Make a copy of the document if any changes are required for customer deliverables.

**Please DO NOT edit this document.**

Copy Instructions:

1. Click **File** -> **Make a Copy.**
2. Replace ‘Copy of ‘ from the **Name** text field with ‘<CUSTOMER> - ‘.
3. Replace ‘<SOLUTION\_NAME> ’ with the Provider’s solution.
4. Select the desired folder to save a copy.
5. Select the **Copy comments and suggestions** box. This will also select the **Share it with the same people** box above it.

# 

# 

<CUSTOMER>

<SOLUTION\_NAME>

Consumer Objects

Version 1

<AUTHOR>

<DATE>

# Revision Summary

| **Date** | **Revision History** | **Comments** |
| --- | --- | --- |
| <REV\_DATE> | 1.0 | Initial Version |

# Table of Contents

[**Prerequisites 6**](#_kumxb0czmehl)

[**Provider vs. Consumer 6**](#_268k2k4i7fjx)

[**Disclaimer 6**](#_b9kgfgw021lx)

[**Objects Created 7**](#_26b945ichfre)

[Consumer Account 7](#_79yi9fl2w7en)

[Application: (APPLICATION OBJECT) 7](#_n83ukvocaxdw)

[Application Role: APP\_ROLE 7](#_xz0tpzfek6cp)

[Schema (Non-versioned): APP 7](#_6dnxaqlczt75)

[Table: APP\_KEY 7](#_bjdiekeqptxa)

[Table: APP\_MODE 7](#_3ufprg2gl9i8)

[Table: LIMIT\_TRACKER 8](#_b505eum11sf0)

[Table: RUN\_TRACKER 9](#_izytkwl7sdnu)

[Task: COUNTER\_RESET\_TASK 9](#_t1w6c9e7tgy)

[Schema (Versioned): UTIL\_APP 9](#_64qen2kwnqod)

[Table: ALL\_PROCS 9](#_8e0axq6qjqvp)

[View (Secure): ALLOWED\_PROCS\_V 10](#_82ai0tj8q5jx)

[Table: REQUEST\_ID\_TEMP 10](#_gxfeizmmx5xw)

[View (Secure): METADATA\_C\_V 10](#_6abrjxr8iufv)

[View: REQUEST\_SUMMARY\_C\_V 11](#_sgsmgc7x1arj)

[Stored Procedure: CONFIGURE\_TRACKER 12](#_ckp3kp408m3u)

[Stored Procedure: REGISTER\_SINGLE\_CALLBACK 12](#_lu5bxhy0ors)

[Stored Procedure: APP\_LOGGER 12](#_7l8pg8qvcg33)

[Stored Procedure: CUSTOM\_EVENT\_BILLING 13](#_x1odkqlqnesh)

[Stored Procedure: LOG\_FORM 13](#_o797a12akfww)

[Stored Procedure: LOAD\_INSTALL\_SQL 13](#_m2f7cvnmof83)

[Schema (Non-versioned): SETUP 13](#_bxckr9rjyxiz)

[Table: SQL 14](#_l55rz7qmyxki)

[Schema (Non-versioned): RESULTS\_APP 14](#_y7ub9mi4ggq2)

[Table: (RESULTS TABLE) 14](#_b4h0rwq0wbsz)

[Schema (Versioned): FUNCS\_APP 14](#_owdyis1q8xxs)

[Schema (Versioned): PROCS\_APP 14](#_jkkxo2qvjqab)

[Stored Procedure: LOG\_SHARE\_INSERT 14](#_nfoti9dsmwjv)

[Stored Procedure: REQUEST 15](#_2r29ubiibvu2)

[Consumer Objects 16](#_sa4wpe8myi5z)

[Database: SIDECAR 16](#_lsa1s3vs01ih)

[Schema: Runner 16](#_elzxmks5fog7)

[Stored Procedure: SidecarRunner 16](#_muaeyro58bj4)

[Role: C\_<APP\_CODE>\_APP\_ADMIN 16](#_lmk6e4ucu4xw)

[Warehouse: C\_<APP\_CODE>\_APP\_WH 16](#_5x31q89z7n5o)

[Database: C\_<APP\_CODE>\_HELPER\_DB 17](#_u26xj92ktte0)

[Schema: SOURCE 17](#_hpls9wqvca9x)

[Table/View: (SOURCE TABLES/VIEWS) 17](#_f3hkjulu0d7b)

[Schema: RESULTS 17](#_y52nj38frzt6)

[Table: (RESULT TABLES) 17](#_cdchqpvqmdyo)

[Schema: PRIVATE 17](#_s8h35xdp14l1)

[Stored Procedure: DETECT\_EVENT\_TABLE 17](#_qj5b5vu26ap3)

[Database: EVENTS 18](#_8n3ljl91cody)

[Schema: EVENTS 18](#_9td1sv8tu9lc)

[Table: EVENTS 18](#_eax9bsc3fyux)

# Prerequisites

* Both the provider and the consumer Snowflake accounts must be on the same cloud provider and region.
* The provider must accept the Snowflake Marketplace Provider Terms of Service.
* The consumer must accept the Snowflake Marketplace Consumer Terms of Service.

# Provider vs. Consumer

The Native Apps Framework term “provider” refers to <CUSTOMER>, as the app’s owner. The Native Apps Framework term “consumer” refers to <CUSTOMER> clients that install the app.

# Disclaimer

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

# Objects Created

## Consumer Account

The following objects are created in the consumer’s account when the app is installed.

### Application: (APPLICATION OBJECT)

**Description**

The app object that contains the native app objects installed in the consumer’s account.

#### Application Role: APP\_ROLE

The app role is granted privileges to objects that should be accessible to the consumer using the app.

There are objects that the app creates that are used by the app, but not accessible to the consumer. The app role will not be granted privileges to these objects.

#### Schema (Non-versioned): APP

**Description**

The non-versioned schema that contains the app\_key table and the run\_id sequence. This schema is non-versioned to avoid updating these objects in the event of an app upgrade.

##### Table: APP\_KEY

**Description**

The table created during installation, that stores the app key for the consumer’s installation. The app key is used to confirm requests are coming from a valid installation.

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| APP\_KEY | VARCHAR | The consumer’s app key for the installation | N |

##### Table: APP\_MODE

**Description**

The table that stores fields related to which mode the app is in. When in FREE mode, this table is used to regulate the terms of usage (i.e. how many records can be processed).

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| KEY | VARCHAR | The metadata key name (i.e. app\_mode) | N |
| VALUE | TIMESTAMP\_LTZ(9) | The metadata value (i.e. ‘Y’) | N |

The following keys are inserted into this table:

* **app\_mode** - The app’s current mode (FREE, PAID, or ENTERPRISE)

##### Table: LIMIT\_TRACKER

**Description**

The table that stores the limits to be enforced for non-free versions of the app.

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| KEY | VARCHAR | The metadata key name (i.e. app\_mode) | N |
| VALUE | VARCHAR | The metadata value (i.e. ‘Y’) | N |

The following keys are inserted into this table:

* **total\_requests** - The number of requests made throughout the life of consumer usage of the native app.
* **requests\_processed\_this\_interval** - The number of requests made within the defined LIMIT\_INTERVAL (this is defined in the METADATA table)
* **input\_records** - The total number of input records submitted
* **input\_records\_this\_interval** - The number of input records submitted this interval (i.e. 30 days)
* **total\_records\_processed** - The number of records processed throughout the life of consumer usage of the native app.
* **records\_processed\_this\_interval** - The number of records processed within the defined LIMIT\_INTERVAL (this is defined in the METADATA table).
* **total\_matches** - The number of matches throughout the life of consumer usage of the native app (if applicable).
* **matches\_this\_interval** - The number of matches within the defined LIMIT\_INTERVAL (this is defined in the METADATA table).
* **last\_request\_timestamp** - The timestamp of the last request

**NOTE**: This is hidden from the consumer.

##### Table: RUN\_TRACKER

**Description**

The table that stores historical runs.

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| TIMESTAMP | TIMESTAMP\_NTZ(9) | The timestamp of the request | N |
| REQUEST\_ID | VARCHAR | The run’s request ID | N |
| REQUEST\_TYPE | VARCHAR | The type of request | N |
| INPUT\_TABLE | VARCHAR | The input table | Y |
| OUTPUT\_TABLE | VARCHAR | The output table | Y |

**NOTE**: This is hidden from the consumer.

##### Task: COUNTER\_RESET\_TASK

**Description**

Task that resets the consumer’s counter values, after a specified time has elapsed. This is created when the consumer executes the CONFIGURE\_TRACKER stored procedure. This task is only created for PAID and ENTERPRISE versions of the app.

* **Warehouse**: N/A (serverless)
* **Schedule**: ‘15 minute’
* **Action**: Updates the counters in the LIMIT\_TRACKER table. The task also logs a message to reset the counters. Once received in the ACF account, the consumer’s counts are reset in the METADATA table.

#### Schema (Versioned): UTIL\_APP

**Description**

Schema containing utility-type objects, such as the consumer’s metadata view, metric views, etc.

##### Table: ALL\_PROCS

**Description**

The table that contains a list of all of the app logic stored procedures created by the app. These stored procedures will be hidden from the consumer, but accessible via the app’s REQUEST stored procedure (described later in this document).

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| PROC\_NAME | VARCHAR | The name of the app logic stored procedure | N |
| PROC\_SIGNATURE | VARCHAR | The stored procedure’s signature | N |
| REQUIRE\_INPUT\_TABLE | VARCHAR | Flag indicating whether the stored procedure requires an input table | N |

**NOTE**: This is hidden from the consumer.

##### View (Secure): ALLOWED\_PROCS\_V

**Description**

The secure view created from the ALL\_PROCS table that reveals the app logic stored procedure(s) the consumer has access to. The consumer uses this to view the parameters passed to the accessible procedures. The consumer’s access to the app logic stored procedures is controlled by the METADATA table.

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| PROC\_NAME | VARCHAR | The name of the app logic stored procedure | N |
| PROC\_SIGNATURE | VARCHAR | The stored procedure’s signature | N |
| REQUIRE\_INPUT\_TABLE | VARCHAR | Flag indicating whether the stored procedure requires an input table | N |

##### Table: REQUEST\_ID\_TEMP

**Description**

A table created when calling the REQUEST stored procedure to store the request id.

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| REQUEST\_ID | VARCHAR | The request ID generated when calling the app’s REQUEST stored procedure | N |

##### View (Secure): METADATA\_C\_V

**Description**

A secure view created from the METADATA\_V view in the app package, only containing the consumer’s metadata.

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| ACCOUNT\_LOCATOR | VARCHAR | The consumer’s Snowflake Account Locator | N |
| CONSUMER\_NAME | VARCHAR | The consumer’s company name | N |
| KEY | VARCHAR | The metadata key name (i.e. ‘enabled’) | N |
| VALUE | TIMESTAMP\_LTZ(9) | The metadata value (i.e. ‘Y’) | N |

##### View: REQUEST\_SUMMARY\_C\_V

**Description**

View presenting the consumer’s metrics entries from the METRICS table, in tabular format.

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| ACCOUNT | VARCHAR | The consumer’s Snowflake Account Locator | N |
| CONSUMER\_NAME | VARCHAR | The consumer’s company name | N |
| ENTRY\_TYPE | VARCHAR | The type of event (i.e. log or metric) | N |
| REQUEST\_ID | VARCHAR | Request ID | N |
| PROC\_NAME | VARCHAR | The allowed procedure called | N |
| PROC\_PARAMETERS | VARCHAR | The allowed procedure’s parameters | Y |
| INPUT\_TABLE\_NAME | VARCHAR | Input table name | N |
| INPUT\_RECORD\_COUNT | NUMBER(38,0) | Input table record count | Y |
| RESULTS\_TABLE\_NAME | VARCHAR | Results table name | Y |
| RESULTS\_RECORD\_COUNT | NUMBER(38,0) | Results table record count | N |
| RESULTS\_RECORD\_COUNT\_DISTINCT | NUMBER(38,0) | Results table record count(distinct) | N |
| STATUS | VARCHAR | Request status | N |
| COMMENTS | VARCHAR | Comments | Y |
| SUBMITTED\_TS | TIMESTAMP\_LTZ(9) | Submitted timestamp | N |
| COMPLETED\_TS | TIMESTAMP\_LTZ(9) | Completed timestamp | Y |

##### Stored Procedure: CONFIGURE\_TRACKER

**Description**:

This procedure creates a serverless task that reset the consumer’s interval counts according to the interval set. This is only available in PAID and ENTERPRISE versions of the app.

**Parameters**:

* **N/A**

##### Stored Procedure: REGISTER\_SINGLE\_CALLBACK

**Description**:

This procedure registers an object to be used by the native app (i.e. table or view).

**Parameters**:

* **ref\_name** (STRING) - The reference name
* **operation** (STRING) - The operation to be performed.
* **ref\_or\_alias** (STRING) - Reference or alias

##### Stored Procedure: APP\_LOGGER

**Description**:

This procedure adds event messages to the consumer’s events table.

**Parameters**:

* **account\_locator** (VARCHAR) - The consumer’s Snowflake Account Locator
* **consumer\_name** (VARCHAR) - The consumer’s name
* **app\_key** (VARCHAR) - The consumer’s app key for the installation
* **app\_mode** (VARCHAR) - The app’s mode (i.e. FREE, PAID, or ENTERPRISE)
* **entry\_type** (VARCHAR) - The type of entry (i.e. LOG or METRIC)
* **event\_type** (VARCHAR) - The type of event being logged (i.e. INSTALL or REQUEST)
* **event\_attributes** (VARCHAR) - Any applicable attributes associated with the event type (can be NULL). **NOTE**: if the message is a string, it should be enclosed in double quotes (i.e. ‘“this is a test event”’).
* **timestamp** (TIMESTAMP\_NTZ) - The UTC timestamp for the event
* **status** (VARCHAR) - The status of the event (i.e. PROCESSING, COMPLETE, or ERROR)
* **message** (VARCHAR) - The message to log. **NOTE**: if the message is a string, it should be enclosed in double quotes (i.e. ‘“this is my message”’).

**NOTE**: This is hidden from the consumer.

##### Stored Procedure: CUSTOM\_EVENT\_BILLING

**Description**:

This procedure registers a custom billing event that charges the consumer a specified amount for an event (i.e. $0.05 for each record processed). Custom billing events are useful in the ENTERPRISE app version, where consumers may be charged different rates for billing events.

**Parameters**: See <https://docs.snowflake.com/en/sql-reference/functions/system_create_billing_event>

**NOTE**: This is hidden from the consumer.

##### Stored Procedure: LOG\_FORM

**Description**:

This procedure adds contact information, from the app’s Streamlit UI, to the events table. This is useful for collecting information from consumers interested in upgrading their app.

**Parameters**:

* **first\_name** (STRING) - The consumer’s first name
* **last\_name** (STRING) - The consumer’s last name
* **title** (STRING) - The consumer’s title
* **business\_email** (STRING) - The consumer’s business email address
* **industry** (STRING) - The consumer’s professional industry
* **contact\_reason** (STRING) - The reason the consumer is contacting the provider (i.e. app upgrade)
* **contact\_reason\_text** (STRING) - More details the consumer provides regarding the contact reason.

**NOTE**: This is hidden from the consumer.

##### Stored Procedure: LOAD\_INSTALL\_SQL

**Description**:

This procedure loads the consumer setup commands into the SQL.SETUP table.

**Parameters**:

* **app\_name** (VARCHAR) - The name of the app, as installed in the consumer’s account.
* **app\_user** (VARCHAR) - The current user

#### Schema (Non-versioned): SETUP

**Description**

The schema that stores the table that contains SQL commands to be executed by the SidecarRunner stored procedure.

##### Table: SQL

**Description**

A table that contains SQL commands to be executed by the SidecarRunner stored procedure.

**Definition**

| **Column** | **Data Type** | **Description** | **Null?** |
| --- | --- | --- | --- |
| SQL | VARCHAR | The SQL commands to be executed by the SidecarRunner stored procedure. | N |

##### 

#### Schema (Non-versioned): RESULTS\_APP

**Description**

The non-versioned schema that stores the results table from calling the allowed stored procedure(s) via the app’s REQUEST stored procedure. This schema is non-versioned to avoid losing results tabled in this schema, in the event of an app upgrade.

##### Table: (RESULTS TABLE)

**Description**

The resulting table from calling the allowed stored procedure(s) via the app’s REQUEST stored procedure.

#### Schema (Versioned): FUNCS\_APP

**Description**

The schema that stores the app’s functions that are used by the consumer and/or the app’s stored procedures.

#### Schema (Versioned): PROCS\_APP

**Description**

The schema that contains the app’s stored procedures.

##### Stored Procedure: LOG\_SHARE\_INSERT

**Description**:

This procedure is shared with the app that inserts the initial app install logs into the logs table that is shared from the consumer to the provider.

**NOTE**: this stored procedure should only be executed after installation or upgrade/downgrade.

**Parameters**:

* **provider\_account\_locator** (VARCHAR) - the provider's Snowflake Account Locator
* **app\_code** (VARCHAR) - The abbreviated/shorthand name of the provider's app

##### Stored Procedure: REQUEST

**Description**:

This procedure is a “helper” stored procedure that allows the consumer to make a request to use an allowed app logic stored procedure. This procedure validates that the consumer can use the app, can access the specified stored procedure, and collects pre-set metrics about the request (i.e. input table record count, result record counts, etc.).

**Parameters**:

* **app\_code** (VARCHAR) - The abbreviated/shorthand name of the provider’s app
* **parameters** (VARCHAR) - The object containing input/output table names, the requested proc, and parameters to pass to the stored procedure.

## 

## Consumer Objects

The following objects are created to streamline app usage, but are not required to use the app.

**NOTE**: References to <APP\_CODE> refer to the shorthand/abbreviated name for the app (i.e. SDE for an app called Sample Data Enrichment).

### Database: SIDECAR

**Description**

Database created to store the SidecarRunner stored procedure.

#### Schema: Runner

**Description**:

This schema stores the SidecarRunner stored procedure.

##### Stored Procedure: SidecarRunner

**Description**:

This procedure executes the commands loaded in the SQL.SETUP table.

**Parameters**:

* **app\_name** (VARCHAR) - The name of the app, as installed in the consumer’s account.

The following objects are created when the consumer executes the SidecarRunner procedure.

### Role: C\_<APP\_CODE>\_APP\_ADMIN

**Description**

The admin role is created to create the log share to the provider, along with the stored procedures that help streamline app usage.

### Warehouse: C\_<APP\_CODE>\_APP\_WH

**Description**

The warehouse used to create the objects required to interact with the provider’s app.

### 

### 

### Database: C\_<APP\_CODE>\_HELPER\_DB

**Description**

Database that contains the consumer’s source data, shared data, and objects required to interact with the app.

#### Schema: SOURCE

**Description**

The schema that contains tables/views created from the source data tables.

##### Table/View: (SOURCE TABLES/VIEWS)

**Description**

The tables/views created from the source data tables.

#### Schema: RESULTS

**Description**

The schema that stores the results tables from each request, if applicable.

**NOTE**: This version of the Snowflake Native Apps framework does not support creating objects outside of the app.

##### Table: (RESULT TABLES)

**Description**

The results tables from each request.

#### Schema: PRIVATE

**Description**

Schema containing the helper stored procedures utilized to streamline the log share and request processes.

##### Stored Procedure: DETECT\_EVENT\_TABLE

**Description**:

This procedure detects if the consumer’s account has an event table and creates one if there isn’t one present.

**Parameters**:

* N/A

### 

### Database: EVENTS

**Description**

Database that contains the consumer’s event table (if the account does not already have one). Events in this table are shared with the provider to enable app usage (non-free app versions).

#### Schema: EVENTS

**Description**:

This schema stores the consumer events.

##### Table: EVENTS

**Description**

Table containing the consumer events.

**Definition**

See <https://docs.snowflake.com/en/developer-guide/logging-tracing/event-table-columns> for event table definition details.