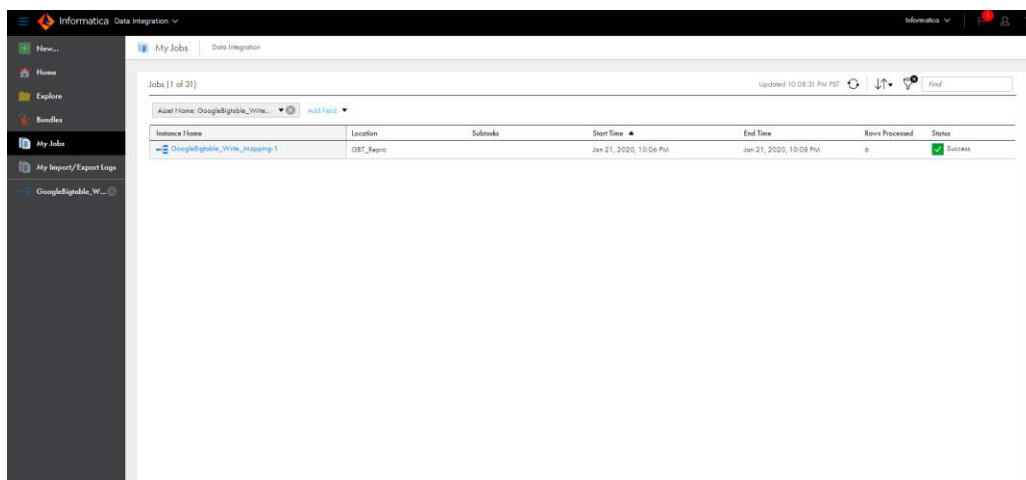
- Field Mapping should be done before running the task.



- In Mapping configuration task for Google Bigtable Read operation, specify Definition task name as **GoogleBigtable\_Write\_Mapping**, Select Runtime environment and run the task.



- Click on My Jobs to see the result.



## Appendix: Data Type Reference

### Data Type Reference Overview

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

#### Google Bigtable data types:

Google Bigtable 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 Bigtable and Transformation Data Types

The following table lists the supporting Google Bigtable data types and the corresponding transformation data types:

Google Bigtable <b>Data Type</b>	<b>Transformation Data Type</b>	<b>Description</b>
String	String	1 to 10485760 characters

## *GUIDELINES FOR INFORMATICA INTELLIGENT CLOUD SERVICES/ INFORMATICA CLOUD DATA INTEGRATION DOCUMENTATION*

We need to maintain and create content for legacy Informatica Cloud and Informatica Intelligent Cloud Services (IICS) aka Florence concurrently.

### *Product Naming Conventions*

The official name for Florence is **Informatica Intelligent Cloud Services** (IICS). IICS is a parent product and includes multiple Informatica products (**Informatica Cloud**, ICRT, DQ, MDM, and so on)  
The official name for the new version of Informatica Cloud is **Informatica Cloud Data Integration** when used on IICS.

Product name in legacy connector user guides: **Informatica Cloud**  
Product name in Florence connector user guides: **Informatica Cloud Data Integration**  
Parent product name in Florence connector user guides: **Informatica Intelligent Cloud Services**

### *When to use IICS and Cloud Data Integration*

Use **Informatica Intelligent Cloud Services** as the product name whenever you are talking about the platform. For example, a user logs in to IICS (not Cloud Data Integration), and you create IICS users (not Cloud Data Integration users).

Use Cloud Data Integration when referring to the Integration Designer only. For example, you create mappings, mapping tasks, and task flows in Cloud Data Integration.

### *Tasks Naming Conventions*

The following table lists tasks names for Informatica Cloud and Cloud Data Integration:

<b>Informatica Cloud</b>	<b>Cloud Data Integration</b>
Data Synchronization task	synchronization task
Data Replication task	replication task
Mapping Configuration task	mapping task
Data Masking task	data masking task

**Note:** Write task names for Cloud Data Integration in lower case unless they are used in headings, as UI labels, or in the beginning of the sentences.

## *Content, Procedures, Examples, and Screenshots*

Make sure that all the content including examples and screenshots have been modified as per the Cloud Data Integration UI in the Florence connector guides.

## *STYLE COMMENTS*

### *Wording*

Connector does not require an article. For example, write “Use Marketo REST Connector...” instead of “Use the Marketo REST Connector...”

Note wording for Secure Agent and runtime environments. Always use the content references that we have set up. This terminology has been through tech review, but the development is still evolving, and the terminology might change.

## *RELEASE NOTES GUIDELINES*

### **Rules and Guidelines for New Features**

When you document a new feature in a release guide, ensure that the content follows the standard structure.

### **Rules and guidelines for all release guides**

Consider the following rules and guidelines:

- Use the following lead-in sentence for each version topic: "This section describes new features in version *<version>*. "
- Alphabetize the sub-topics and sections in each functional category topic, unless you receive different guidance from product management. PM might occasionally want to highlight certain information first.
- In general, do not document new UI features. Document changed behavior, not changed appearance. For example, if there is a new dialog box for unlocking repository objects in Developer tool, add an entry about unlocking objects in the Developer tool. The exception is if the UI for the entire tool has changed.
- Optionally, add images to show UI or other changes when you think these will benefit the user.
- Include a book reference for each feature.

Use the following syntax for the book reference: "For more information, see the *<full book name, including version number>*." Use the `<cite>` tag for the product name, version, number, and book name. For example, "See the *Administrator Guide*."

- Optionally, include a chapter reference for a feature.

### **Known Limitations**

A known limitation is a limitation of the designed and implemented product. Unsupported functionality is not a limitation. The known limitations section of the release notes provides brief descriptions and workarounds to high priority and high severity product limitations.

**Important:** Limitations can be difficult to write. Although you might write just a few sentences, you must be fully familiar with the product and the circumstances that surround the limitation before you can convey it accurately and succinctly in the Release Note writeup. Do not rely on terminology in the bug tracking system, as much of the terminology is internal. Verify that you use terminology that is consistent with user doc and the product.

## Known Limitations Criteria

Before you document a known limitation, verify that the limitation meets criteria for the release notes. Informatica includes high priority and high severity product limitations that Development is committed to fixing.

When you review known limitations that are release note candidates, consider the following criteria:

### Limitation type

Known limitations must be functional limitations. Do not include enhancement requests or documentation limitations. If the nature of the limitation is functional and the type is "Documentation," look for the linked functional limitation.

### Content to include

After you verify the limitation type, priority, and status, review the content of the limitation. Include the following types of errors in the known limitations:

- Product failures, such as hanging, termination, or error response to user request
- System failures, such as core dumps and crashing
- Inconsistent data or unexpected results
- Regressions
- Any other issue that might result in a call to Informatica Global Customer Support

### Content to exclude

Do not include the following types of bugs in known limitations:

- Any limitation that might affect install or upgrade. For example, the installation fails with memory errors, or the upgrade fails to upgrade privileges properly. Document all install and upgrade limitations in the Installation topic of the release notes.
- PAM-related issues of supported systems. For example, do not include a limitation requesting the support of a particular version of an OS for a product.

- Corner case bugs, or bugs that are unlikely to be found by a customer. If a limitation is extremely difficult to reproduce or was possibly discovered by QA under severe stress testing, consider exclusion from the release notes.
- Requests for additional functionality. A bug that indicates a request for support of additional functionality or a feature is not considered a limitation for release notes. Key words to watch for are "support" and "should have."
- Bugs that are minor irritations. This can include misaligned user interface text, misspellings, and too many clicks or scrolls.
- Bugs that are not reproducible.

## Exceptions

If you question whether to include a limitation or not, consider whether excluding the limitation from the release notes would result in a call to GCS. Note the following circumstances when we can relax the release note criteria:

- New products with a customer base that is vocal about documenting all limitations
- End-user products where limitations such as abnormal scrolling or too many clicks are higher priority
- Special requests from product management, development, or QA

## Known Limitations Write-up

Write up a known limitation to describe the limitation instead of the expected behavior. Include any workaround.

A known limitation has the following elements:

### Bug number

The bug number is the functional bug number in Jira. If Development linked a bug for the doc impact, do not use the documentation bug number.

### Bug description

When you write a known limitation, write a short description of the bug behavior instead of the expected behavior.

- **Change:** The RestoreDomain command should not generate an exception if you set the -tc option.
- **To:** The RestoreDomain command generates an exception if you set the -tc option.

### Workaround

If a limitation has a workaround, include it under the limitation. If Jira does not include a workaround, ask Dev/QA to provide one.

Do not document the following types of bugs as limitations that have the following workarounds:

#### **Custom property**

Document custom properties, called undocumented flags, as an internal KB article.

#### **System patch**

Document operating system patches in the installation section of the release notes. QA provides this information to Documentation. If the workaround is a patch that is not in the list of patches, verify with QA whether it belongs in the list. If you need to document the behavior, it might fit as a separate topic in the installation section of the release notes.

## **Fixed Limitations**

Document fixed limitations that were reported by customers or were reported as limitations in a previous release.

When you write a fixed limitation, write a short description of the bug behavior instead of the fixed behavior.

- **Change:** The Column Profiling Details dialog box appears when you view the column profile for a source column in a mapping specification.
- **To:** The Column Profiling Details dialog box does not appear when you view the column profile for a source column in a mapping specification.

If a fixed limitation was previously documented as a known limitation, move the entry to the Fixed topic. Delete any workaround that was documented with limitation.

## **Fixed Limitations Criteria**

Before you write a fixed limitation, verify that it meets the criteria for the release notes.

Most of the time, bugs are fixed in the code, tested, and closed. However, Dev might close a bug for multiple reasons. For example, it might be a duplicate of another bug, or it might be closed with a workaround. When you review a closed bug, you need to read the closing comments at the end of the notes to find out why a bug was actually closed. If it is not checked in to the code, we cannot publish it as being closed.

When you review fixed limitations that are release note candidates, consider the following criteria:

#### **Fixed release**

Verify that the limitation was verified and closed in the release that you are documenting.



**Note:** If the limitation was linked from a previous release and this fix was merged from a previous release, do not document it again as fixed. The content reference informing customers about fixes in previous releases is sufficient.

**Status**

Verify that the status of the limitation is "closed." If it is "resolved," verify with Dev and QA that they will be able to close it for the release. Watch the bug for the status change.

**Content to include**

Include fixed limitations that meet the following conditions:

- The fix was checked in to the code.
- The bug was not opened in the current release.
- The bug was previously documented as a known limitation, or it was reported by a customer or GCS.

**Content to exclude**

Do not include fixed limitations that meet the following conditions:

- If a limitation is closed as a duplicate, look at the linked bug to see if it belongs with known limitations.
- If a limitation is closed as "will not fix" or "as-designed," consider documenting the issue in the Knowledge Base or user documentation. If the behavior still seems buggy, put this in a Knowledge Base article. If the workaround, or the user actions required to get the desired behavior are fairly simple, consider including it in the user docs.
- If a limitation is closed with a workaround, you can include it in a Knowledge Base article. Workarounds can include undocumented flags, registry edits, and system patches.
- If an issue was issue that is closed with a custom property, it is a candidate for the Knowledge Base. We do not expose these properties to all customers, so any documentation will be through an internal KB article. Ask Global Customer Support and QA if they want this documented.
- If a limitation is closed with a note that Documentation is adding it as a limitation in the release notes, let Development know that we cannot document anything in the release notes without Dev commitment to fix.

**MESSAGE WRITING EXAMPLES**

The examples in this section show messages that have been rewritten to follow the message writing guidelines.

The following table shows original and edited messages and the guidelines used for the rewrite:

Original Message	Edited Message	Guidelines for Rewrite
Attachment file {filename} not found.	Cannot find the post-session email attachment file {filename}.	-

Error executing stored procedure...	An error occurred while executing the stored procedure for transformation {transformationname}.	<p>Provide as much information as possible.</p> <p>-</p> <p>End each message with a period.</p> <p>-</p> <p>Provide as much information as possible.</p> <p>-</p> <p>Qualify all parameters.</p>
NULL external procedure name.	The External Procedure transformation {transformationname} failed because the name of the external procedure is null. Specify the name of the external procedure.	<p>-</p> <p>Provide as much information as possible.</p> <p>-</p> <p>Qualify all parameters.</p> <p>-</p> <p>Tell where the error occurred.</p>
Getting free block from exchange failed!!	Internal error. The Integration Service encountered a fatal error while getting a block from the buffer pool. Contact Informatica Global Customer Support.	<p>-</p> <p>End each sentence with a period.</p> <p>-</p> <p>Provide as much information as possible.</p> <p>-</p> <p>Use words that the customer will understand.</p>
ERROR: Field name used in join not found in transform definition.	Internal error. The Integration Service cannot find the port {portname} used in the join condition for the Joiner transformation {transformationname}. Contact Informatica Global Customer Support.	<p>-</p> <p>Provide as much information as possible.</p> <p>-</p> <p>Tell where the error occurred.</p> <p>-</p> <p>Qualify all parameters.</p> <p>-</p> <p>Spell out words.</p>
Pushdown optimization is not supported because you enabled row error logging. To use pushdown optimization, disable row error logging.	Pushdown optimization is skipped because session {sessionname} has row error logging enabled.	<p>-</p> <p>Avoid offending the user.</p> <p>-</p> <p>Provide as much information as possible.</p>
{transformationname} cannot be pushed to the target database because it is connected to both {targetname1} and {targetname2}. No transformation can be pushed to more than one target.	The transformation {transformationname} cannot be pushed to the target database because it is connected to multiple targets: {targetname1}, {targetname2}	<p>-</p> <p>Provide as much information as possible.</p> <p>-</p> <p>Be concise and direct.</p> <p>-</p> <p>Do not start a message with a parameter.</p>