snowflake®

SNOWFLAKE VALIDATED CONNECTOR REQUIREMENTS GUIDE

CHAMPION
GUIDES

INTRODUCTION

Snowflake's core principle is putting customers first. In order for our customers to have the best experience possible with our joint solutions, we work to enable our partners to build the most optimal technology integrations.

This guide walks through the key areas of integration: ease of use, performance, and security. By optimizing each of these areas, customers will feel confident they will have a Snowflake validated connector and an accelerated path to becoming data-driven.

VALIDATED CONNECTOR REQUIREMENTS GUIDE

VALIDATED CONNECTOR REQUIREMENTS CHECKLIST

1. Embedded Driver
2. UI Integration
3. Driver Currency
4. Connection String Identifier
5. Deployments: AWS and Azure
6. Loading Data into Snowflake:
   a. Internal Stage
   b. Bulk Load API
   c. Push Down Transformations
   d. Security: Key-Pair Authentication
7. Reading Data for Snowflake:
   a. Live Query (Query Pushdown)
   b. Bulk Unload API
   c. Security: OAuth/SSO

VALIDATED CONNECTOR REQUIREMENTS DETAILS

Embedded Driver

Snowflake connectivity works through an extensive network of connectors, drivers, programming languages, and utilities including JDBC, ODBC, Python, .NET, and so on. Partners develop solutions for connecting to Snowflake through our drivers and connectors. In order to make the experience frictionless for customers, partners should embed the driver into their runtime.

UI Integration

Allow customers to easily choose Snowflake as a connection and as a supported data source option from a pulldown menu or some other UI option to allow for a seamless experience connecting to Snowflake.

Driver Currency

Partners are required to implement a process to keep the embedded drivers as current as possible. Snowflake releases a new driver version typically 1–2 times per month. Partners should update drivers quarterly.

Connection String Identifier

Snowflake has the ability to identify details about connections made to customers' Snowflake instances. We use this information to help optimize customer environments. Partners can add a string to help identify which customers are using their tools. This helps us determine trends and usage and there is the potential to share this level of detail with our partners.

- Set the Application parameter in the connection string to some unique identifier, for example, Application='PartnerName'. This allows us to track all incoming connections and have a better view of adoption and usage.

Deployments

Support connectivity with Snowflake on both AWS and Azure

Loading Data into Snowflake

- Use the Snowflake bulk load API (COPY INTO) for optimized data loading.
- Use an internal named stage for data loading to avoid overhead for the customer.
- Push down transformations into Snowflake for optimized performance and cost.
- For security, add support for key-pair authentication to avoid passing credentials.

Reading Data from Snowflake

- Support live/direct query using query pushdown.
- Use the Snowflake bulk unload API (COPY INTO) for optimized data unloading.
- For security, add support for OAuth to enforce RBAC and avoid passing credentials.

NOTE: The bulk load APIs (PUT/GET and COPY INTO) are supported with current versions of the ODBC, JDBC, and Python Connector. Support for the .NET driver is on the product roadmap.

VALIDATED CONNECTOR BEST PRACTICES

BI and Analytics

Direct Query: This is the preferred approach because it enables customers to query the most current data without waiting for batch loads to complete. Snowflake is optimized for this use case through query pushdown.

1. Push down queries into Snowflake through a standard driver (JDBC, ODBC, and so on).
2. Embed the driver when possible.
3. Provide an integrated UI for selecting a Snowflake-specific option.

Extract Model: This model is not recommended but is supported. Instead of reading data directly over one of the drivers, we recommend using the bulk unload API to unload data to S3 and then pull the data in locally as CSV files.

1. Use the bulk fetch (COPY INTO) API to pull the data into an S3 internal stage and then read from there.
2. Use the GET API to pull the data from an internal stage into the BI cache.
3. Embed the driver when possible.
4. Provide an integrated UI for selecting a Snowflake-specific option.
5. Refer to "Unloading Data from Snowflake."

Data Integration (ETL, ELT, replication, and so on)

Loading Data into Snowflake:

1. First, break up the data into smaller chunk files on the source system (~100 MB compressed).
2. Call PUT in parallel to load the files.
   a. Use an internal named stage.
   b. Use a temp stage. Details on using different types of stages can be found in the following CREATE STAGE documentation.
   c. Data will be encrypted and compressed.
3. Run the COPY API (bulk load) to load data into a staging table in Snowflake.
   a. The table should already be created.
   b. Capture the response from Snowflake through the connector (Run COPY in validate mode).
   c. A file format encapsulates information, such as file type (CSV, JSON, and so on) and formatting options specific to each type, for data files used for bulk loading/unloading. A file format can be defined as a default for a CREATE STAGE command, or it can be specified as part of a COPY INTO command for specific file loads.
   d. Add any transformations such as mapping of columns, ignoring columns, and so on.
   e. Set the purge option to TRUE.
4. Push down additional transformation logic through SQL.
5. Combine all actions such as UPDATE, UPSERT, DELETE, INSERT, and so on through a MERGE statement.
   a. When possible, apply a filter (for example, use a WHERE clause), which will activate pruning and avoid a full table scan.
   b. Review clustering of the target table.
6. Use one session to handle all steps for better orchestration and management of rollbacks.
7. Define field sizes based on the data source. For example, when using VARCHAR, if you use the default, the size will be set to full (16 MB), which will have performance implications for BI and other tools reading these fields.

ABOUT SNOWFLAKE

Snowflake is the only data warehouse built for the cloud, enabling the data-driven enterprise with instant elasticity, secure data sharing, and per-second pricing, across multiple clouds. Snowflake combines the power of data warehousing, the flexibility of big data platforms, and the elasticity of the cloud at a fraction of the cost of traditional solutions. Snowflake: Your data, no limits. Find out more at **snowflake.com**