

Application Control Framework Sample Data Enrichment Demo Setup Guide

Version 1
<AUTHOR>
<DATE>

Revision Summary

Date	Revision History	Comments
<REV_DATE>	1.0	Initial Version

Table of Contents

Prerequisites	4
Disclaimer	4
Key Native App Components	5
Application Logic	5
Multiple Native App Modes	5
Setup Script	6
Manifest File	6
Readme	6
Sidecar Loader	6
App Control Manager	7
Build and Deploy Demo Native App	9
Step 1: Build the Demo Native App	9
Part 1: Create the Application Package	9
Part 2: Create/Patch a Version for the Application Package	11
Part 3: Release Patch	16
Step 2: Create the Native App Listing	17
Step 3: Install the Native App in the Consumer Account	18
Step 4: Run the Scripts from the Native App's Readme	18
Additional Test Scenarios	24
Reach the Five Requests Limit Then Increase the Limit	24
Attempt to Exceed Limit	24
Increase the Limit	25
Make a New Request	26
Consumer Usage of the App is Blocked, when Disabled	26
Disable the Consumer	26
Attempt to Exceed Limit	28
Re-enable the Consumer	28

Prerequisites

- The Events and ACF accounts have been configured, per the Deployment Guide
- The ACF Demo scripts have been executed, per the Deployment Guide
- The user that will build and deploy the native app must be granted the `P_<APP_CODE>_ACF_ADMIN` role.
- For demo purposes, a demo consumer account should be created in the same organization as the provider's account.

NOTE: `<APP_CODE>` = an abbreviated representation for the name of the app (e.g. `SDE` for Sample Data Enrichment)

Disclaimer

Any screenshots included in this guide are examples. Please refer to the text in the steps below when installing this app.

Key Native App Components

Application Logic

Application logic is the critical component of the native app. It provides the intended functionality the consumer executes in their Snowflake account. Application logic is deployed in the form of one or more stored procedures. Apps built by the ACF can support multiple functionality/procedures and allows the provider the ability to control which consumers get access to what functionality.

For this demo, the application logic resides in the ENRICH stored procedure. The instructions later in this document will build a native app that includes this stored procedure.

Multiple Native App Modes

The framework comes with the ability to build a native app with custom functionality, depending on the version of the app. For example, the consumer can evaluate the “free” version of the app from the Snowflake Marketplace, without interaction from the provider. Once the consumer is interested in the one or more paid versions of the app, they can be granted access to the desired version.

By default, the ACF supports three app modes:

- **FREE:** a free version of the app that is publicly available in the Snowflake Marketplace. This version offers limited functionality, meant to entice the consumer to convert to a paid version of the app. Each consumer of this app version has the same entitlements/limits (i.e. five requests).
- **PAID:** a paid version of the app that is publicly available in the Snowflake Marketplace. This version offers more or complete app functionality. Each consumer of this app has the same entitlements/limits (if any) enforced (i.e. process 1MM records every 30 days).
- **ENTERPRISE:** a version of the app where unique entitlements/limits can be set for each consumer. The entitlements/limits are managed via the ACF's App Control Manager. This is ideal for providers that want to create custom deals with consumers where the default entitlements/limits of the other app versions are not ideal for the consumer. Enterprise versions of the app should be listed privately and only made available to a single consumer.

For the purposes of this demo, the **FREE** version of the native app will be built.

Setup Script

The setup script contains SQL statements, including application logic DDL, that are executed when the consumer installs or upgrades an application or when a provider installs or upgrades an application for testing. Every application must contain a setup script. The setup script defines the objects that are created when an application is installed or upgraded. For more information, visit <https://docs.snowflake.com/en/developer-guide/native-apps/creating-setup-script>.

The ACF includes a setup script template that is used to construct each app version/patch's setup script. The ACF's App Control Manager UI automatically generates the setup script, based on the selected table/view, functions, and procedures required for each version/patch.

Manifest File

The Snowflake native app Framework requires that every application package contains a manifest file. This file defines properties required by the application package, including the location of the setup script and version definitions.

- The manifest file has the following requirements:
 - The name of the manifest file must be manifest.yml.
 - The manifest file must be uploaded to a named stage so that it is accessible when creating an application package or Snowflake native app.
 - The manifest file must exist at the root of the directory on the named stage.
 - For more information, visit <https://docs.snowflake.com/en/developer-guide/native-apps/creating-manifest>.

The ACF includes a manifest template that is used to construct each app version/patch's setup script. The ACF's App Control Manager UI automatically generates the manifest file.

Readme

A readme file is included when the consumer installs the corresponding version. Each readme is slightly different due to setup steps required for each app version.

Sidecar Loader

Sidecar is a utility that allows the consumer to execute pre-set commands in their account that cannot be executed by the application during installation. The Sidecar loads commands into a table to be executed by the consumer, via the SidecarRunner stored procedure. The commands are visible and can be evaluated prior to execution.

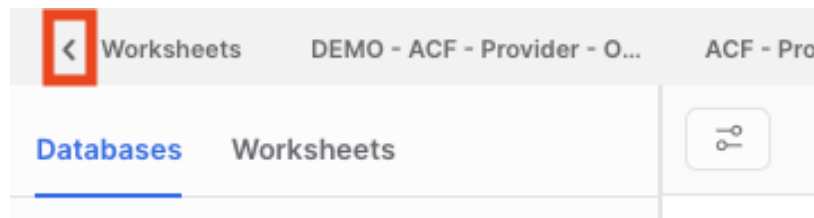
App Control Manager

The Application Control Framework includes the App Control Manager, a Streamlit UI available in the provider account. The App Control Manager allows the provider to easily build and manage an app built on the ACF, manage consumers, and remove the ACF if/when desired.

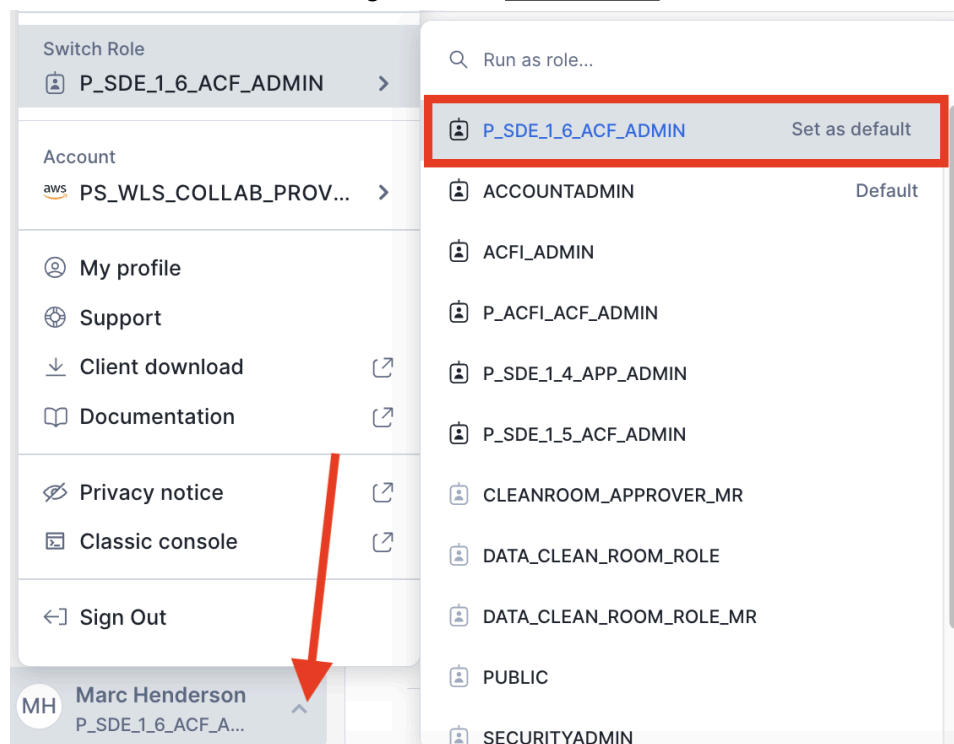
The App Control Manager can be accessed via the following steps:

Step 1: Log into **Snowsight**.

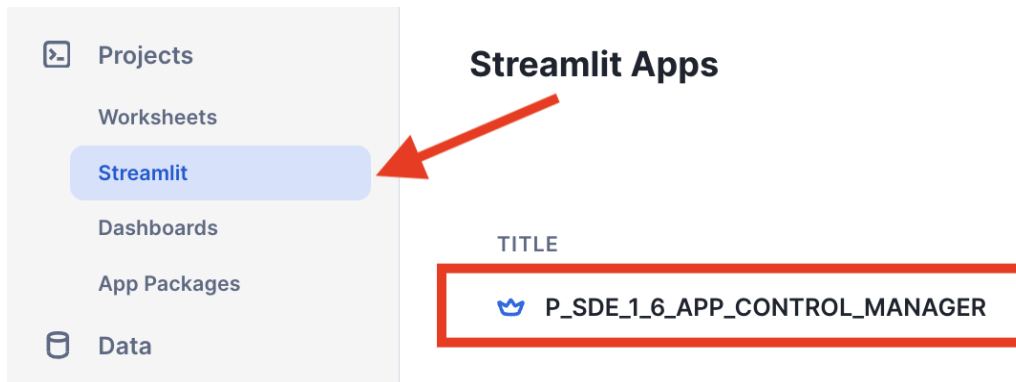
Step 2: Once logged in, if not at the Snowsight home screen, click the **Back** button, in the top left area of the UI, to open the left navigation menu.



Step 3: Switch to the **P_<APP_CODE>_ACF_ADMIN** role, by clicking the drop-down in the bottom left area of the UI, then hovering over the Switch Role menu item.



Step 4: Click **Projects** >> **Streamlit**, then **P_<APP_CODE>_APP_CONTROL_MANAGER**.



App Control Manager: SDE_1_6

1.22.0

Use the following options below to **Manage** or **Remove** an App built on the **App Control Framework**, along with the ability to **Manage App Consumers**.

Manage App



Click the button below to manage an existing Native app built on the App Control Framework.

Manage App

Consumers



Click the button below to manage existing consumers of a Native App.

Manage Consumers

Remove ACF



Click the button below to remove all objects created by the Application Control Framework.

Remove ACF

⚠ NOTE: The ACCOUNTADMIN role must be granted to the user in order to remove the ACF.

(App Control Manager Home)

Build and Deploy Demo Native App

Step 1: Build the Demo Native App

The following sections outline how to use the App Control Manager to build the demo native app, included with the ACF.

This demo app is a simple data enrichment app that enriches a consumer's dataset with attributes from the provider's dataset, when the records have the same email address.

Part 1: Create the Application Package

The native app's application package consists of views of the native app's source data, along with views of the required Application Control Framework tables used to enforce the rules defined in the ACF and collect logs/metrics. In this demo, an application package will be created:

Step 1: In the App Control Manager, click **Manage App >> App Package >> Create**.

Create Application Package



Click the button below to create a new Application Package

Create

Manage Application Versions



Click the button below to create, patch, or drop a version

Versions

Promote Application Package



Click the button below to promote an application package to the PROD environment.

Promote

Drop Application Package



Click the button below to drop an application package.

Drop

NOTE: the Marketplace listing created from the application package must be first unpublished and deleted.

Step 2: Enter **SDE_DEMO** in the Application Package Name text field. This name will be prefixed by: **P_<APP_CODE>_APP_PKG_**.

Step 3: Select the **P_<APP_CODE>_SOURCE_DB_DEV** database from the Select Database drop-down.

Step 4: Select the **DATA** schema from the Select Schema drop-down

Step 5: Select the **P_<APP_CODE>_SOURCE_DB_DEV.DATA.CUSTOMER** table from the Select Table(s)/View(s) drop-down.

Step 6: Click **Create**.

Application Package Name *

SDE_DEMO 2

Please select source table(s)/view(s).

⚠ NOTE: The source table(s)/view(s) should be granted to the app admin role, prior to creating the application package.

Select Database:

P_SDE_1_6_SOURCE_DB_DEV 3

Select Schema:

DATA 4

Select Table/Views(s):

P_SDE_1_6_SO... x 5

☐ Diagnostic Mode:

Selected Source Data:

P_SDE_1_6_SOURCE_DB_DEV.DATA.CUSTOMER

Remove

Home

Back

6
Create

Part 2: Create/Patch a Version for the Application Package

Once the application package is created, the application functions/procedures are tied to the application package by creating a version and a patch. In this demo, the App Control Manager will be used to create a new version and patch:

Step 1: In the App Control Manager, click **Manage App >> App Package >> Versions**.

Create Application Package



Click the button below to create a new Application Package

Create

Manage Application Versions



Click the button below to create, patch, or drop a version

Versions

Promote Application Package



Click the button below to promote an application package to the PROD environment.

Promote

Drop Application Package



Click the button below to drop an application package.

Drop

NOTE: the Marketplace listing created from the application package must be first unpublished and deleted.

Step 2: Select **P_SDE_APP_PKG_SDE_DEMO** from the Select Application Package drop-down.

Step 3: Select **CREATE** from the Create, Patch, or Drop Version drop-down.

Step 4: Enter a **v1_0_0** in the Version Name field.

Select Application Package

P_SDE_1_6_APP_PKG_SDE_DEMO

2

Create, Patch, or Drop Version

CREATE

3

Version Name

v1_0_0

4

It is recommended to use major/minor versioning i.e.: [v1_0_0](#). This will also be the name of the stage that stores the setup scripts for this app version

Step 5: For this demo, select **FREE** from the Select App Mode drop-down

Step 6: Select **Y** or **N** from the Enforce Limits drop-down. By default, limits should be enforced, but this can be set to N if limit enforcement should be turned off (i.e. during testing).

Select App Mode

FREE

5

Enforce Limits

Y

6

Step 7: Select DEV from the Environment drop-downs to pull the **Streamlit** and **template** files from the dev environment.

Please select environment containing the Streamlit artifacts.

⚠ NOTE: If the Streamlit artifacts have not changed since the latest release, choose **PROD**.

Environment:

DEV ▼

Please select environment containing the template files.

⚠ NOTE: If the template files have not changed since the latest release, choose **PROD**.

Environment:

DEV ▼

Step 8: Select the **P_<APP_CODE>_SOURCE_DB_DEV** database from the Select Database drop-down.

Step 9: Select the **FUNCS_APP** schema from the Select Schema drop-down.

Step 10: Select the **P_<APP_CODE>_SOURCE_DB_DEV.FUNCS_APP.JS_FACTORIAL(FLOAT)** function from the Select Function(s) drop-down.

Step 11: Select the **Accessible** checkbox next to the function under Selected Functions to make the function accessible to the consumer.

Please select source function(s).

△ NOTE 1: The source functions should be created by/granted to the app admin role, prior to creating the application package.

△ NOTE 2: If **NONE** of the source functions have changed since the latest release, choose **P_SDE_1_6_SOURCE_DB_PROD**, otherwise choose **P_SDE_1_6_SOURCE_DB_DEV**

Select Database:

P_SDE_1_6_SOURCE_DB_DEV 8 ▾

Select Schema:

FUNCS_APP 9 ▾

Select Function(s):

P_SDE_1_6_S... x 10 x ▾

☐ Diagnostic Mode:

Selected Functions:

<div><input checked="" type="checkbox"/> Accessible</div>	P_SDE_1_6_SOURCE_DB_DEV.FUNCS_APP.JS_FACTORIAL(FLOAT)	Remove
---	---	--------

Step 12: Select the **P_<APP_CODE>_SOURCE_DB_DEV** database from the Select Database drop-down.

Step 13: Select the **PROCS_APP** schema from the Select Schema drop-down.

Step 14: Select the **P_<APP_CODE>_SOURCE_DB_DEV.PROCS_APP.ENRICH(VARCHAR,VARCHAR,VARCHAR,VARCHAR)** procedure from the Select Procedure(s) drop-down.

Step 15: Select the **Input Table** checkbox next to the procedure under Selected Procedures.
Please select source procedure(s).

△ NOTE 1: The source procedures should be created by/granted to the app admin role, prior to creating the application package.

△ NOTE 2: If **NONE** of the source procedures have changed since the latest release, choose **P_SDE_1_6_SOURCE_DB_PROD**, otherwise choose **P_SDE_1_6_SOURCE_DB_DEV**

Select Database:

P_SDE_1_6_SOURCE_DB_DEV

12

▼

Select Schema:

PROCS_APP

13

▼

Select Procedures(s):

P_SDE_1_6_S... x

14

x ▼

☐ Diagnostic Mode:

Selected Procedures:

<input checked="" type="checkbox"/>	Input Table	P_SDE_1_6_SOURCE_DB_DEV.PROCS_APP.ENRICH(VARCHAR, VARCHAR, VARCHAR, VARCHAR)	Remove
-------------------------------------	-------------	--	--------

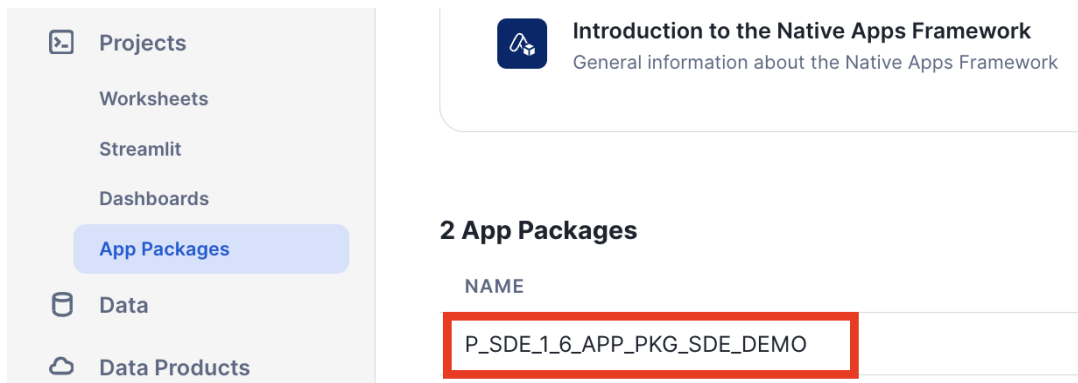
15

Step 16: Click the **CREATE** button.

Part 3: Release Patch

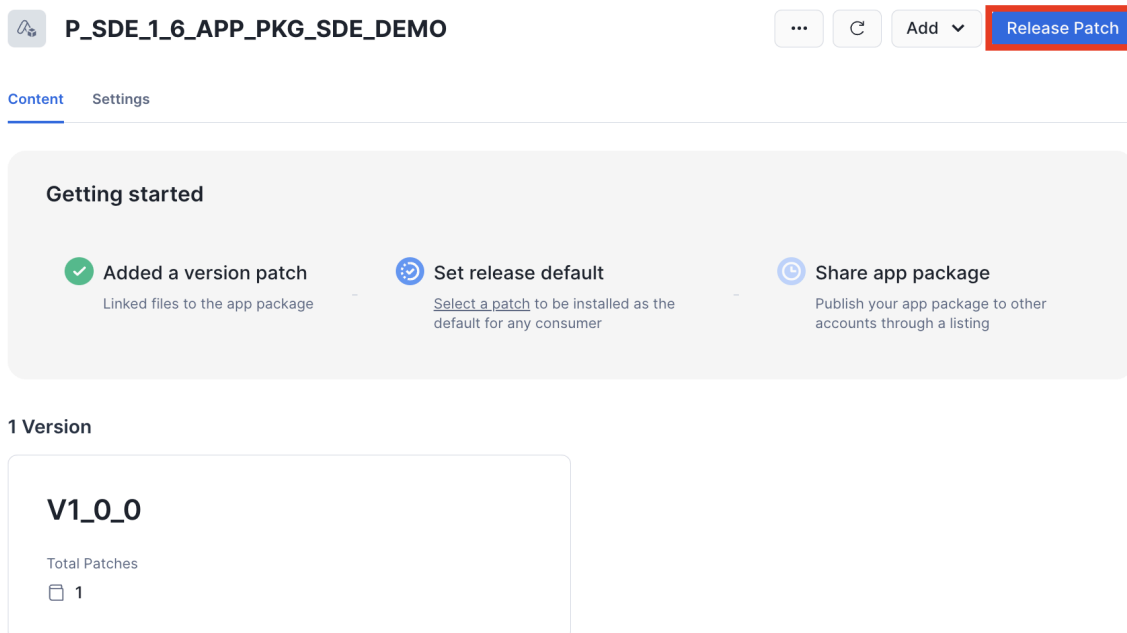
To grant consumers access to the latest version/patch, the provider must create a release for the application package. The following steps detail how to release an application package's version/patch:

Step 1: In Snowsight, click **Projects >> App Packages**. Select the **P_SDE_APP_PKG_SDE_DEMO** application package created in [Part 2](#).



The screenshot shows the Snowflake Snowsight interface. On the left, a sidebar contains navigation items: Projects, Worksheets, Streamlit, Dashboards, App Packages (highlighted), Data, and Data Products. The main content area displays '2 App Packages'. Below this, there is a table with a header 'NAME' and one row containing 'P_SDE_1_6_APP_PKG_SDE_DEMO', which is highlighted with a red rectangular box.

Step 2: Click **Release Patch**.



The screenshot shows the Snowflake Snowsight interface for the 'P_SDE_1_6_APP_PKG_SDE_DEMO' application package. At the top, the package name is displayed. To the right of the name are three buttons: a menu button (three dots), a refresh button (circular arrow), and an 'Add' button with a dropdown arrow. The 'Release Patch' button, which is highlighted with a red rectangular box, is located to the right of the 'Add' button. Below the package name, there are two tabs: 'Content' (active) and 'Settings'. The 'Content' tab shows a 'Getting started' section with three steps: 'Added a version patch' (with a green checkmark), 'Set release default' (with a blue gear icon), and 'Share app package' (with a blue share icon). Below this, there is a section for '1 Version' showing 'V1_0_0' with 'Total Patches: 1'.

Step 3: Select **V1_0_0 patch 0** from the Patch to release drop-down.

Step 4: Choose **Set this patch to default**.

Step 5: Click **Save**.

Release Patch

P_SDE_1_6_ACF_ADMIN

Patch to release

V1_0_0 patch 0 ("v1_0_0") 3

Release options

☒ **Set this patch to default**
This patch will be released to all new and existing consumers unless they are explicitly targeted with another patch 4

☐ **Target this patch to specific accounts**
Release this patch to targeted accounts only.

5

Cancel

Save

Step 2: Create the Native App Listing

Once the application package has a released version and patch, the native app is ready to be privately listed. For instructions on how to create a Private Listing for the native app and add a demo consumer account, visit the **Create a Listing for Your Application** section:

<https://docs.snowflake.com/en/developer-guide/native-apps/tutorials/getting-started-tutorial#publish-and-install-your-application>.

Step 3: Install the Native App in the Consumer Account

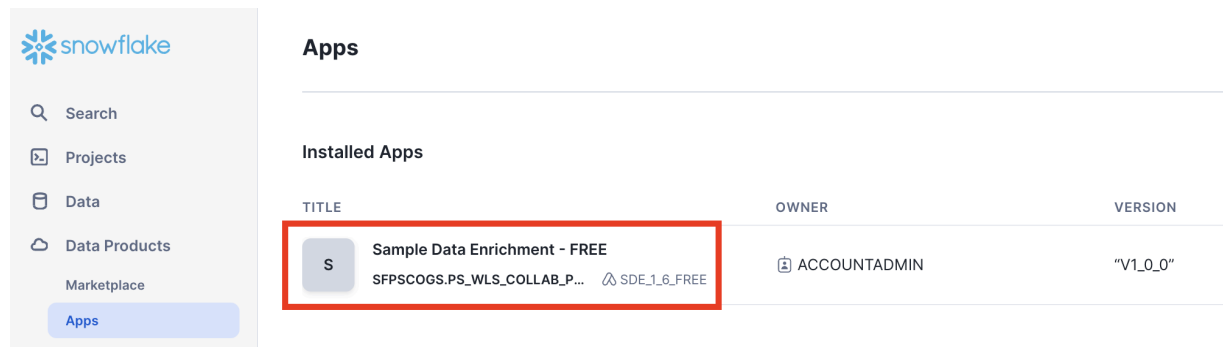
In the demo consumer account added to the listing created in Step 2, install the native app available from the private listing. For instructions on how to install a native app, visit the **Install the Application** section

<https://docs.snowflake.com/en/developer-guide/native-apps/tutorials/getting-started-tutorial#id6>.

Step 4: Run the Scripts from the Native App's Readme

With each installation, the consumer is barred from using the native app until the consumer has executed a series of commands, including sharing events with the provider. These commands are located in the native app's Readme.

Step 1: In Snowsight, click **Data Products** >> **Apps**, under the Installed Apps section, click the native app installed in the previous section.



TITLE	OWNER	VERSION
Sample Data Enrichment - FREE SFPSCOGS.PS_WLS_COLLAB_P... SDE_1_6_FREE	ACCOUNTADMIN	"V1_0_0"

Step 2: Once loaded, the Streamlit UI will display the following error message:

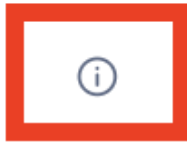


Sample Data Enrichment

1.22.0

Please complete installation by running the setup script found in the ⓘ icon in the top right of the page.

Step 3: Click the ⓘ icon in the top right of the page.



Manage Access

• ADHOC_WH

Step 4: Make note of the NOTES section of the Readme. Hover over and click the **Open in Worksheets** button.

Sample Data Enrichment App - FREE


Prerequisites

Prior to using the this app, the following one-time setup steps below must be executed:

NOTES:

- The ACCOUNTADMIN role is required for this step.
- Replace `<MY_WAREHOUSE>` with the desired warehouse.
- For this step, an XSMALL warehouse can be used.
- Replace all `<APP_NAME>` references with the name of the native app, as installed in the consumer account.
 - The App Name can be found by executing (as ACCOUNTADMIN or the role that installed the app):
`SHOW APPLICATIONS;` (reference the **name** column)

Open in Worksheets



```
USE ROLE ACCOUNTADMIN;
USE WAREHOUSE <MY_WAREHOUSE>;

CREATE DATABASE IF NOT EXISTS SIDECAR;
CREATE SCHEMA IF NOT EXISTS SIDECAR.RUNNER;
CREATE OR REPLACE PROCEDURE SIDECAR.RUNNER.SidecarRunner(app_name string)
RETURNS STRING
LANGUAGE PYTHON
RUNTIME_VERSION = '3.8'
PACKAGES = ('snowflake-snowpark-python')
HANDLER = 'run_sidecar_sql'
```

App Usage

Please refer to the Consumer Guide for details on how to use the app.

Step 5: In the new worksheet, replace the macros specified in the Readme with the appropriate values. Once the macros are replaced, **run all commands**.

Step 6: Open the native app. The Streamlit UI will display the warning message below. It will take a few minutes for the native app to be enabled.



Sample Data Enrichment

1.22.0

Please wait a moment while we finish onboarding your account. This automated process may take a few minutes. If you continue to have issues, please contact us immediately.

[Contact Us](#)

Step 5: App Usage

Once the native app has been enabled, the consumer can now use the app to enrich data. Sample data was created during executing the Readme commands. The following steps detail how to use the app to enrich data:

Step 1: Open the app. Once the app is enabled, the warning message is removed.

Step 2: Click **Select Table**. Since this is the first time using the app, it needs to be granted privileges to the sample data table.

Sample Data Enrichment

1.22.0

App Version: FREE

Please Select a Table or View to Enrich.

Please Select a Table or View

Select Table

Select View

Step 3: The app's security window will appear. Click **+ Select Data**.



This app would like privileges to access tables in your account

Grant **SELECT** on Enrichment Table

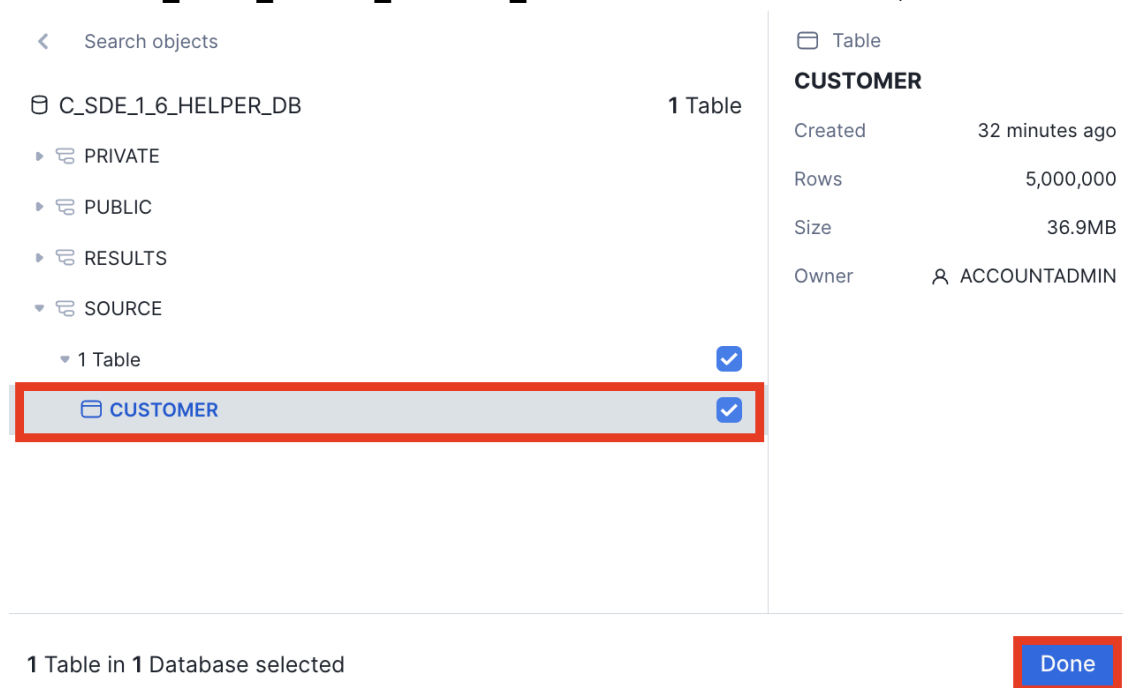
Select data table

+ Select Data

Cancel

Save

Step 4: Select **C_<APP_CODE>_HELPER_DB.SOURCE.CUSTOMERS**, then click **Done**.



Search objects

C_SDE_1_6_HELPER_DB 1 Table

- PRIVATE
- PUBLIC
- RESULTS
- SOURCE
 - 1 Table
 - CUSTOMER

Table

CUSTOMER

Created 32 minutes ago

Rows 5,000,000

Size 36.9MB

Owner ACCOUNTADMIN

1 Table in 1 Database selected

Done

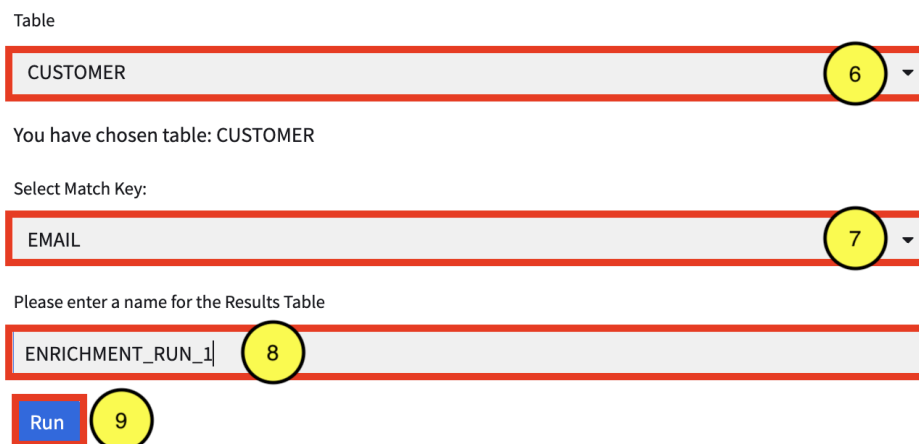
Step 5: In the security window, click **Save**.

Step 6: Verify that the **CUSTOMER** table is selected from the Table drop-down.

Step 7: Select **EMAIL** from the Match Key drop-down.

Step 8: Enter a **name** for the results table in the Please enter a name for the Results Table field.

Step 9: Click **Run**.



Table

CUSTOMER 6

You have chosen table: CUSTOMER

Select Match Key:

EMAIL 7

Please enter a name for the Results Table

ENRICHMENT_RUN_1 8

Run 9

Step 10: Once complete, the following message will briefly appear. In addition a new entry will be added to the Run History section.

Please Select a Table or View to Enrich.

Select Table

Select View

Table

CUSTOMER

You have chosen table: CUSTOMER

Select Match Key:

EMAIL

Please enter a name for the Results Table

ENRICHMENT_RUN_1

Run



Updating...

Enrichment complete 🎉. Results are in table: SDE_1_6_FREE.RESULTS_APP.ENRICHMENT_RUN_1

Run History

	TIMESTAMP	REQUEST_ID	REQUEST_TYPE	INPUT_TABLE	OUTPUT
0	2024-03-08 02:52:48	f39703a1-f149-4494-9dce-5bb53caa200e	enrich	SDE_1_6_FREE.UTIL_APP.CONSUMER_VIEW	SDE_1_

Step 11: If desired, the output table can be inspected. In a worksheet, execute the following command: `SELECT * FROM <APP_NAME>.RESULTS_APP.<TABLE_NAME> WHERE email IS NOT NULL LIMIT 100;`

NOTES:

- <APP_NAME> = the name of the app
- <TABLE_NAME> = the name of the results table entered in Step 8.

Additional Test Scenarios

Reach the Five Requests Limit Then Increase the Limit

By default, the ACF sets limits of the FREE version of the app to five requests. Once the limit is reached, the app is no longer usable, until the limit is increased by the provider in the App Control Manager.

Attempt to Exceed Limit

In the app, make five requests to reach the limit, then attempt a sixth, which will fail.

Please Select a Table or View to Enrich.

Table

CUSTOMER

You have chosen table: CUSTOMER

Select Match Key:

EMAIL

Please enter a name for the Results Table

ENRICHMENT_RUN_6

You've reached your FREE limit of 5 requests(s). If you wish to run additional queries, please contact us to discuss our Paid or custom Enterprise solutions.

Increase the Limit

Once the limit is reached, increase the consumer's request limit in the App Control Manager.

Step 1: In the App Control Manager, click **Manage Consumers** >> **Manage**.

Step 2: Select the **Consumer** from the Select Consumer drop-down. The consumer name will have a **FREE_** prefix, followed by the consumer's Snowflake account name.

Step 3: Select the **limit** control from the Select Consumer Controls to Update drop-down

Step 4: In the **New Value** field, enter **10**, then press Enter.

Step 5: Click **Update**.

Select Consumer:

FREE_PS_WLS_COLLAB_CONSUMER

2

Select Consumer Controls to Update:

limit x

3

☐ Diagnostic Mode:

Selected Controls:

Control:

limit

New Value:

10

Press Enter to apply

☐ Clear Value? ?

☐ Apply to Other Consumers

Home

Back

Update

5

Make a New Request

In the native app in the consumer's account, attempt to make a new request. Verify that the request is allowed and is successful.

Consumer Usage of the App is Blocked, when Disabled

The App Control Manager allows the provider to control a consumer's access.

Disable the Consumer

In the App Control Manager, disable the consumer.

Step 1: In the App Control Manager, click **Manage Consumers >> Manage**.

Step 2: Select the **Consumer** from the Select Consumer drop-down. The consumer name will have a **FREE_** prefix, followed by the consumer's Snowflake account name.

Step 3: Select the **enabled** control from the Select Consumer Controls to Update drop-down

Step 4: In the **New Value** field, enter **N**, then press Enter.

Step 5: Click **Update**.

Select Consumer:

FREE_PS_WLS_COLLAB_CONSUMER

2

Select Consumer Controls to Update:

enabled x

3

☐ Diagnostic
Mode:

Selected Controls:

Control:

enabled

New Value:

N

4

☐ Clear Value? ?

☐ Apply to
Other
Consumer
s

Home

Back

Update

5

Attempt to Use the Native App

In the consumer account, relaunch the app and note that the consumer's access has been revoked.

Sample Data Enrichment

1.22.0

Your app usage has been disabled. Please contact us if you would like to re-enable your app usage.

Contact Us

Re-enable the Consumer

Repeat the steps in the [Disable the Consumer](#) section above, but set the enabled value back to Y. Then relaunch the app in the consumer's account to verify that usage has been restored.