

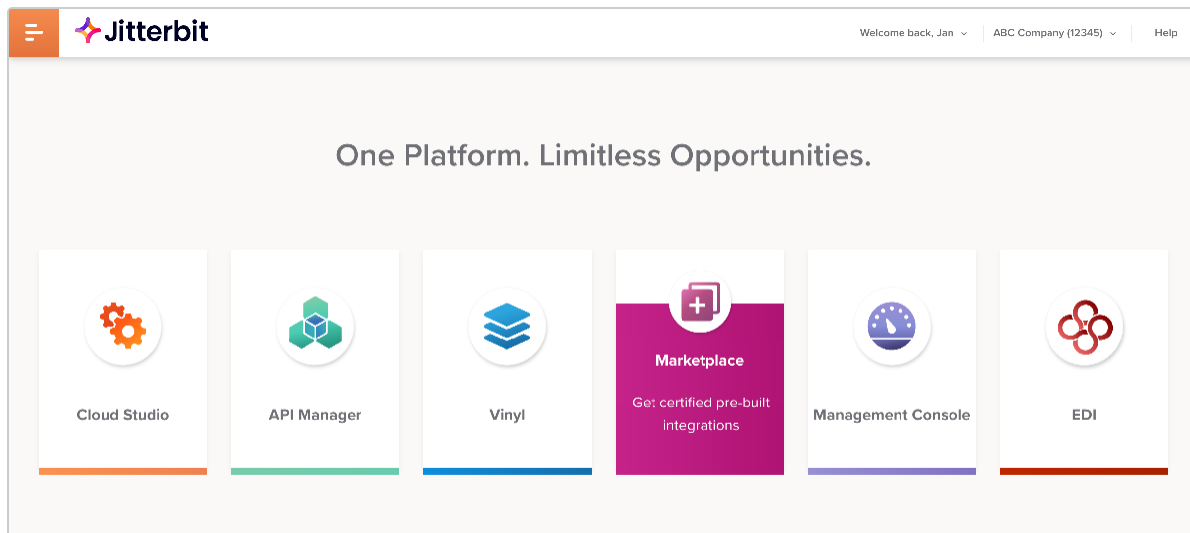
# Jitterbit's Harmony Marketplace

## Overview

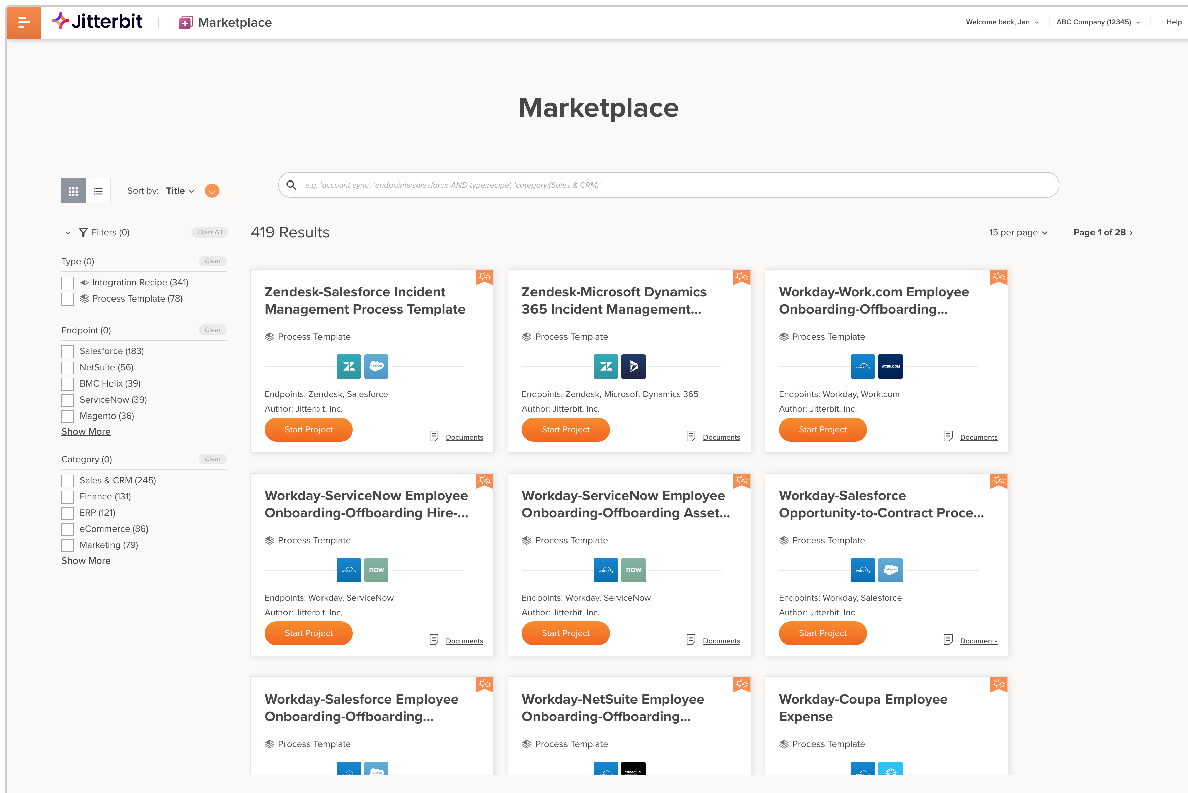
Jitterbit's Harmony Marketplace (Marketplace) is a repository of over 400 ready-made [Cloud Studio](#) integration project [templates](#).

Marketplace project templates are a quick and easy way to create Cloud Studio projects with pre-configured endpoints that connect pairs of systems from vendors such as Salesforce, NetSuite, Autodesk, Dynamics CRM, Zoho, Zendesk, Shopify, Magento, ConstantContact, DotMailer, ServiceNow, SAP, Clarizen, and others.

To open Marketplace, click its tile in the [Harmony Portal landing page](#), or select it from the [Harmony Portal menu](#):



When it first opens, Marketplace shows all available templates as pages of tiles:



Use the [search bar](#) and [filter panel](#) to find templates, and other features of the [user interface](#) to switch to [list view](#), and [sort](#) and [page](#) through templates.

## Template Types

There are two types of project templates in Marketplace:

- . **Integration Recipes:** These are single, pre-built integration projects that move data in one direction between objects across two applications or systems (such as syncing Salesforce contacts to Constant Contact contacts). All Harmony subscribers can use integration recipes.
- . **Process Templates:** These are pre-built integration use cases that speed up the deployment of specific business processes by 50 to 80 percent using various objects across multiple applications or systems. You can implement projects started with process templates yourself or have them delivered by Jitterbit Professional Services or an implementation partner.

Note that access to some process templates depends on your organization's subscriptions.

# Role Requirements

Any user whose role has Read or Admin permission can browse all of an organization's integration recipes and process templates. To use one, the role must also have Write access in the environment where you want to create the Cloud Studio project.

For more information on permissions and access levels, see [Harmony Permissions and Access](#).

# Search Expressions

## Introduction

The Marketplace [search bar](#) lets you build a search expression by combining [keywords](#) and [field-keyword](#) pairs with [operators](#).

## Keyword Search

To search by keyword, enter one or more words separated by spaces, then press return. Keyword searches are case-insensitive. To group words into a phrase, enclose them within double quotes.

With two or more words, the search results are templates with any of the words in them. To find templates with all of the words, separate search terms with the [AND](#) operator.

Use the wildcard character to match any single character. Use the wildcard Character to match any number of characters, including none.



Search expression	Search results
work <input type="text"/>	Templates where the word <code>work</code> occurs.
work item <input type="text"/>	Templates where either the words <code>work</code> or <code>item</code> occur.
"work item" <input type="text"/>	Templates where the phrase <code>work item</code> occurs.
work* <input type="text"/>	Templates where the title or description contains any word beginning with <code>work</code> (case insensitive). Matches <code>Workday</code> , <code>Work</code> .

## Field-Keyword Search

Enter a field name, a colon, an optional space, and a search keyword. To match multiple keywords, enclose them in parentheses. To match a group of words as a phrase, enclose them in double-quotes.

The following field names are available:

Field	Description
author <input type="text"/>	Match the author name.
name <input type="text"/>	Match the template name.
description <input type="text"/>	Match the template description.
endpoints <input type="text"/>	Match the endpoint name.
category <input type="text"/>	Match the category.
type <input type="text"/>	Match the type (recipe or template).



Except for `category`, all fields match against case-insensitive keywords, and can include wildcard characters.

Search expression	Search results
<code>endpoints:netsuite</code>	Templates using NetSuite endpoint (same as selecting NetSuite in the Endpoints filter).
<code>endpoints:s*</code>	Templates with endpoints beginning with <code>s</code> or <code>S</code> .
<code>type:recipe</code>	Templates of type <code>recipe</code> (same as selecting Integration Recipe in the Type filter).
<code>type:re*</code>	Templates with type beginning with <code>re</code> .
<code>name:(work item)</code>	Templates with either the words <code>work</code> or <code>item</code> in the title.
<code>name:"work item"</code>	Templates with the phrase <code>work item</code> in the title.
<code>NOT author:(bmc jitterbit)</code>	Templates with author containing neither of the words <code>bmc</code> or <code>jitterbit</code> .
<code>name:"work item"</code>	Templates with the phrase <code>work item</code> in the title.
<code>NOT author:(bmc jitterbit)</code>	Templates with author containing neither of the words <code>bmc</code> or <code>jitterbit</code> .

## Search Operators

The search operators are `AND`, `OR`, and `NOT`.


By default, a search of two or more search terms matches any of the terms—the `OR` operator is implied between each term. To match all search terms, separate them with the `AND` operator.

To invert a search term, add the operator keyword `NOT` before it.

Search bar expression	Search results
work AND item	Templates where both the words <code>work</code> and <code>item</code> occur in either the title or description.
work OR item	Identical search to <code>work item</code> . OR between terms is implied.
NOT "work item"	Templates where the phrase <code>work item</code> does not occur in neither the title nor the description.
NOT author:jitterbit	Templates not authored by Jitterbit.

# Snowflake Connector

## Summary

The Harmony Snowflake Connector establishes access to [Snowflake](#) .

The Snowflake connector provides an interface for creating a Snowflake connection, the foundation used for generating instances of Snowflake activities. These activities, once configured, interact with Snowflake through the connection.

The Snowflake connector is accessed from the design component palette's Connections tab (see [Design Component Palette](#)).

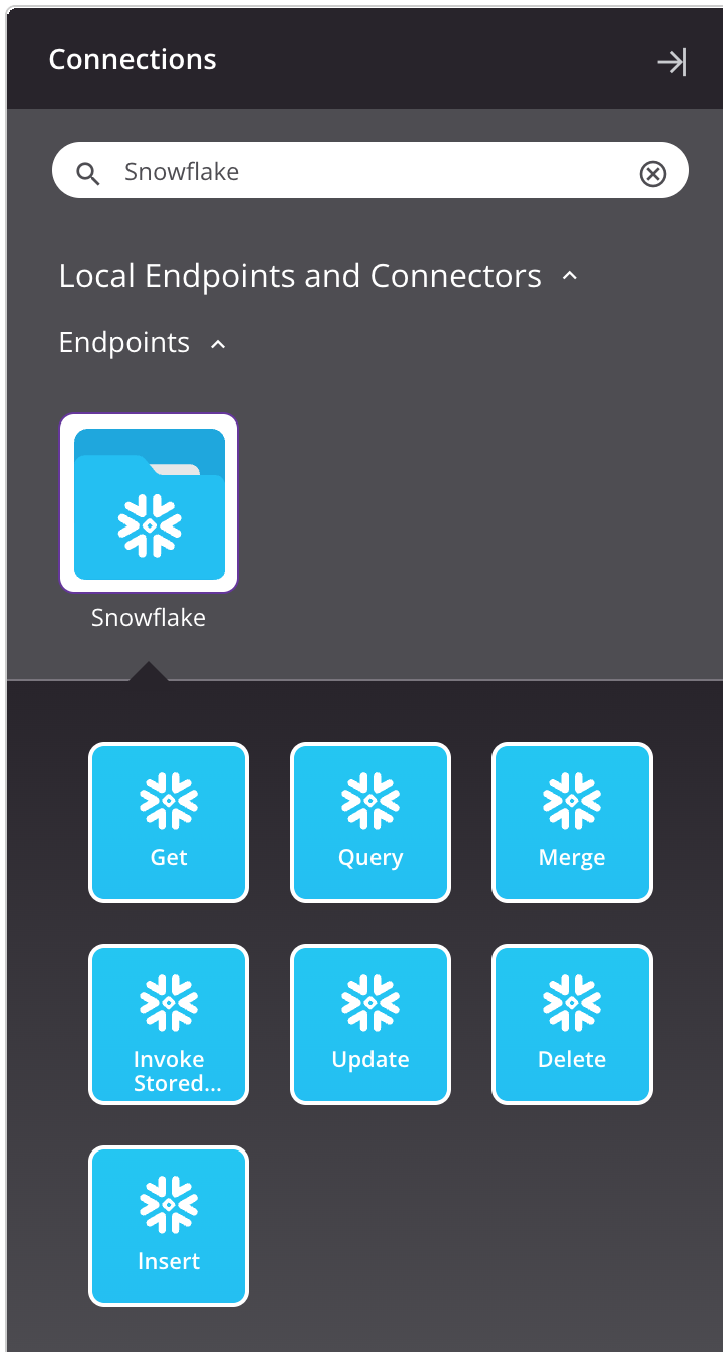
NOTE: Snowflake Developer Instances are known to "go to sleep" if you haven't recently logged in to the Snowflake UI or used the instance. If a table does not populate with available objects, first ensure that the Snowflake instance is active, and then ensure you are connected to it by reopening the Snowflake connection and retesting the credentials.

## Connector Overview

This connector is used to configure a [Snowflake connection](#). Activity types associated with that connection are then used to create instances of activities that are intended to be used as sources (to provide data in an operation) or targets (to consume data in an operation).

Together, a specific Snowflake connection and its activities are referred to as a Snowflake endpoint:





- ◆ **Get:** This function retrieves a CSV file of table or view data from Snowflake and is intended to be used as a source in an operation.
- ◆ **Query:** This function retrieves a CSV file of table or view data from Snowflake and is intended to be used as a source in an operation.

- [Merge](#): Inserts or updates a CSV of table data into Snowflake, which is intended to be used as a target in an operation.
- [Invoke Stored Procedure](#): This function invokes a stored procedure created in Snowflake that is intended to be used as a target in an operation.
- [Update](#): Updates table data in Snowflake and is intended to be used as a target in an operation.
- [Delete](#): Deletes table data and view data from Snowflake and is intended to be used as a target in an operation.

[Insert](#): Inserts table data (either as a CSV file or directly mapped to columns of a table) into Snowflake and is intended to be used as a target in an operation.

NOTE: This connector is a Connector SDK-based connector, which may be referred to by Jitterbit when communicating changes made to connectors built with the Connector SDK.

## Prerequisites and Supported API Versions

The Snowflake connector requires the use of an agent version [10.1](#) or later. These agent versions automatically download the latest version of the connector when required.

The Snowflake connector uses the [Snowflake JDBC Driver](#) [↗](#) and the [Snowflake SQL commands](#) [↗](#). Refer to the API documentation for information on the schema nodes and fields.


## Troubleshooting

If you experience issues with the Snowflake connector, these troubleshooting steps are recommended:

- . Click the Test button in the configuration to ensure the connection is successful and to download the latest version of the connector to the agent (unless using the Disable Auto Connector Update [organization policy](#)).
- . Check the [operation logs](#) for any information written during the execution of the operation.
- . Enable operation debug logging ([for Cloud Agents](#) or [Private Agents](#)) to generate additional log files and data.
- . If using Private Agents, you can check the [agent logs](#) for more information.

# Snowflake Connection

## Introduction

A Snowflake connection, created using the [Snowflake connector](#), establishes access to [Snowflake](#) . Once a connection is configured, you can create instances of

Snowflake activities associated with that connection to be used either as sources (to provide data in an operation) or as targets (to consume data in an operation).

NOTE: This connector supports the Enable Re-authentication on Change organization policy. If enabled, a change to the Host Name, Username, Account, or Database Name in this connection requires users to re-enter the Password for the connection.

## Create or Edit a Snowflake Connection

A new Snowflake connection is created using the Snowflake connector from one of these locations:

- The design component palette's Connections tab (see [Design Component Palette](#)).
- The Global Connections page (see [Create a Global Connection](#) in [Global Connections](#)).

An existing Snowflake connection can be edited from these locations:

- The design component palette's Connections tab (see [Design Component Palette](#)).
- The project pane's Components tab (see [Component Actions Menu](#) in [Project Pane Components Tab](#)).

- The Global Connections page (see [Edit a Global Connection](#) in [Global Connections](#)).

## Configure a Snowflake Connection

Each user interface element of the Snowflake connection configuration screen is described below.

# Configure Snowflake Connection

CONNECTION NAME\*

Snowflake

HOST NAME\*

eve58367.snowflakecomputing.com

USER NAME\*

eve@jbexample.com

PASSWORD\*

.....

ACCOUNT\*

eve58367

DATABASE NAME\*

DEMO\_DB

SCHEMA NAME\*

PUBLIC

▼ OPTIONAL SETTINGS

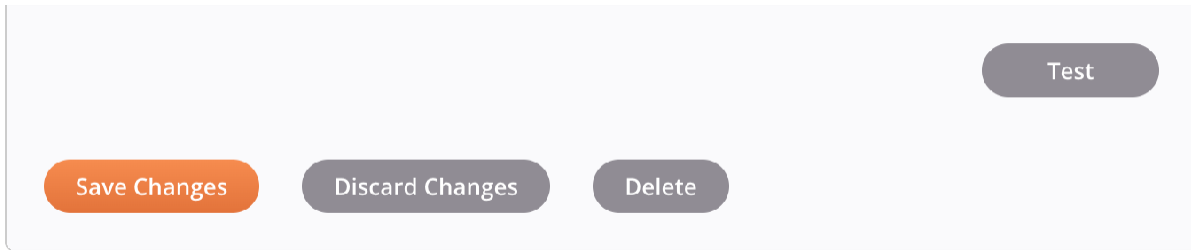
Role

GENERAL

Warehouse Name

compute\_wh

Use quote for Snowflake identifiers



TIP: Fields with a variable icon support using global variables, project variables, and Jitterbit variables. Begin either by typing an open square bracket into the field or by clicking the variable icon to display a list of the existing variables to choose from.

- **Connection Name:** Enter a name to use to identify the connection. The name must be unique for each Snowflake connection and must not contain forward slashes or colons. This name is also used to identify the Snowflake endpoint, which refers to both a specific connection and its activities.
- **Host Name:** Enter the Snowflake host name. This is the Snowflake account name (such as `eve58367`) and the Snowflake domain name of `snowflakecomputing.com` concatenated together, such as `eve58367.snowflakecomputing.com`.
- **User Name:** Enter the Snowflake user name.
- **Password:** Enter the Snowflake password.
- **Account:** Enter the Snowflake account name.
- **Database Name:** Enter the Snowflake database name.
- **Schema Name:** Enter the Snowflake schema name.
- **Optional Settings:** Click to expand additional optional settings:
  - **Role:** Enter the Snowflake role. The available roles are GENERAL, PUBLIC, and READONLY.
  - **Warehouse Name:** Enter the Snowflake warehouse name.

- **Use Quote for Snowflake Identifiers:** Select this option to use quotes around Snowflake identifiers such as table names to preserve their case instead of converting their case to uppercase, which can cause an error. By default, this is selected.
- **Test:** Click to verify the connection using the provided configuration. When the connection is tested, the latest version of the connector is downloaded by the agent(s) in the agent group associated with the current environment. This connector supports suspending the download of the latest connector version by using the [Disable Auto Connector Update organization policy](#).
- **Save Changes:** Click to save and close the connection configuration.
- **Discard Changes:** After making changes to a new or existing configuration, click to close the configuration without saving. A message asks you to confirm that you want to discard changes.
- **Delete:** After opening an existing connection configuration, click to permanently delete the connection from the project and close the configuration (see [Component Dependencies, Deletion, and Removal](#)). A message asks you to confirm that you want to delete the connection.

## Next Steps

After a Snowflake connection has been created, you place an activity type on the design canvas to create activity instances to be used either as sources (to provide data in an operation) or as targets (to consume data in an operation).

Menu actions for a connection and its activity types are accessible from the project pane and design component palette. For details, see [Actions Menus](#) in [Connector Basics](#).

These activity types are available:

- **Get:** Retrieves a CSV file of table or view data from Snowflake and is intended to be used as a source in an operation.



- **Query**: Retrieves a CSV file of table or view data from Snowflake and is intended to be used as a source in an operation.
- **Merge**: Inserts or updates a CSV file of table data into Snowflake and is intended to be used as a target in an operation.
- **Invoke Stored Procedure**: Invokes a stored procedure created in Snowflake and is intended to be used as a target in an operation.
- **Update**: Updates table data in Snowflake and is intended to be used as a target in an operation.
- **Delete**: Deletes table data and view data from Snowflake and is intended to be used as a target in an operation.
- **Insert**: Inserts table data (either as a CSV file or directly mapped to columns of a table) into Snowflake and is intended to be used as a target in an operation.

# Snowflake Query Activity

## Introduction

A Snowflake Query activity uses the Snowflake connection to retrieve a CSV file of table or view data from Snowflake and is intended to be used as a source to provide data in an operation.

NOTE: Depending on your Snowflake permissions, the Query activity will return only data that you have access to. For example, if you have permissions only for view data, then the Get activity will return only view data and will not return table data.

## Create a Snowflake Query Activity

An instance of a Snowflake Query activity is created from a [Snowflake connection](#) using its Query activity type.

To create an instance of an activity, drag the activity type to the design canvas or copy the activity type and paste it on the design canvas. For details, see [Creating an Activity Instance](#) in [Component Reuse](#).

An existing Snowflake Query activity can be edited from these locations:

- The design canvas (see [Component Actions Menu](#) in [Design Canvas](#)).
- The project pane's Components tab (see [Component Actions Menu](#) in [Project Pane Components Tab](#)).

# Configure a Snowflake Query Activity

Follow these steps to configure a Snowflake Query activity:

- [Step 1: Enter a Name and Select an Object](#)  
Provide a name for the activity and select an object.
- [Step 2: Build Your Query](#)  
Set conditions on a query using the object fields and apply paging to a query.
- [Step 3: Review the Data Schemas](#)  
Any request or response schemas generated from the endpoint are displayed.

## Step 1: Enter a Name and Select an Object

In this step, provide a name for the activity and select a table or view (see Snowflake's [Overview of Views](#)). Each user interface element of this step is described below.

Snowflake

## Query

1 2 3

NAME\*


Select an Object *Click **Refresh** to retrieve a complete list of objects*

Selected Snowflake Object: NAME  Refresh

Name	Type
MULTIPLE_TYPES	Table
MarchTests	Table
NAME	Table
PERFORMTEST	Table
QAPERFORM	Table

Save & Exit Next Discard Changes

- ♦ **Name:** Enter a name to identify the activity. The name must be unique for each Snowflake Query activity and must not contain forward slashes or colons.
- ♦ **Select an Object:** This section displays objects available in the Snowflake endpoint. When reopening an existing activity configuration, only the selected object is displayed instead of reloading the entire object list.
  - ♦ **Selected Snowflake Object:** After an object is selected, it is listed here.
  - ♦ **Search:** Enter any part of the object name into the search box to filter the list of objects. The search is not case-sensitive. If objects are already displayed within the table, the table results are filtered in real time with each keystroke. To reload objects from the endpoint when searching, enter search criteria and then refresh, as described below.

- Refresh: Click the refresh icon  or the word Refresh to reload objects from the Snowflake endpoint. This may be useful if objects have been added to Snowflake. This action refreshes all metadata used to build the table of objects displayed in the configuration.
- Selecting an Object: Within the table, click anywhere on a row to select an object. Only one object can be selected. The information available for each object is fetched from the Snowflake endpoint:

- Name: The name of an object, either a table or a view.

Type: The type of the object, either a table or a view.

TIP: If the table does not populate with available objects, the Snowflake connection may not be successful. Ensure you are connected by reopening the connection and retesting the credentials.

- Save & Exit: If enabled, click to save the configuration for this step and close the activity configuration.
- Next: Click to temporarily store the configuration for this step and continue to the next step. The configuration will not be saved until you click the Finished button on the last step.
- Discard Changes: After making changes, click to close the configuration without saving changes made to any step. A message asks you to confirm that you want to discard changes.

## Step 2: Build Your Query

In this step, set conditions on a query using the object fields and apply paging to a query. Each user interface element of this step is described below.

Snowflake

## Query

1 2 3

### Build Your Query

Test Query

FIELDS TO RETRIEVE

Search for an object

Select All

NAME	Data Type
<input checked="" type="checkbox"/> COLOR	[VARCHAR(16777216)]
<input checked="" type="checkbox"/> FLOWER	[VARCHAR(16777216)]

PAGING

Limit  Offset

Apply Remove

CONDITIONS

Object: Field  Operator

Value

Add Remove All

QUERY STRING

```
SELECT "COLOR", "FLOWER" FROM "NAME"
```

Use Static Schema

< Back Next Discard Changes

**TIP:** Fields with a variable icon support using global variables, project variables, and Jitterbit variables. Begin either by typing an open square bracket into the field or by clicking the variable icon to display a list of the existing variables to choose from.

**Search:** Enter any part of a field name into the search box to filter the list of fields for the selected object. The search is not case-sensitive. The listed results are filtered in real time with each keystroke.

**Refresh:** Click the refresh icon or the word Refresh to reload fields of the object from the Snowflake endpoint.

- **Select All:** When using the search box to filter, you can use this checkbox to select all visible fields at once.
- **Select Fields:** Select the checkboxes of the fields you want included in the query to have them automatically added to the SELECT statement in the Query String. You can also Select All of the fields at once using the checkbox.
- **Paging:** To add a paging clause (a limit on the number of records with an optional record offset), you can use the dropdown to set the paging limit and the field to enter an offset. If an offset is not specified, it defaults to 0. A single paging clause is supported. If the paging clause is not included, all records are returned.
  - **Apply:** Click to automatically construct the clause based on the dropdown selections and entered value. The automatically constructed paging clause appears in the Query String text box.
  - **Remove:** Click to remove a paging clause that has been applied.
- **Conditions:** To add conditional clauses, use the fields below as input to help construct the clauses, which then appear in the Query String text box.
  - **Object: Field:** Use the dropdown to select a field from the selected object.
  - **Operator:** Use the dropdown to select an operator that is appropriate for the field data type:

Operator	Label	Description
=	Equals	
!=	Not equals	
IN (value1, value2)	In	In list of values.
LIKE 'string'	Like	Like string.
LIKE 'string%'	Starts with	Starts with string.
LIKE '%string'	Ends with	Ends with string.

Operator	Label	Description
LIKE '%string%'	Contains	Contains string.
<	Less than	
<=	Less or equal	
>	Greater than	
>=	Greater or equal	

- Value: Enter the desired value to use with the dropdown selections.

**TIP:** When using global variables in the conditions of a WHERE clause, it is recommended to specify a default value so that script testing is possible. Otherwise, since global variables obtain their value at runtime, the syntax may be invalidated during testing or no data may be returned if no default value is specified.

- Add: Click to automatically construct the clause based on the dropdown selections and entered value. The conditional clause is added to the Query String text box.
- Remove All: Click to remove all entered conditional clauses.

**Query String:** As you select fields, specify conditions, and set paging, the query statement in this text box is autopopulated with the selected fields, conditions, and paging limits. This statement is editable.

**IMPORTANT:** To use a query statement that contains elements other than those selected via this interface, you must also select the Use Static Schema checkbox.



- **Use Static Schema:** Select to set the response schema to a static schema that is independent of the query statement. This allows custom query statements entered in the Query String text box to be used.
- **Test Query:** Click to validate the query. This button must be clicked in order to enable the Next button.
- **Back:** Click to temporarily store the configuration for this step and return to the previous step.
- **Next:** Click to temporarily store the configuration for this step and continue to the next step. The configuration will not be saved until you click the Finished button on the last step.
- **Discard Changes:** After making changes, click to close the configuration without saving changes made to any step. A message asks you to confirm that you want to discard changes.

### Step 3: Review the Data Schemas

Any request or response schemas generated from the endpoint are displayed. Each user interface element of this step is described below.

Snowflake

## Query

1 — 2 — 3

Data Schema:

Request Refresh

queryRequest

- [ 0+ ] whereClause
  - [ 0, 1 ] columnName (String)
  - [ 0, 1 ] condition (String)
  - [ 0, 1 ] value (String)

NAME

- [ 0+ ] Entity
  - [ 0, 1 ] COLOR (String)
  - [ 0, 1 ] FLOWER (String)

[Back](#) Finished Discard Changes

Data Schemas: These data schemas are inherited by adjacent transformations and are displayed again during [transformation mapping](#).

NOT: Data supplied in a transformation takes precedence over the activity configuration.

The Snowflake connector uses the [Snowflake JDBC Driver](#) and the [Snowflake SQL commands](#). Refer to the API documentation for information on the schema nodes and fields.

The default request and response data schemas consist of these nodes and fields:

- ◆ Request

Request Schema Field/Node	Notes
<code>queryRequest</code>	Node of the query request.
<code>whereClause</code>	Node of the WHERE clause request.
<code>columnName</code>	Column name of the WHERE clause.
<code>condition</code>	Condition of the WHERE clause.
<code>value</code>	Value of the WHERE clause.

- Response


Response Schema Field/Node	Notes
<code>table</code>	Node showing the table name NAME.
<code>Entity</code>	Node of the entity.
<code>column_A</code>	Value for the first column, COLOR.
<code>column_B</code>	Value for the second column, FLOWER.
<code>...</code>	Values for the succeeding table columns.

When Use Static Schemas is selected, the response data schema consists of these nodes and fields:

- Response

Response Schema Field/Node	Notes
<code>queryResponse</code>	Node of the query response.

Response Schema Field/Node	Notes
<code>row</code>	Node of the response row.
<code>item</code>	Node of the response item.
<code>columnName</code>	Column name of the response item.
<code>columnValue</code>	Column value of the response item.

- **Refresh:** Click the refresh icon  or the word Refresh to regenerate schemas from the Snowflake endpoint. This action also regenerates a schema in other locations throughout the project where the same schema is referenced, such as in an adjacent transformation.
- **Back:** Click to temporarily store the configuration for this step and return to the previous step.
- **Finished:** Click to save the configuration for all steps and close the activity configuration.
- **Discard Changes:** After making changes, click to close the configuration without saving changes made to any step. A message asks you to confirm that you want to discard changes.

## Next Steps

After configuring a Snowflake Query activity, complete the configuration of the operation by adding and configuring other activities, transformations, or scripts as operation steps. You can also configure the operation settings, which include the ability to chain operations together that are in the same or different workflows.

Menu actions for an activity are accessible from the project pane and the design canvas. For details, see [Activity Actions Menu](#) in [Connector Basics](#).

Snowflake Query activities can be used as a source with these [operation patterns](#):

- Transformation Pattern
- Two-target Archive Pattern (as the first source only)
- Two-target HTTP Archive Pattern (as the first source only)
- Two-transformation Pattern (as the first or second source)

To use the activity with scripting functions, write the data to a temporary location and then use that temporary location in the scripting function.

When ready, [deploy and run](#) the operation and validate behavior by checking the [operation logs](#).

# Snowflake Get Activity

## Introduction

A Snowflake Get activity, using its [Snowflake connection](#), retrieves a CSV file of table or view data from Snowflake and is intended to be used as a source to provide data in an operation.

NOTE: Depending on your Snowflake permissions, the Get activity will return only data that you have access to. For example, if you have permissions only for view data, then the Get activity will return only view data and will not return table data.

## Create a Snowflake Get Activity

An instance of a Snowflake Get activity is created from a [Snowflake connection](#) using its Get activity type.

To create an instance of an activity, drag the activity type to the design canvas or copy the activity type and paste it on the design canvas. For details, see [Creating an Activity Instance in Component Reuse](#).

An existing Snowflake Get activity can be edited from these locations:


- The design canvas (see [Component Actions Menu in Design Canvas](#)).
- The project pane's Components tab (see [Component Actions Menu in Project Pane Components Tab](#)).

# Configure a Snowflake Get Activity

Follow these steps to configure a Snowflake Get activity:

- [Step 1: Enter a Name and Select an Object](#)  
Provide a name for the activity and select an object, either a table or a view.
- [Step 2: Review the Data Schemas](#)  
Any request or response schemas generated from the endpoint are displayed.

## Step 1: Enter a Name and Select an Object

In this step, provide a name for the activity and select a table or view (see Snowflake's [Overview of Views](#) ). Each user interface element of this step is described below.

Snowflake

## Get

1 — 2

NAME\*

Get

Select an Object Click **Refresh** to retrieve a complete list of objects

Selected Snowflake Object: NAME

Name	Type
HIMANSU_SHARMA	Table
JFARIS_TEST	Table
MarchTests	Table
<b>NAME</b>	<b>Table</b>
PERFORMTEST	Table


NULLIF

FIELD OPTIONALLY ENCLOSED BY

**TIP:** Fields with a variable icon support using [global variables](#), [project variables](#), and [Jitterbit variables](#). Begin either by typing an open square bracket into the field or by clicking the variable icon to display a list of the existing variables to choose from.

**Name:** Enter a name to identify the activity. The name must be unique for each Snowflake Get activity and must not contain forward slashes or colons.



- **Select an Object:** This section displays objects available in the Snowflake endpoint. When reopening an existing activity configuration, only the selected object is displayed instead of reloading the entire object list.
  - **Selected Snowflake Object:** After an object is selected, it is listed here.
  - **Search:** Enter any part of the object name into the search box to filter the list of objects. The search is not case-sensitive. If objects are already displayed within the table, the table results are filtered in real time with each keystroke. To reload objects from the endpoint when searching, enter search criteria and then refresh, as described below.
  - **Refresh:** Click the refresh icon  or the word Refresh to reload objects from the Snowflake endpoint. This may be useful if objects have been added to Snowflake. This action refreshes all metadata used to build the table of objects displayed in the configuration.
  - **Selecting an Object:** Within the table, click anywhere on a row to select an object. Only one object can be selected. The information available for each object is fetched from the Snowflake endpoint:
    - **Name:** The name of an object, either a table or a view.
    - **Type:** The type of the object, either a table or a view.

**TIP:** If the table does not populate with available objects, the Snowflake connection may not be successful. Ensure you are connected by reopening the connection and retesting the credentials.

- **NullIf:** The value to be returned if a result is `NULL`.
- **Field Optionally Enclosed By:** The optional character used to enclose the NullIf field.
- **Save & Exit:** If enabled, click to save the configuration for this step and close the activity configuration.

- Next: Click to temporarily store the configuration for this step and continue to the next step. The configuration will not be saved until you click the Finished button on the last step.
- Discard Changes: After making changes, click to close the configuration without saving changes made to any step. A message asks you to confirm that you want to discard changes.

## Step 2: Review the Data Schemas

Any request or response schemas generated from the endpoint are displayed. Each user interface element of this step is described below.

Snowflake

### Get

1 2

Data Schema:

Request Refresh

```
▼ getRequest
  [ 0, 1 ] stageName (String)
  [ 0, 1 ] fileName (String)
  [ 0, 1 ] compressData (Boolean)
  ▼ [ 0, 1 ] fileFormat
    [ 0, 1 ] nullIf (String)
    [ 0, 1 ] FIELD_OPTIONALLY_ENCLOSED_BY (String)
```

```
▼ getResponse
  [ 0, 1 ] fileName (String)
  [ 0, 1 ] content (String)
```

◀ Back Finished Discard Changes

- Data Schemas: These data schemas are inherited by adjacent transformations and are displayed again during [transformation mapping](#).

NOTE: Data supplied in a transformation takes precedence over the activity configuration.

The Snowflake connector uses the [Snowflake JDBC Driver](#) and the [Snowflake SQL commands](#). Refer to the API documentation for information on the schema nodes and fields.

The request and response data schemas consist of these nodes and fields:

- Request

Request Schema Field/Node	Notes
<code>stageName</code>	Name of the <a href="#">internal Snowflake stage</a> used while getting table data.  NOTE: The stageName is the name of an internal Snowflake staging area, which needs to be pre-existing before running an operation using this activity. It must be copied to the staging area before it is available to the connector.
<code>fileName</code>	Name of the file on the <a href="#">internal Snowflake stage</a> to be used while getting table data.
<code>compressData</code>	Boolean ag for whether to compress the data before uploading it to the <a href="#">internal Snowflake stage</a> .
Request Schema Field/Node	Notes
<code>fileFormat</code>	Node of the file format.

nullif	Display value for NULLIF.
FIELD_OPTIONALLY_ENCLOSED_BY	Delimiter used to enclose the NULLIF entry.

- Response

Response Schema Field/Node	Notes
fileName	Name of the file on the <a href="#">internal Snowflake stage</a> used while getting table data (identical to request).
content	Content of the Snowflake table in CSV format.

- Refresh: Click the refresh icon or the word Refresh to regenerate schemas from the Snowflake endpoint. This action also regenerates a schema in other locations throughout the project where the same schema is referenced, such as in an adjacent transformation.
- Back: Click to temporarily store the configuration for this step and return to the previous step.
- Finished: Click to save the configuration for all steps and close the activity configuration.
- Discard Changes: After making changes, click to close the configuration without saving changes made to any step. A message asks you to confirm that you want to discard changes.

## Next Steps

After configuring a Snowflake Get activity, complete the configuration of the operation by adding and configuring other activities, transformations, or scripts as operation steps. You can also configure the operation settings, which include the ability to chain operations together that are in the same or different workflows.

Menu actions for an activity are accessible from the project pane and the design canvas. For details, see [Activity Actions Menu](#) in [Connector Basics](#).

Snowflake Get activities can be used as a source with these [operation patterns](#):

- [Transformation Pattern](#)
- [Two-target Archive Pattern](#) (as the first source only)
- [Two-target HTTP Archive Pattern](#) (as the first source only)
- [Two-transformation Pattern](#) (as the first or second source)

To use the activity with scripting functions, write the data to a temporary location and then use that temporary location in the scripting function.

When ready, [deploy and run](#) the operation and validate behavior by checking the [operation logs](#).