

Google PubSubV2 Connector Template and General Guidelines

Table of Contents

GOOGLE PUBSUBV2 CONNECTOR TEMPLATE AND GENERAL GUIDELINES.....	1
TABLE OF CONTENTS.....	1
INFORMATICA CLOUD GOOGLE PUBSUBV2 CONNECTOR USER GUIDE.....	2
CHAPTER 1 - INTRODUCTION TO GOOGLE PUBSUBV2 CONNECTOR.....	2
GOOGLE PUBSUBV2 CONNECTOR OVERVIEW (REQUIRED)	2
GOOGLE PUBSUBV2 CONNECTOR TASK AND OBJECT TYPES (REQUIRED).....	2
GOOGLE PUBSUBV2 OPERATIONS (BASED ON CONTENT)	2
INTRODUCTION TO GOOGLE PUBSUBV2	3
ADMINISTRATION OF GOOGLE PUBSUBV2 CONNECTOR (REQUIRED).....	4
CHAPTER 2 - GOOGLE PUBSUBV2 CONNECTIONS.....	6
GOOGLE PUBSUBV2 CONNECTION OVERVIEW	6
GOOGLE PUBSUBV2 CONNECTION PROPERTIES.....	6
EXAMPLE GOOGLE PUBSUBV2 MAPPING TASK	7
CHAPTER 3 - MAPPING AND MAPPING TASKS WITH GOOGLE PUBSUBV2	7
GOOGLE PUBSUBV2 SOURCES IN MAPPING TASKS.....	7
PUBSUBV2 TARGET IN MAPPINGS	ERROR! BOOKMARK NOT DEFINED.
READ OPERATION MAPPING.....	8
WRITE OPERATION MAPPING	10
APPENDIX: DATA TYPE REFERENCE.....	13
DATA TYPE REFERENCE OVERVIEW	13
GOOGLE PUBSUB V2 AND TRANSFORMATION DATA TYPES	14
GUIDELINES FOR INFORMATICA INTELLIGENT CLOUD SERVICES/ INFORMATICA CLOUD DATA INTEGRATION DOCUMENTATION	14
PRODUCT NAMING CONVENTIONS.....	14
WHEN TO USE IICS AND CLOUD DATA INTEGRATION.....	14
TASKS NAMING CONVENTIONS	14
CONTENT, PROCEDURES, EXAMPLES, AND SCREENSHOTS.....	14
STYLE COMMENTS.....	15
WORDING.....	15
RELEASE NOTES GUIDELINES	15
MESSAGE WRITING EXAMPLES	19

INFORMATICA CLOUD GOOGLE PUBSUB V2 CONNECTOR USER GUIDE

Chapter 1 - Introduction to Google PubSub V2 Connector

Google PubSub V2 Connector Overview (Required)

You can use Google PubSub V2 Connector to pull the message from Google Pub/Sub and publish the message to Google PubSub. It can also be used to read, create, update and delete topics, subscriptions And snapshots.

Google PubSub V2 Connector Task and Object Types (Required)

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

The following table lists the Google PubSub V2 object types that you can include in Informatica Cloud tasks:

Task Type	Source	Target
Mapping	Yes	Yes

Google PubSub V2 Objects (Based on content)

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

Standard objects

The standard Objects supported by Google PubSub are mentioned in the below table.

Object Name	Read	Insert	Delete	Update
topics_create	NA	Yes	NA	NA
topics_get	Yes	NA	NA	NA
topics_list	Yes	NA	NA	NA
topics_getIamPolicy	Yes	NA	NA	NA
topics_delete	NA	NA	Yes	NA
topics_TestIamPermissions	NA	Yes	NA	NA
topics_SetIamPolicy	NA	Yes	NA	NA
topics_Update	NA	NA	NA	Yes
subscriptions_Create	NA	Yes	NA	NA
subscriptions_Delete	NA	NA	Yes	NA
subscription_GetIamPolicy	Yes	NA	NA	NA

subscriptions_SetIamPolicy	NA	Yes	NA	NA
subscriptions_TestIamPermissions	NA	Yes	NA	NA
subscriptions_ModifyAckDeadline	NA	Yes	NA	NA
subscription_Update	NA	NA	NA	Yes
subscription_Get	Yes	NA	NA	NA
subscriptions_ModifyPushConfig	NA	Yes	NA	NA
snapshots_TestIamPermissions	NA	Yes	NA	NA
subscriptions_Seek	NA	Yes	NA	NA
subscriptions_List	Yes	NA	NA	NA
snapshots_Create	NA	Yes	NA	NA
snapshot_SetIamPolicy	NA	Yes	NA	NA
snapshot_GetIamPolicy	Yes	NA	NA	NA
snapshots_Delete	NA	NA	Yes	NA
snapshots_List	Yes	NA	NA	NA
snapshots_Update	NA	NA	NA	Yes
topic:<topicName>	NA	Yes	NA	Yes
subscription:< subscriptionName>	Yes	NA	NA	NA

Introduction to Google PubSub

Pub/Sub is an asynchronous messaging service that decouples services that produce events from services that process events.

You can use Pub/Sub as messaging-oriented middleware or event ingestion and delivery for streaming analytics pipelines

Pub/Sub offers durable message storage and real-time message delivery with high availability and consistent performance at scale. Pub/Sub servers run in all Google Cloud regions around the world.

Core concepts

Topic: A named resource to which messages are sent by publishers.

Subscription: A named resource representing the stream of messages from a single, specific topic, to be delivered to the subscribing application.

Message: The combination of data and (optional) attributes that a publisher sends to a topic and is eventually delivered to subscribers.

Message attribute: A key-value pair that a publisher can define for a message. For example, key `iana.org/language_tag` and value `en` could be added to messages to mark them as readable by an English-speaking subscriber.

Administration of Google PubSub V2 Connector (required)

1. Create a Google cloud platform account to access Google PubSub V2.
2. Create a Google could platform project.

Google Cloud Platform

New Project

You have 22 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
INF-PUBSUB

Project ID *
inf-pubsub

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 Pub/Sub Api.

The screenshot shows the Google Cloud Platform 'APIs & Services' page. The left sidebar contains navigation options: Dashboard, Library, Credentials, OAuth consent screen, Domain verification, and Page usage agreements. The main content area displays a table of API statistics with columns for Name, Requests, Errors (%), Latency, median (ms), and Latency, 95% (ms). The table lists several APIs including Compute Engine API, Cloud Pub/Sub API, Cloud Bigtable Admin API, Cloud Functions API, Cloud Bigtable API, BigQuery API, BigQuery Storage API, Cloud Datastore API, and Cloud Debugger API.

Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Compute Engine API	400,872	0.002	235.523	491.081
Cloud Pub/Sub API	1,071	7.75	1,445.888	9,097.301
Cloud Bigtable Admin API	569	0	1,801.183	5,849.715
Cloud Functions API	36	0	207.178	429.916
Cloud Bigtable API	18	0	12,058.624	16,305.357
BigQuery API				
BigQuery Storage API				
Cloud Datastore API				
Cloud Debugger API				

- 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 PubSubV2 connection in Informatica Cloud.

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

The screenshot shows the 'Create service account' page in the Google Cloud Platform IAM & Admin console. The page has a three-step process: 1. Service account details, 2. Grant this service account access to project (optional), and 3. Grant users access. The 'Service account details' section includes a text input for 'Service account name' (Infometry Google Pub/Sub), a text input for 'Service account ID' (infometry-google-pub-sub), and a text input for 'Service account description'. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

Chapter 2 - Google PubSubV2 Connections

Create a Google PubSubV2 connection to read data from Google PubSub source and write data to a Google PubSub target. You must create a connection for Google PubSubV2 Object that you want to connect. You can use Google PubSub V2 connections in mapping tasks.

Google PubSub V2 Connection Overview

Google PubSubV2 connector has 3 connection attributes, 3 of them are mandatory to create a Google PubSubV2 connection to access Google PubSub data from Data Integration. You can create a Google PubSubV2 connection on the Connections page. After you create a connection, it becomes available to all users who have access to the organization.

Google PubSubV2 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:

The screenshot shows a configuration form for a Google PubSubV2 connection. It is organized into three sections:

- Connection Details:**
 - Connection Name: PubSub_Attr
 - Description: (empty field)
 - Type: GooglePubSub (dropdown menu)
- GooglePubSub Properties:**
 - Runtime Environment: ASH_PAT (dropdown menu)
- Connection Section:**
 - Service Account Id: pubsub-bigtable@inf-pubsub.iam.gserviceaccount.c
 - Service Account Key: (masked with dots)
 - Project Id: inf-pubsub

Chapter 3 - Mapping and Mapping Tasks with Google PubSubV2

Google PubSubV2 Sources in Mapping Tasks

TopicsGet, TopicsList, TopicsGetIamPolicy, SubscriptionsGet, SubscriptionsList, SubscriptionGetIamPolicy, subscription:< subscriptionName>, SnapshotsList and SnapshotsGetIamPolicy objects are supported under source.

Google PubSub V2 Target in Mapping Tasks

TopicsCreate, TopicsSetIamPolicy, TopicsTestIamPermissions, topic:<topicName>, SubscriptionsCreate, SubscriptionsSeek, SubscriptionsSetIamPolicy, SubscriptionsTestIamPermissions, SubscriptionsModifyAckDeadline, SubscriptionsModifyPushConfig, SnapshotsCreate, SnapshotsTestIamPolicy, SnapshotsSetIamPolicy, TopicsDelete, SubscriptionsDelete, SnapshotsDelete, TopicsUpdate, SubscriptionUpdate and SnapshotsUpdate objects are supported under target.

Data Filters

TopicsGet, TopicsList, TopicsGetIamPolicy, SnapshotsList, SnapshotsGetIamPolicy SubscriptionGet, SubscriptionGetIamPolicy and SubscriptionList.

Filter Fields:

The following tables provides information on various filter fields available for Litmos objects:

Object	Filters Fields	Data Type	Example
TopicGet	Topic	String	Topic = topicname
TopicsList	NextPageToken	String	nextPageToken = djdefkikaks
TopicsGetIamPolicy	resource	String	Resource = topicname
SnapshotList	NextPageToken	String	nextPageToken = djdefkikaks
SnapshotGetIamPolicy	Resource	String	Resource = snapshotName
SubscriptionGet	Subscription	String	Subscription = subscriptionName
SubscriptionGetIamPolicy	Resource	String	Resource = subscriptionName
SubscriptionList	NextPageToken	String	nextPageToken = djdefkikaks

Write Attributes:

Object	Attribute Name	Data Type
topic:<topicName>	maxMessageForBatch	Integer

Read Attributes:

Object	Attribute Name	Data Type
subscription:<subscriptionName>	ACK	Boolean
	maxMessages	Integer

Examples of Google PubSub V2 Mapping Tasks

Read Operation Mapping

1. In Mapping for Google PubSubV2 Read operation, specify Task name as **GooglePubSubV2_Read**, take new Source, new Target and give Properties to Source and target.

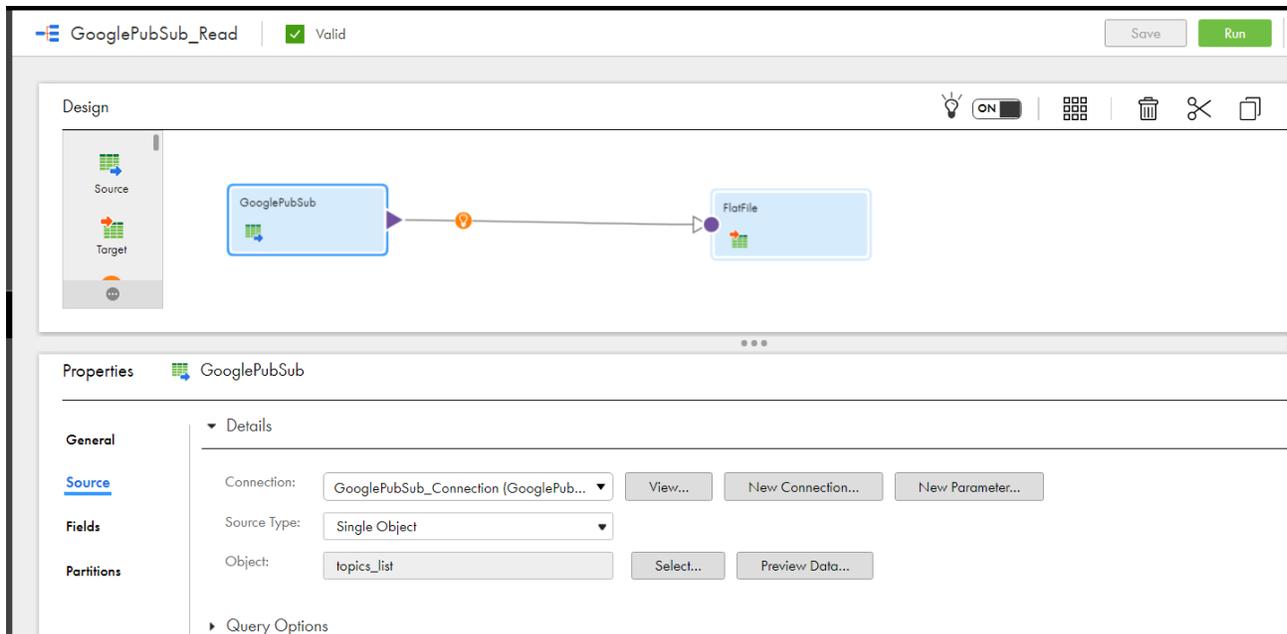
Select Source connection which you have created for Google PubSubV2.

Source properties: -

 Connection: GooglePubSub V2_Connection

 Source type: Single Object

 Object: topics_list



2. In Mapping for Google PubSubV2 Read operation, select target connection which you have created for Google PubSub V2.

Target properties: -

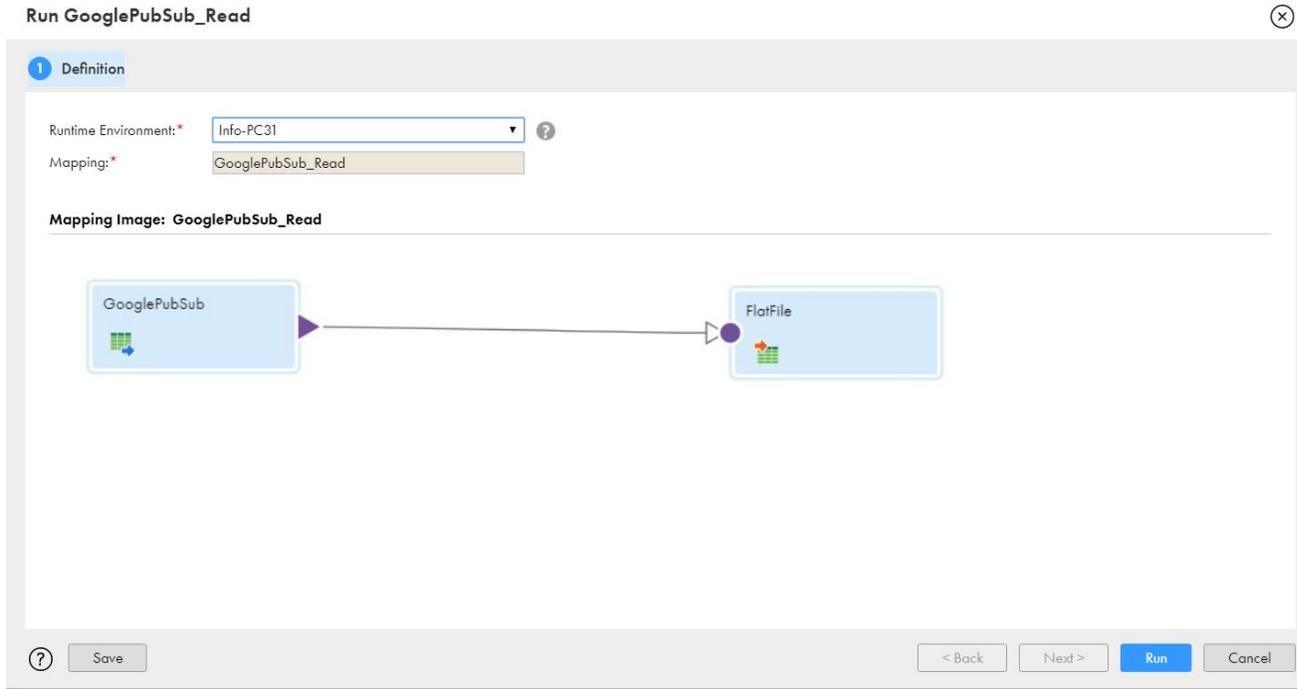
- a. Connection: FlatFile_Connection

- b. Target type: Single Object
- c. Object: TopicsList.csv
- d. Operation: Insert

The screenshot displays the Informatica Cloud Connector configuration interface. The top bar shows the task name "GooglePubSub_Read" and a "Valid" status. The "Design" pane shows a flow from a "GooglePubSub" source to a "FlatFile" target. The "Properties" pane for the "FlatFile" target is expanded to show the "Details" section, which includes the following configuration:

Property	Value	Buttons
Connection:	FlatFile_Connection (Flat File)	View... New Connection... New Parameter...
Target Type:	Single Object	
Object:	TopicsList.csv	Select... Formatting Options... Preview Data...
Operation:	Insert	

3. In Mapping configuration task for Google PubSub V2 Read operation, specify Definition task name as **GooglePubSub V2_Read**, Select Runtime environment and run the task.



4. Click on My Jobs to see the result.

My Jobs | Data Integration

Jobs (2 of 75) Updated 12:58:02 AM PDT

Asset Name: GooglePubSub_Read Add Field

Instance Name	Location	Subtasks	Start Time	End Time	Rows Processed	Status
GooglePubSub_Read-2	GdTest\PubSub		Apr 13, 2020, 12:56 AM	Apr 13, 2020, 12:5...	12	Success

Write Operation Mapping

1. In Mapping for Google PubSub V2 Write operation, specify Task name as **GooglePubSub V2_Write**, take new Source, new Target and give Properties to Source and target. Select Source connection which you have created for Google PubSub V2. Source properties: -
 - a. connection: FlatFile_Connection(Flat File)
 - b. source type: Single Object

c. Object: topicCreate.csv

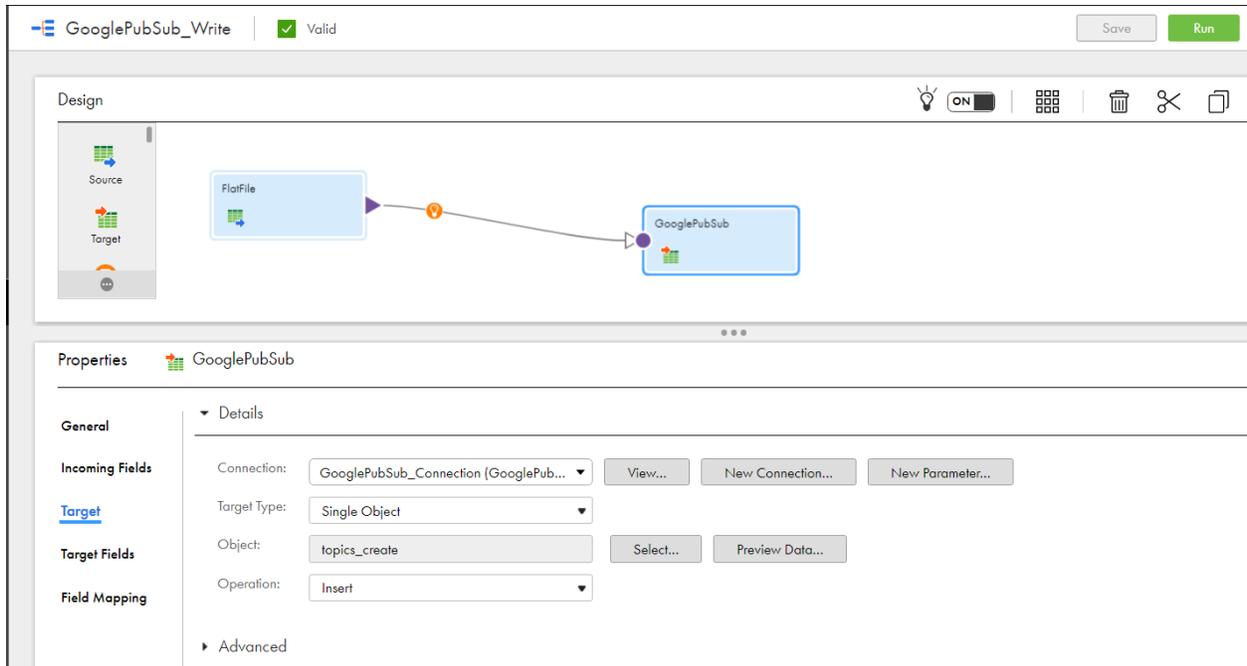
The screenshot displays the Informatica Cloud Connector configuration interface. At the top, the window title is "GooglePubSub_Write" with a "Valid" status indicator. The "Design" pane shows a data flow from a "FlatFile" source to a "GooglePubSub" target. The "Properties" pane is open for the "FlatFile" source, showing the following configuration:

- General**
- Source**
- Fields**
- Partitions**
- Details**
 - Connection: FlatFile_Connection (Flat File) [View... New Connection... New Parameter...]
 - Source Type: Single Object
 - Object: topicCreate.csv [Select... Formatting Options... Preview Data...]
 - Query Options
 - Advanced

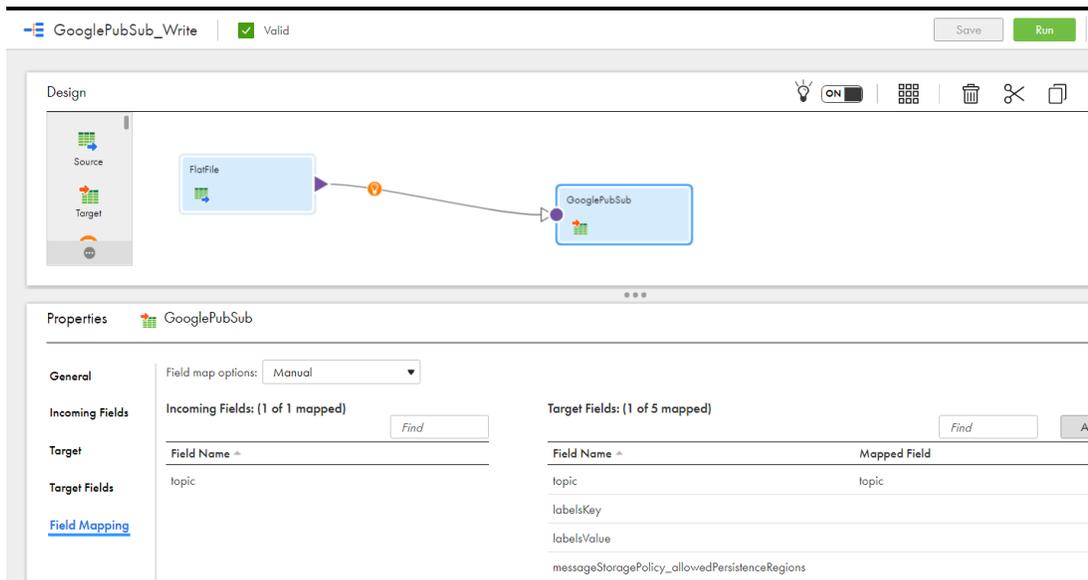
2. In Mapping for Google PubSub V2 Write operation, select target connection which you have created for Google PubSub V2.

Target properties: -

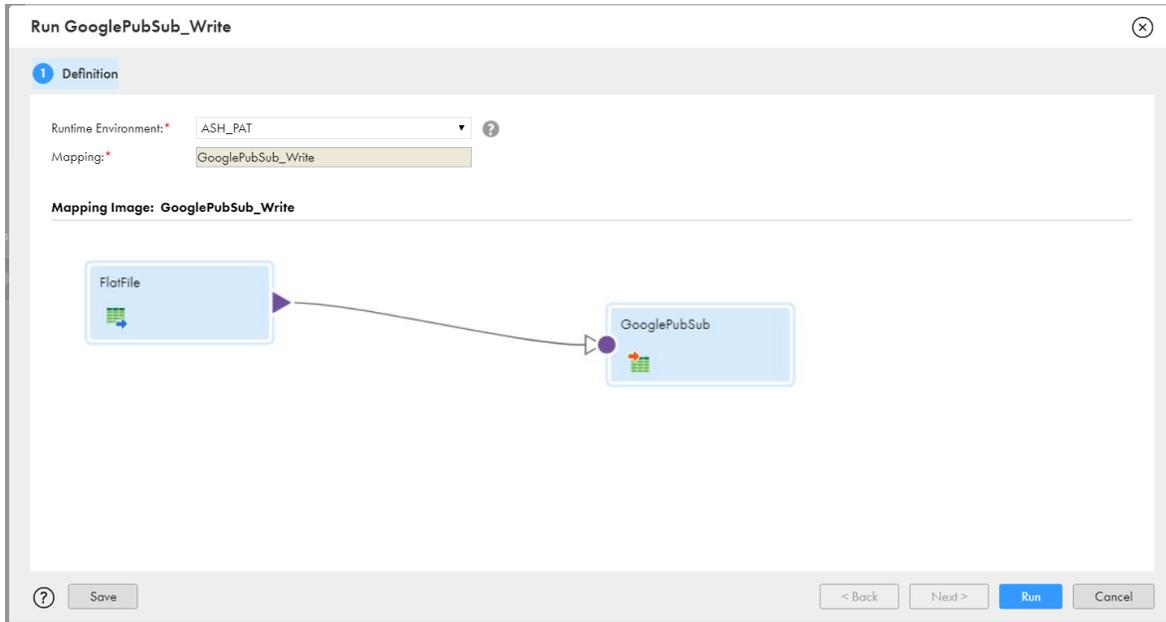
- Connection: GooglePubSub V2_Connection
- Target type: Single Object
- Object: topics_create
- Operation: Insert



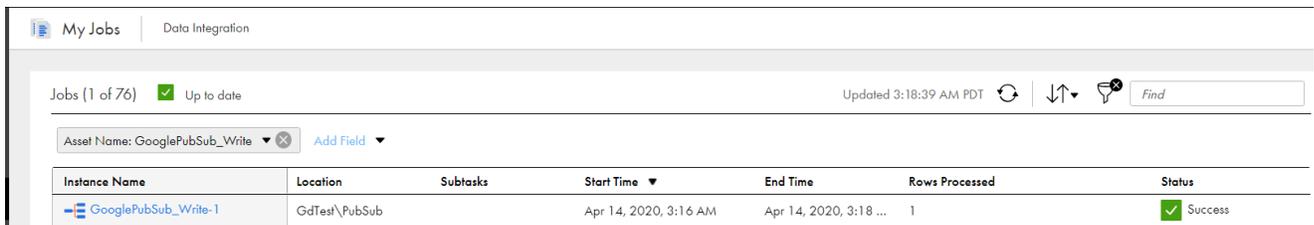
3. Field Mapping should be done before running the task.



4. In Mapping configuration task for Google PubSub V2 Read operation, specify Definition task name as **GooglePubSub V2_Write**, Select Runtime environment and run the task.



5. 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 PubSub V2:

Google PubSub V2 data types:

Google PubSub V2 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 PubSub V2 and Transformation Data Types

The following table lists the supporting Google PubSub V2 data types and the corresponding transformation data types:

Google PubSub V2 Data Type	Transformation Data Type	Description
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:

Informatica Cloud	Cloud Data Integration
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}.	- Provide as much information as possible.
Error executing stored procedure...	An error occurred while executing the stored procedure for transformation {transformationname}.	- End each message with a period. - Provide as much information as possible. - Qualify all parameters.
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.	Provide as much information as possible. - Qualify all parameters. - Tell where the error occurred.
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.	- End each sentence with a period. - Provide as much information as possible. - Use words that the customer will understand.
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.	- Provide as much information as possible. - Tell where the error occurred. - Qualify all parameters. - Spell out words.
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.	- Avoid offending the user. - Provide as much information as possible.
{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}	- Provide as much information as possible. - Be concise and direct. - Do not start a message with a parameter.