

HubSpot Connector Template and General Guidelines

Table of Contents

HUBSPOT CONNECTOR TEMPLATE AND GENERAL GUIDELINES.....	1
TABLE OF CONTENTS.....	1
INFORMATICA CLOUD HUBSPOT CONNECTOR USER GUIDE	2
CHAPTER 1 - INTRODUCTION TO HUBSPOT CONNECTOR.....	2
HUBSPOT CONNECTOR OVERVIEW (REQUIRED)	2
HUBSPOT CONNECTOR TASK AND OBJECT TYPES (REQUIRED).....	2
HUBSPOT OBJECTS (BASED ON CONTENT)	2
INTRODUCTION TO HUBSPOT	3
ADMINISTRATION OF HUBSPOT CONNECTOR (REQUIRED).....	3
CHAPTER 2 - HUBSPOT CONNECTIONS.....	4
HUBSPOT CONNECTION OVERVIEW	4
HUBSPOT CONNECTION PROPERTIES.....	4
CHAPTER 3 - MAPPING AND MAPPING TASKS WITH HUBSPOT.....	5
HUBSPOT SOURCES IN MAPPING TASKS.....	5
HUBSPOT TARGET IN MAPPING TASKS	5
EXAMPLES OF HUBSPOT MAPPING TASKS	5
READ OPERATION MAPPING.....	5
WRITE OPERATION MAPPING	7
APPENDIX: DATA TYPE REFERENCE.....	10
DATA TYPE REFERENCE OVERVIEW	10
HUBSPOT AND TRANSFORMATION DATA TYPES	10
GUIDELINES FOR INFORMATICA INTELLIGENT CLOUD SERVICES/ INFORMATICA CLOUD DATA INTEGRATION DOCUMENTATION	10
PRODUCT NAMING CONVENTIONS.....	10
WHEN TO USE IICS AND CLOUD DATA INTEGRATION.....	11
TASKS NAMING CONVENTIONS	11
CONTENT, PROCEDURES, EXAMPLES, AND SCREENSHOTS	11
STYLE COMMENTS	11
WORDING.....	11
RELEASE NOTES GUIDELINES	11
MESSAGE WRITING EXAMPLES	16

INFORMATICA CLOUD HUBSPOT CONNECTOR USER GUIDE

Chapter 1 - Introduction to HubSpot Connector

HubSpot Connector Overview (Required)

HubSpot Connector can be used to provide connectivity between HubSpot and from variety of external sources like Snowflake. This connector gives organizations the ability to streamline data and processes across information systems for operational efficiency, including the automation of training assignments and the moderation of administrative tasks.

Examples use case: Data Integration between Snowflake and HubSpot. Reading data from Snowflake and writing data into HubSpot. Reading data from HubSpot and Writing data into Snowflake.

HubSpot Connector Task and Object Types (Required)

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

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

Task Type	Source	Target
Mapping	Yes	Yes

HubSpot Objects (Based on content)

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

Standard objects

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

Object Name	Read	Insert	Update	Delete
Contacts	Yes	Yes	Yes	Yes
Deals	Yes	Yes	Yes	Yes
Companies	Yes	Yes	Yes	Yes
Products	Yes	Yes	Yes	Yes
Line_Items	Yes	Yes	Yes	Yes
Tickets	Yes	Yes	Yes	Yes
Quotes	Yes	No	No	No

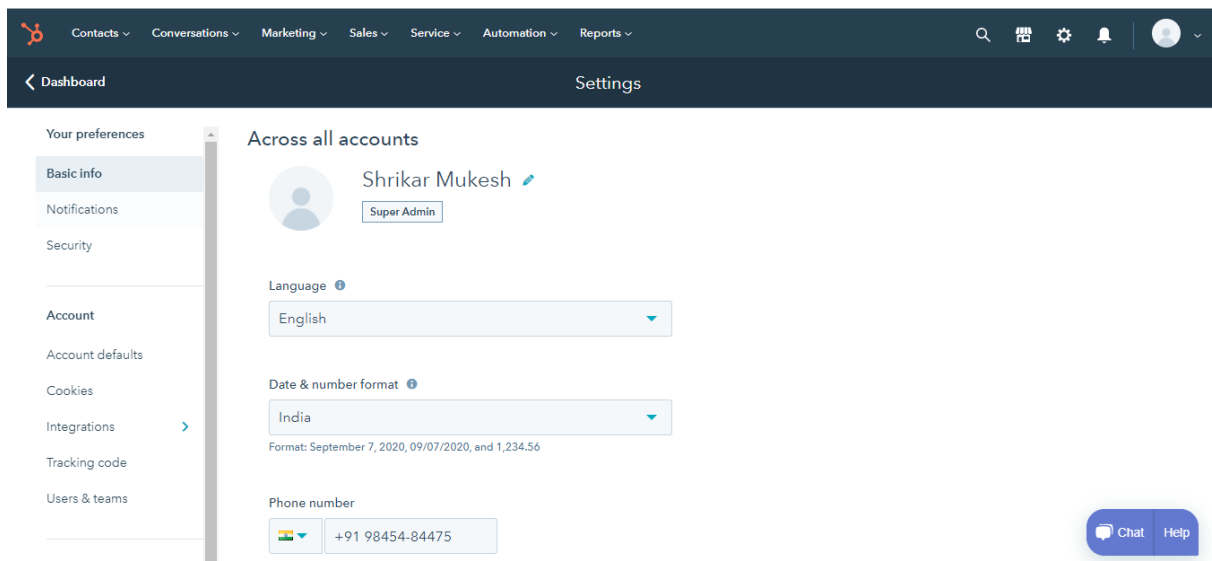
Introduction to HubSpot

HubSpot is an inbound marketing and sales software that helps companies attract visitors, convert leads, and close customers. HubSpot is a very popular CRM and marketing automation tool in the SMB space Online Course Builder

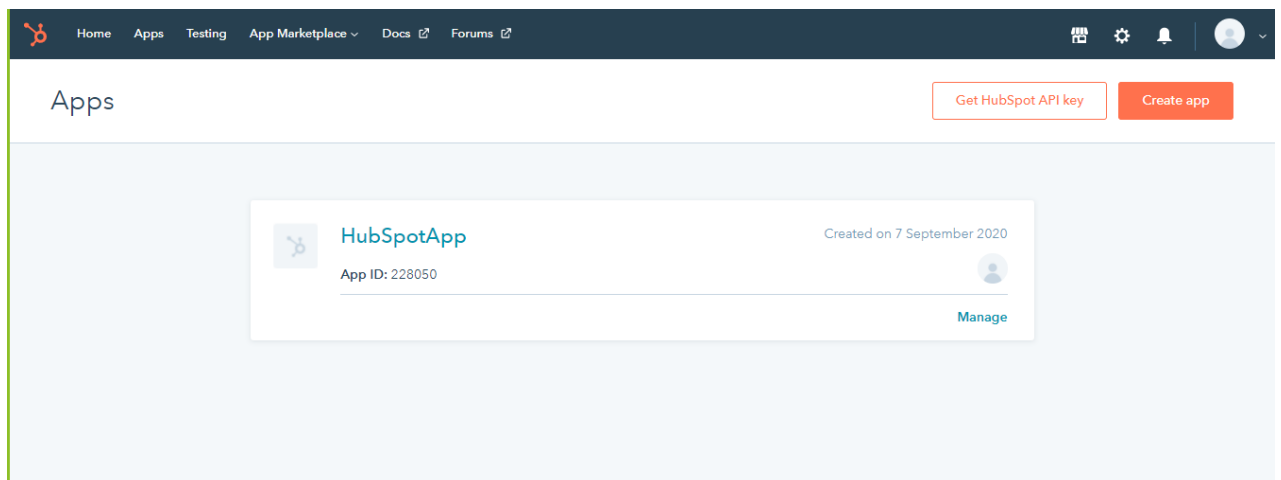
Administration of HubSpot Connector (required)

Before you use HubSpot Connector, complete the following prerequisite tasks:

1. Create a HubSpot account to access HubSpot.

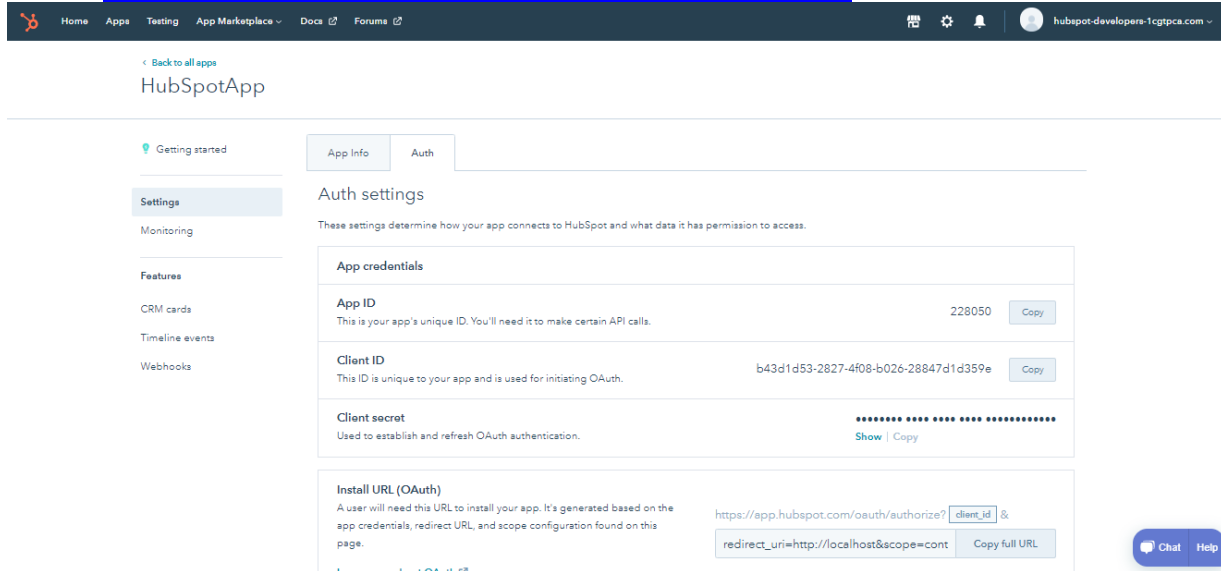


2. Create a developer HubSpot account for generating authorization credentials.
3. Before Generating Authorization, we should create app for the developer account.



- After creating app, we will get required connection credentials like Client Id and Client Secret, To generate third connection attribute refresh token, use the below HubSpot Oath QuickStart guide

<https://developers.hubspot.com/docs/api/oauth-quickstart-guide>



Chapter 2 - HubSpot Connections

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

HubSpot Connection Overview

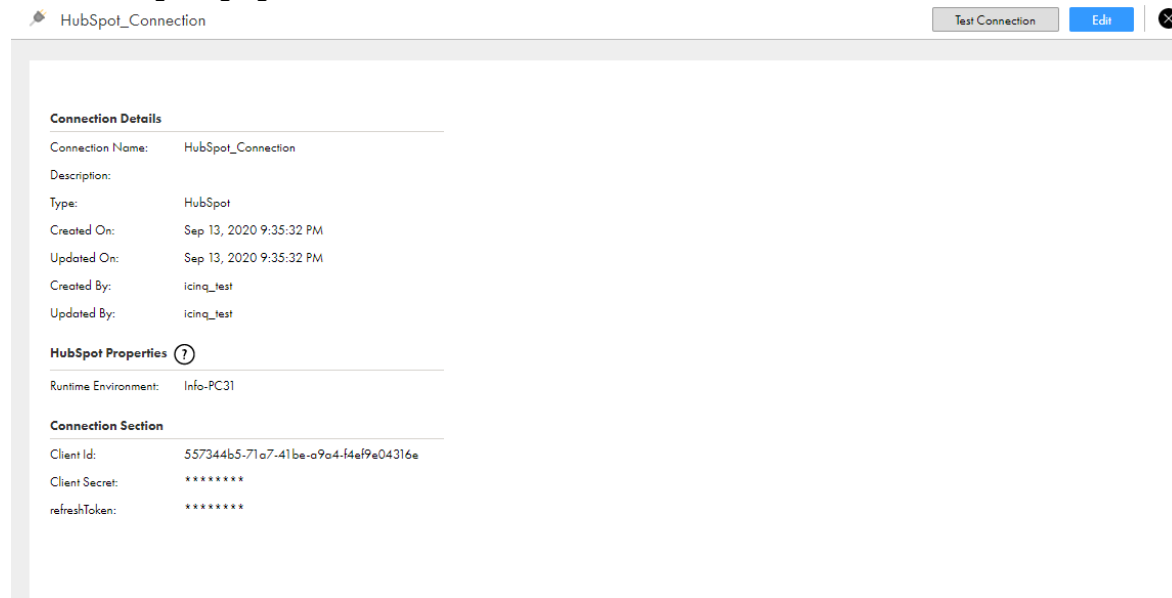
HubSpot connector has 3 connection attributes all of them are mandatory to create a HubSpot connection to access HubSpot data from Data Integration, and one is used to define the batch size to publish messages to a topic.

You can create a HubSpot connection on the Connections page. After you create a connection, it becomes available to all users who have access to the organization.

HubSpot Connection Properties

Connection Attributes	Description
Client Id	Specifies the Client Id value from the app
Client Secret	Specifies the Client Secret value from the app
Refresh Token	The refresh token received when the user authorized your app.
hapikey	Specifies the API key value from the app.(Mandatory for Quotes object).

The following image gives the connection information:



Chapter 3 - Mapping and Mapping Tasks with HubSpot

HubSpot Sources in Mapping Tasks

Contacts, Deals, Companies, Products, Line_Items and Tickets.

HubSpot Target in Mapping Tasks

Contacts, Deals, Companies, Products, Line_Items and Tickets.

Examples of HubSpot Mapping Tasks

Read Operation Mapping

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

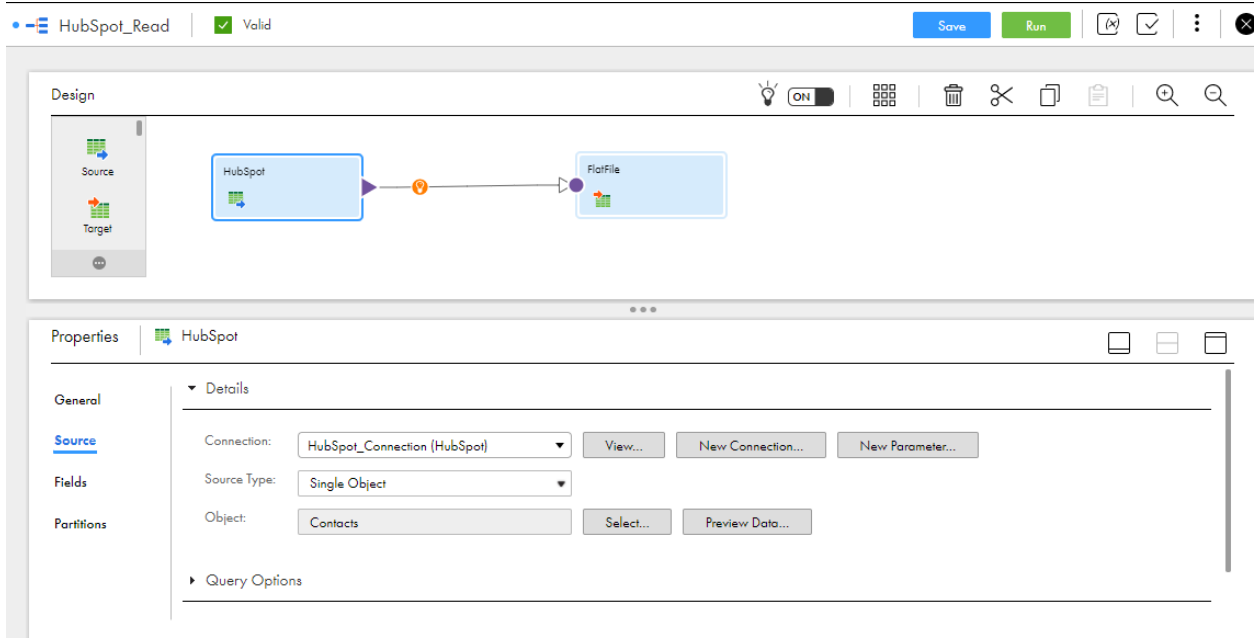
Select Source connection which you have created for HubSpot.

Source properties: -

Connection: HubSpot_Connection

Source type: Single Object

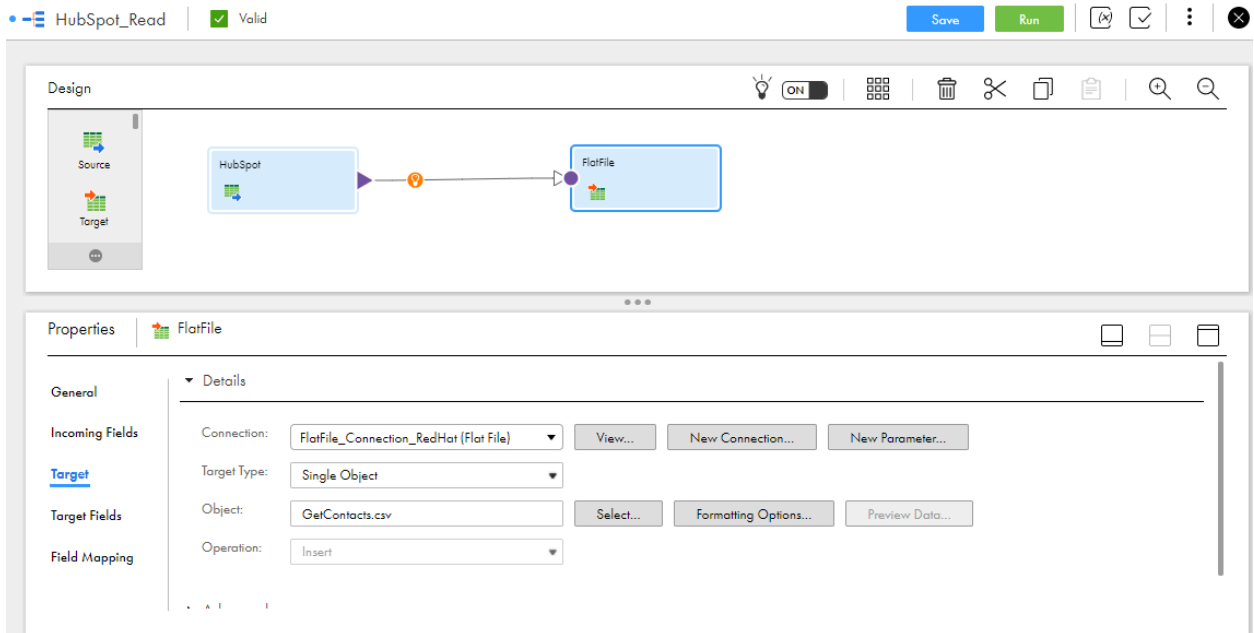
Object: Contacts



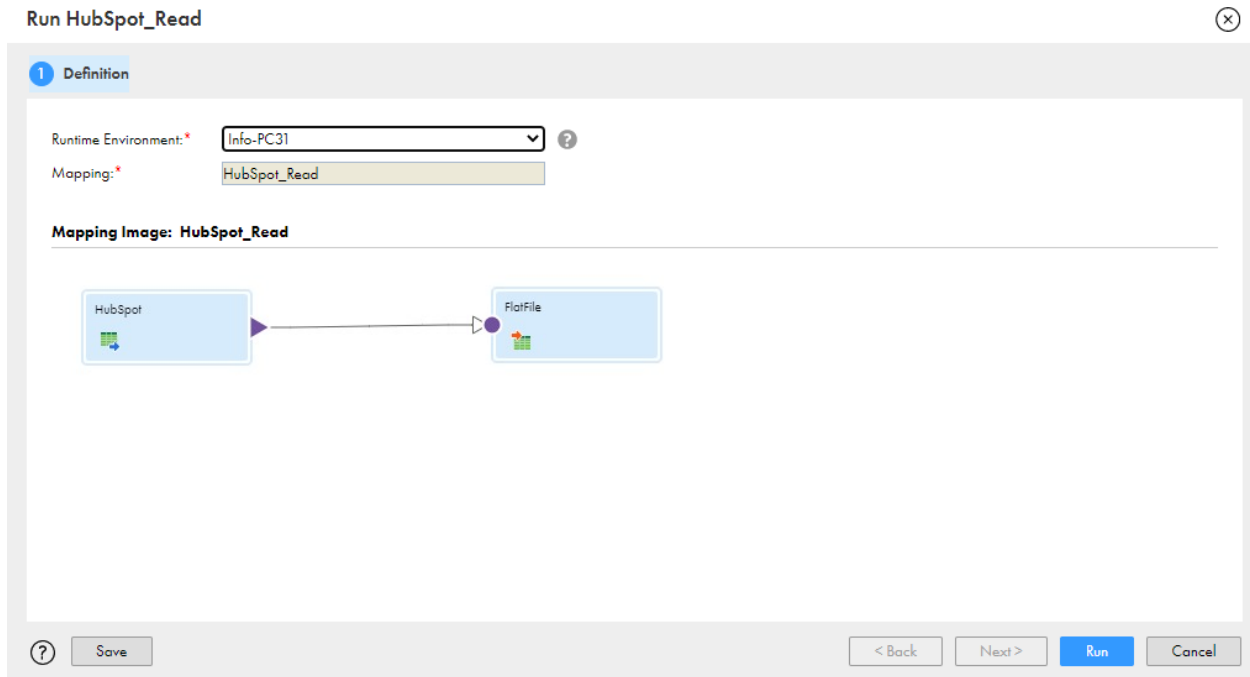
2. In Mapping for HubSpot Read operation, select target connection which you have created for HubSpot.

Target properties: -

- a. Connection: FlatFile_Connection
- b. Target type: Single Object
- c. Object: GetContacts.csv
- d. Operation: Insert

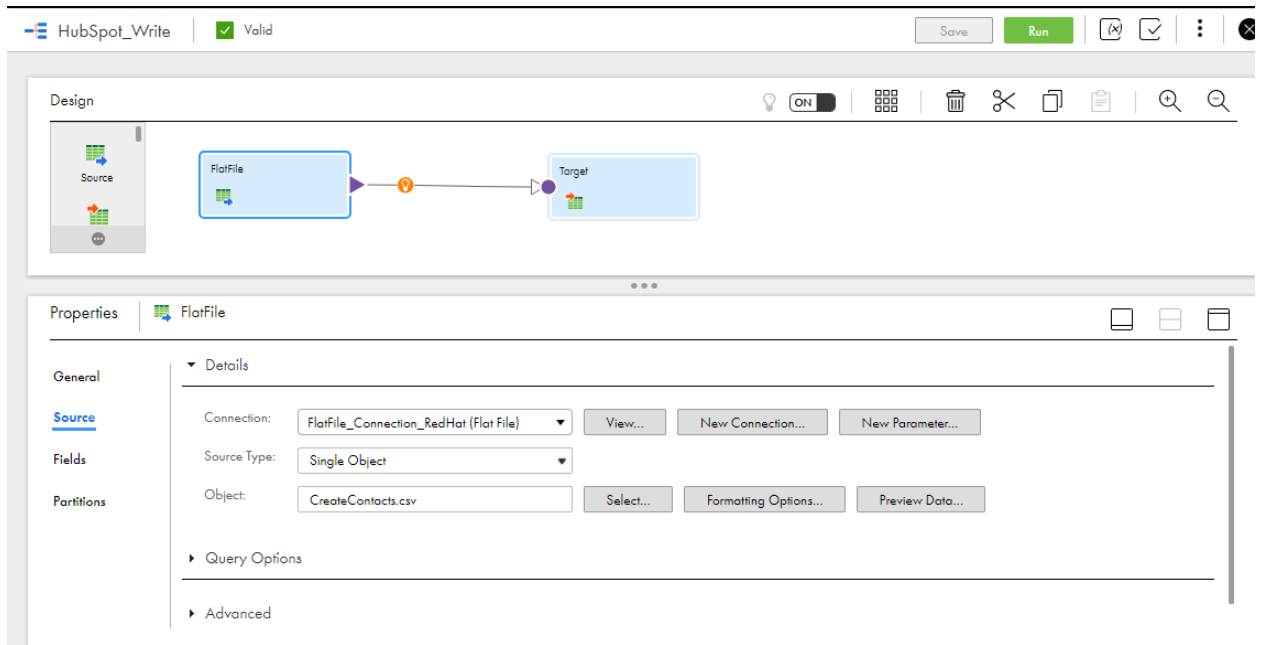


3. In Mapping configuration task for HubSpot Read operation, specify Definition task name as **HubSpot_Read**, Select Runtime environment and run the task.



Write Operation Mapping

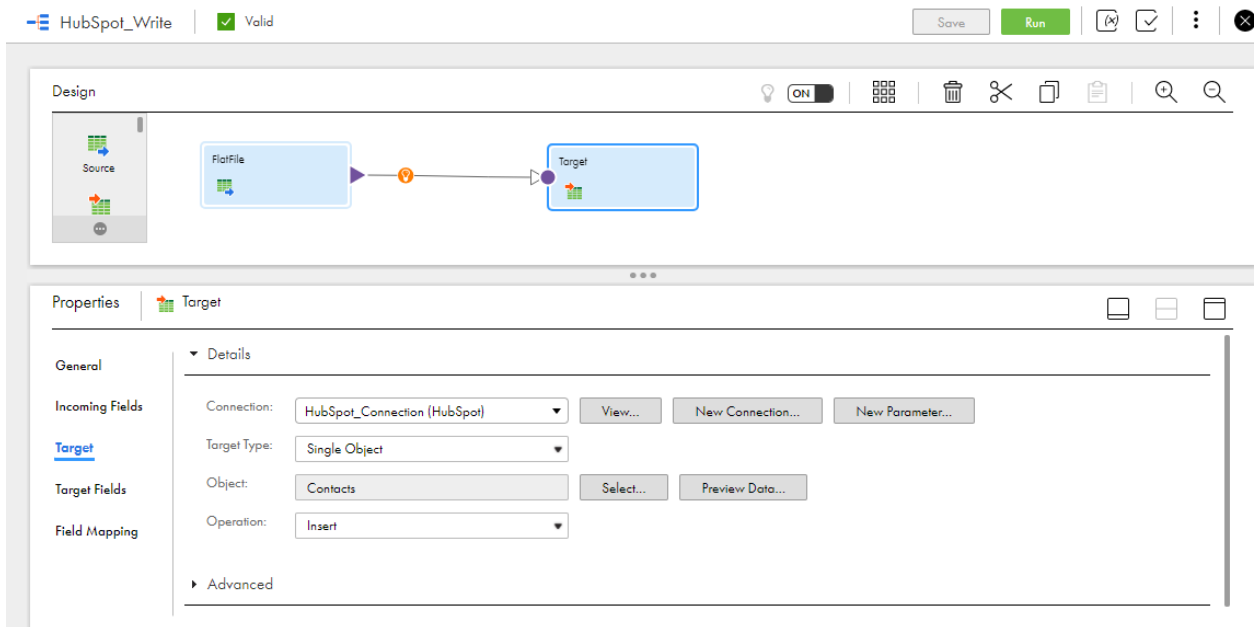
1. In Mapping for HubSpot Write operation, specify Task name as **HubSpot_Write**, take new Source, new Target and give Properties to Source and target. Select Source connection which you have created for HubSpot.
Source properties: -
 - a. connection: FlatFile_Connection(Flat File)
 - b. source type: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



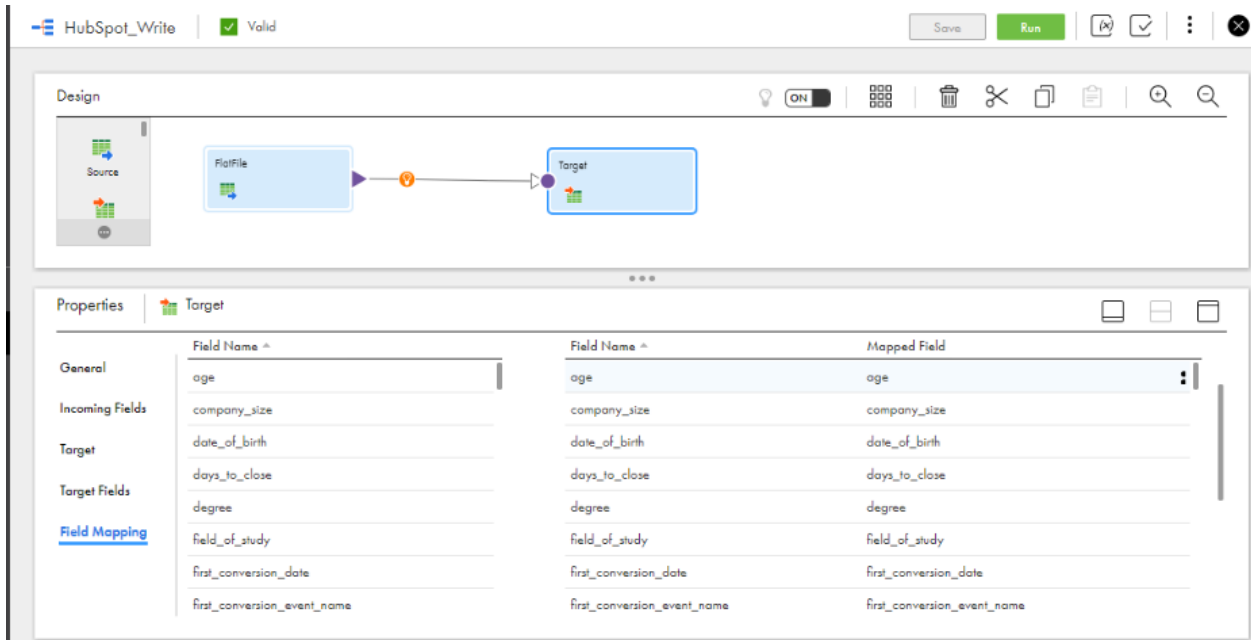
2. In Mapping for HubSpot Write operation, select target connection which you have created for HubSpot.

Target properties: -

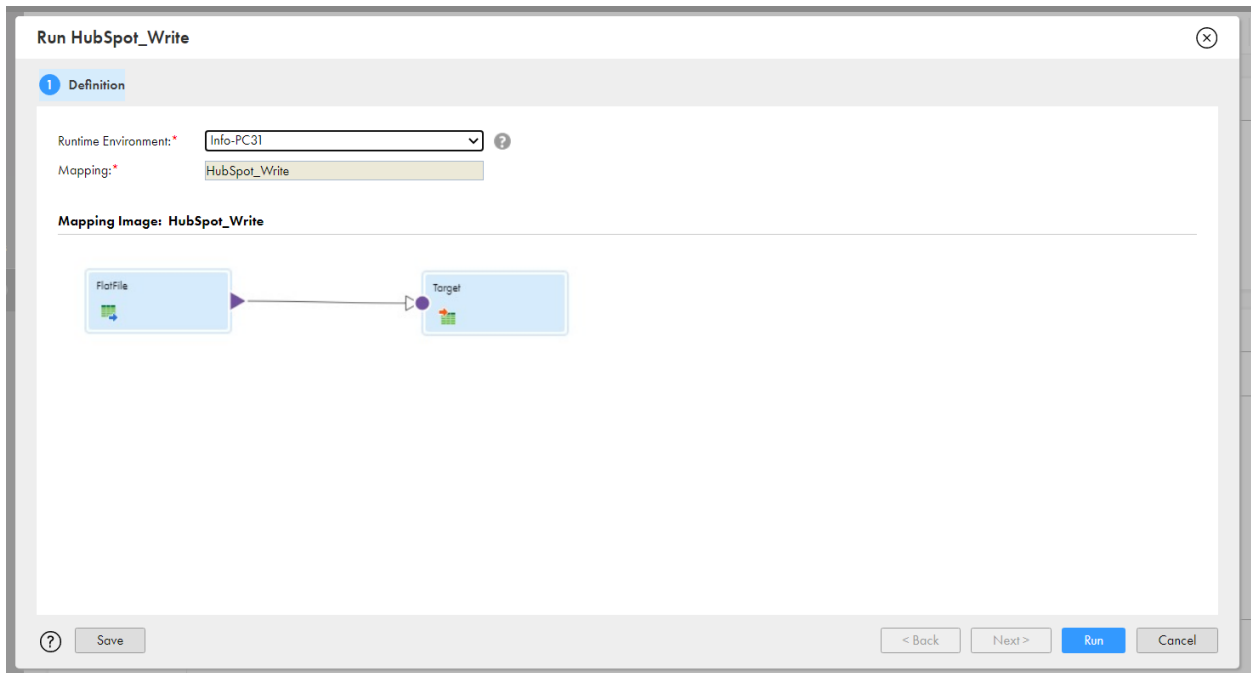
- a. Connection: HubSpot_Connection
- b. Target type: Single Object
- c. Object: Contacts
- d. Operation: Insert



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



4. In Mapping configuration task for HubSpot Read operation, specify Definition task name as **HubSpot_Write**, Select Run time environment and run the task.



Appendix: Data Type Reference

Data Type Reference Overview

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

HubSpot data types:

HubSpot 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.

HubSpot and Transformation Data Types

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

HubSpot 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 an 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	Pushdown optimization is skipped because session {sessionname} has row error logging enabled.	- Avoid offending the user. -

logging.

{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.

-
Provide as much information as possible.

-
Be concise and direct.

-
Do not start a message with a parameter.