PUBLIC

SAP Cloud Integration for data services 1.0.11

2023-10-18

Help Center for SAP Cloud Integration for data services

THE BEST RUN SAP

Content

1 What Is SAP Cloud Integration for data services?

Integrate data between on-premise and cloud on a scheduled (batch) basis.

SAP Cloud Integration for data services is an ETL solution that extracts data from a variety of on-premise systems and then transforms the data using transformations and functions optimized for cloud applications. The data is loaded into cloud-based SAP applications such as SAP Integrated Business Planning. Predefined templates are provided for some use cases. You can also extract data from cloud-based SAP applications and load it into a variety of on-premise SAP and non-SAP systems.

Features

Extract data	Extract data from a variety of on-premise SAP systems, on-premise non-SAP systems, or cloud-based SAP applications.
Transform data	Transform data using transformations and functions that are optimized for cloud applications.
Load data	Load the data into cloud-based SAP applications such as SAP Integrated Business Planning.

Environment

This service is available in the Neo environment.

Prerequisites

For information about supported operating systems and web browsers, and for other important requirements, see the Product Availability Matrix.

2 Initial Setup

Follow these processes to set up your SAP Cloud Integration for data services environment.

Checklist: Setting Up [page 8]
: This checklist lists the steps required to set up SAP Cloud Integration for data services.

Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]

Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]

Checklist: Planning [page 11]
: This checklist provides a list of items that should be reviewed before moving data in SAP Cloud Integration for data services.

What is a Project? [page 12]
: A project is a container that groups related tasks and processes.

Checklist: Moving Your Data [page 14]
: This checklist provides a high-level overview of the steps required to move data to or from the cloud using SAP Cloud Integration for data services. It assumes the setup process is complete.

Test and Review [page 16]
: The following diagram provides a guideline to test the validity of tasks and preview the resulting data in SAP Cloud Integration for data services. The best practice is to get the first data flow working as planned before moving on to the next data flow or task.

Promoting a Task or Process [page 16]
: Promotion is the application lifecycle management tool in SAP Cloud Integration for data services. It allows you to copy and move a task or process from one environment to the next available environment, for example, from Sandbox to Production.

Run a Task or Process Immediately [page 18]
: Rather than waiting for a task or process to run at a later time, you can run it at the current time.

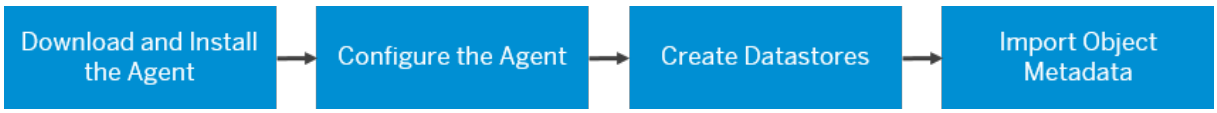
Schedule a Task or Process to Run Later [page 19]
: Set up a future time to run a task or process, either once or repeatedly.

Working in Multiple Environments [page 20]
: SAP Cloud Integration for data services comes with two environments (Sandbox and Production). The option to add additional environments is available.

Related Information

2.1 Checklist: Setting Up

This checklist lists the steps required to set up SAP Cloud Integration for data services.



√	Step	Description	More Information
	(Optional) Enable SAP Cloud Integration for data services pay-per-use (PPU) subaccount	Enable the SAP Cloud Integration for data services pay-per-use (PPU) subaccount to use the Cloud platform version of the product.	Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
	Download and install Data Service Agents to your on-premise locations.	Agents enable the secure transfer of data between your on-premise data sources and SAP Cloud Integration for data services.	SAP Data Services Agent
	Configure your agents.	Configuration is done in the web UI and in the host system.	SAP Data Services Agent
	Create datastores in the web UI.	Datastores connect SAP Cloud Integration for data services to your source and target databases and applications.	Create Datastores [page 25]
	Import object metadata into your datastores.	Object metadata such as database table and column names are used to map sources and targets for your data integration tasks.	Import Metadata Objects [page 132]

Parent topic: Initial Setup [page 7]

Related Information

Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]

Video: Required Setup Workflow

2.2 Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount

Ensure that you have completed the steps to create a subaccount using these instructions, Creating a Subaccount

To enable your SAP Cloud Integration for data services consumption-based license model (CPEA) subaccount, follow these steps.

> **i Note**
>
> SAP Cloud Integration for data services is only available on select Neo-based data centers.

> **! Restriction**
>
> Main tenant provisioning in sandbox and production environments is supported. Suborg provisioning is not supported.

1. In the SAP Business Technology Platform, navigate to the global account.
2. Select your previously created subaccount.
3. Find and select the *Data Integration* tile under *Services*.
4. Select the *Enable* button to activate the service.
5. Your organization's SAP Cloud Integration for data services service will be activated. You will receive an email notification containing a unique URL and and your SAP Cloud Integration for data services account information.

   > **i Note**
   >
   > The provisioning process may take up to 15 minutes.

   When you receive the email notification, navigate to the unique *Web UI URL* to access your SAP Cloud Integration for data services server and SAP Cloud Integration for data services organization information.

Task overview: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]

Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]

2.3 Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount

Complete the following before disabling the *Data Integration* service:

- Stop all connected agents.

  > **i** Note
  >
  > An agent is stopped only when its status is red.

- In the tenant's user interface under the Agent tab, delete all the agents from the agent's list.

To disable the *Data Integration* service:

1. In the SAP Cloud Platform cockpit, navigate to the global account.
2. Select your subaccount.
3. Find and select the *Data Integration* service tile and select *Disable*.

   When the decommissioning request is received, the text *Not Enabled* appears. You will receive an email notification when the tenant is deactivated and the organization is deleted.

Task overview: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Checklist: Planning [page 11]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]

2.4 Checklist: Planning

This checklist provides a list of items that should be reviewed before moving data in SAP Cloud Integration for data services.

√	Item	Details
	Business specifications	• Analyze business specifications to understand the data model. • Make sure that the target structure is finalized to avoid rework. • Verify in the Product Availability Matrix (PAM) that the required sources and targets are supported. • Determine if writeback to on-premise system is required. • For writeback, define the technique to load the data such as web services or files.
	Data mapping logic	• Identify the source tables and fields that the data should be extracted from. • Identify the target tables and fields that the data should be loaded to. • Understand any transformations that need to occur including filters, aggregations, and so on.
	Data load strategy	• Determine the schedule and frequency for the tasks to run. • Determine if you need full loads or a combination of full and delta loads (change data capture). • For delta loads, determine how changes in the source are identified.
	Template options	• Find out if predefined task templates are available. • Decide if any customization to the template is necessary to meet the business requirements.
	Data connectivity	• Identify technical connection information for source and target datastores (system names, usernames, passwords, and so on). • If you use files, make sure the file structure is defined.
	Naming convention	• Develop a meaningful naming convention, which enables easy navigation and organization.
	Environment check	• Log in and make sure that your internal Administrator has created an agent and the datastores.

Parent topic: Initial Setup [page 7]

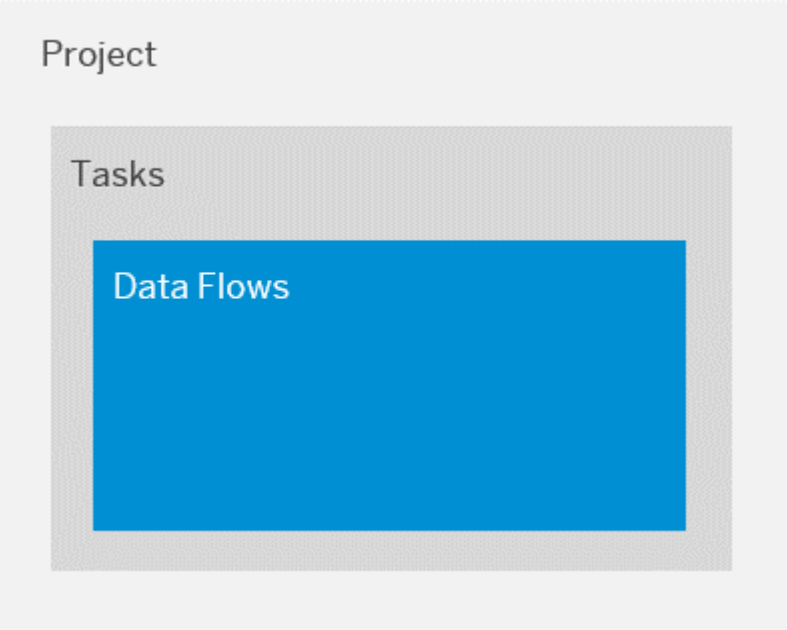
Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]
Product Availability Matrix (PAM)

2.5 What is a Project?

A project is a container that groups related tasks and processes.

The relationship between a project, tasks, and data flows is illustrated in the following diagram:



The *Projects* tab is where you can create and manage your projects, tasks, and processes. Most of the design work you do is launched from this tab.

The *Projects* tab contains a table of all the available projects in the current environment. Nested under each project are its tasks as well as any processes defined within the project. Each row includes details about the object, such as the status of its last run and "last changed by" information.

> **Note**
>
> The available actions differ based on the object selected (project, task, or process) and the environment, for example Sandbox or Production.

Behaviors on the Projects Tab

Filtering

You can filter the list of projects, tasks, and processes by clicking on the *Name* column heading and entering the keywords by which you want to filter. All names that contain the string you enter appear in the list. For example, if you enter a filter of **abc**, the resulting list of names might be `abc`, `123abc`, and `ABC678`. To reflect the connection of a task or process to its project, you may see the name of a project in a filtered list when the project contains a task or process that matches your filter criteria.

You can filter the list on the Projects tab using an asterisk (*) wildcard.

- Filtering with **abc*** returns names **starting with abc**.
- Filtering with ***abc** returns names **ending with abc**.
- Filtering with ***abc*** returns names **containing abc**.

Applying a filter containing two dots (..) such as **abc..def** creates an alphabetical range that returns all names between and including `abc*` and `def*`.

You can filter using greater than (>), greater than or equal to (>=), less than (<), less than or equal to <=), equal to (=), and different than (!=) operators. The system ranks characters in alphabetical order, as in **a < b**. These behave similarly to a between operator with a single argument. For example, `>=b` would return anything alphabetically after b*.

Filtering is not case-sensitive.

When you are viewing a filtered list, you see ▽ in the *Name* column heading.

When you have filtered the list and then perform an action on a selection, the system continues to display the filtered list on the *Projects* tab.

To return to the full, unfiltered list, do one of the following:

- Remove the keywords.
- Click the (Refresh) icon.
- Move to another tab and return to the *Projects* tab.
- Refresh your browser.

> **Note**
>
> Switching between environments such as Sandbox and Production clears the filters.

Selecting a Task or Process

When you have selected a task or process in the list and then perform an action on your selection, the task or process will still be selected on the *Projects* tab.

To unselect the task or process in the list, do one of the following:

- Click the (Refresh) icon.
- Click on another tab and return to the *Projects* list.
- Refresh your browser.

Sorting

When you open the *Projects* tab, the projects in the list and the processes and tasks beneath each project are sorted alphabetically.

Use the (Sort Ascending) and (Sort Descending) icons to sort the list as needed.

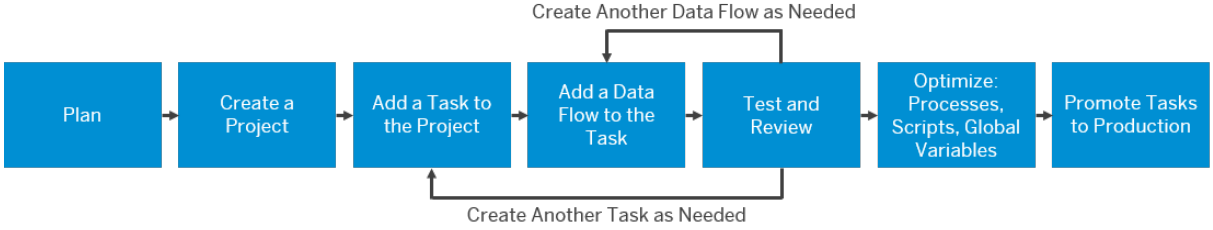
Parent topic: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]

2.6 Checklist: Moving Your Data

This checklist provides a high-level overview of the steps required to move data to or from the cloud using SAP Cloud Integration for data services. It assumes the setup process is complete.



√	Step	Description	More Information
	Begin with a solid plan.	Planning is the foundation of everything that is implemented in SAP Cloud Integration for data services.	Checklist: Planning [page 11]
	Create a project.	A project is a container that groups related tasks.	What is a Project? [page 12]
	Add a task to the project.	A task is the element that SAP Cloud Integration for data services executes at run-time. A task can contain one or more data flows.	Add Tasks to a Project [page 143]
	Add a data flow to the task.	A data flow defines what gets done to data on its way from one or more sources to a single target.	Add a Data Flow from Scratch [page 167]
	Test and review.	Testing the validity of your tasks and previewing the resulting data sets ensures that they work as expected.	Test and Review [page 16]
	Optional. Optimize with processes, scripts, and global variables	Processes, scripts, and global variables are designed to improve data loading, enhance customization, and reduce repetitive work.	What is a Process? [page 145] Scripts [page 223] Set Global Variables [page 235]
	Promote tasks to the next environment in your flow, for example from the Sandbox to Production.	Promoting tasks make them ready to run in your production environment.	Promoting a Task or Process [page 16]

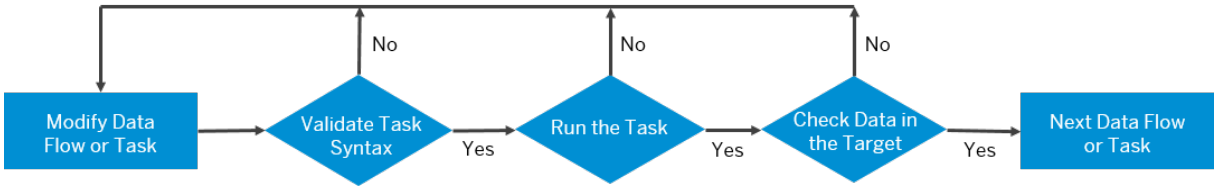
Parent topic: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]

2.7 Test and Review

The following diagram provides a guideline to test the validity of tasks and preview the resulting data in SAP Cloud Integration for data services. The best practice is to get the first data flow working as planned before moving on to the next data flow or task.



Parent topic: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]
What is a Task? [page 142]
Run a Task or Process Immediately [page 18]

2.8 Promoting a Task or Process

Promotion is the application lifecycle management tool in SAP Cloud Integration for data services. It allows you to copy and move a task or process from one environment to the next available environment, for example, from Sandbox to Production.

The application lifecycle often involves multiple environments, with each environment used for a different development phase. SAP Cloud Integration for data services comes with two environments, Sandbox and Production.

- Use the Sandbox environment to create and edit objects.
- Once the design phase is done, promote tasks and processes to the Production environment where no further modification is allowed on the objects to be executed.

Only a user with the Administrator role can promote a task or process.

You can modify tasks and processes in Sandbox after they have been promoted. Most changes do not affect the already-promoted version in the Production environment until they are promoted; changing the name of a task or process, however, directly takes effect in the next environment in the promotion path.

You may see the following icons in the Promoted column:

✓ The version of the task or process in this environment has been promoted to the next environment in the promotion path and the versions match.

⚠ The version of the task or process in this environment has been modified after being promoted and therefore does not match the version in the next environment in the promotion path. You must promote the modified task or process to the next environment for them to match.

Therefore, after editing a task or process, move the modified version to the next environment in your promotion path when you are ready by promoting it on the *Projects* tab. Promote the tasks within a process before promoting the process itself. For more information, see Edit a Task or Process [page 153].

If no projects exist in the Production environment when you promote a task or process from Sandbox to Production, the system creates a new project in Production called `Default` and places the promoted task or process into this project.

Datastore configurations

When a task or process is promoted from Sandbox to Production for the first time, its datastore configuration information is automatically carried over to the Production repository. The Administrator needs to edit and verify the datastore configuration information in the Production repository to make sure the datastore is pointing to the correct productive repository.

When a task or process is modified in the Sandbox environment, it may be promoted again. The changes that the Administrator has made in the Production datastore configurations will remain unchanged. The Sandbox datastore configuration information will not overwrite the configuration information and all defined objects in the Production repository. However, if needed, a user can *Include source datastore configurations* and *Include target datastore configurations* when re-promoting a task or process to overwrite the Production datastore configurations with the Sandbox datastore configurations.

Parent topic: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]

Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Run a Task or Process Immediately [page 18]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]
Working in Multiple Environments [page 20]

2.9 Run a Task or Process Immediately

Rather than waiting for a task or process to run at a later time, you can run it at the current time.

You can run tasks and processes in sandbox and production environments. After you have sufficiently tested and revised a task or process and promoted it from your sandbox to your production environment, you can run it in the production environment.

1. Go to the *Projects* tab.
2. Select the task or process that you want to run.
3. Select *Run Now*.
4. Enter or edit the information in the window that opens, and click *OK* to run the task or process.

> **i** Note
>
> Select *View History* to see recent details about tasks or processes.

Task overview: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]

Test and Review [page 16]
Promoting a Task or Process [page 16]
Schedule a Task or Process to Run Later [page 19]
Working in Multiple Environments [page 20]
What is a Task? [page 142]
What is a Process? [page 145]
Schedule a Task or Process to Run Later [page 19]
Troubleshooting During Task or Process Development [page 412]

2.10 Schedule a Task or Process to Run Later

Set up a future time to run a task or process, either once or repeatedly.

You can schedule tasks and processes to run in both sandbox and production environments. After you have sufficiently tested and revised a task or process and promoted it from your sandbox to your production environment, you can schedule it to run in the production environment.

1. Go to the *Projects* tab.
2. Select the task or process that you want to schedule.
3. Select *Schedule*.
4. Select *New* to create a new schedule for the selected task or process.
5. In the window that opens, specify details about this schedule, including a name, timing, and frequency. Note that if you select `Daily` for *Run Frequency*, *Repeat Every N Days* must be a number from 1 to 9999.
6. Select *Submit*.
7. To turn on the new schedule, select *Activate*.

> **i Note**
>
> Select *View History* to see recent details about tasks or processes that have run.

Task overview: Initial Setup [page 7]

Related Information

Daylight Savings Time with regard to Task and Process Schedules [page 20]
Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]

Checklist: Moving Your Data [page 14]
Test and Review [page 16]
Promoting a Task or Process [page 16]
Run a Task or Process Immediately [page 18]
Working in Multiple Environments [page 20]
What is a Task? [page 142]
What is a Process? [page 145]
Run a Task or Process Immediately [page 18]
Activate or Deactivate Multiple Schedules [page 387]
Troubleshooting During Task or Process Development [page 412]
Daylight Savings Time with regard to Task and Process Schedules [page 20]

2.10.1 Daylight Savings Time with regard to Task and Process Schedules

SAP Cloud Integration for data services recognizes Daylight Savings Time (DST) for locations where it is used, which may be important to you when choosing a time zone for a task or process schedule.

If you are in a location that **does not** follow Daylight Savings Time and you set the time zone for a schedule by selecting a location that **does** use DST, then the run time of the job will be different for you during Daylight Savings Time.

To have jobs run at the same time all year long, set a schedule's time zone to one that reflects your UTC offset and also contains a location that reflects whether you use Daylight Savings Time or not.

Related Information

Schedule a Task or Process to Run Later [page 19]

2.11 Working in Multiple Environments

SAP Cloud Integration for data services comes with two environments (Sandbox and Production). The option to add additional environments is available.

Your organization may have a flow similar to the flows shown below:

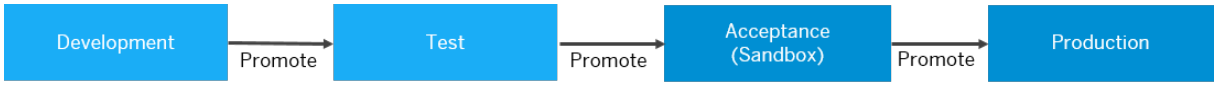
Or

SAP Cloud Integration for data services supports these flows by allowing additional organizations connected to your primary organization. Each of the additional organizations supports a single environment, such as Development or Test, and requires its own agent.

Promotion path

Objects must be promoted through the defined chain. For example, in the diagram below, tasks and processes would be promoted as follows:

1. Development to Test
2. Test to Acceptance (Sandbox)
3. Acceptance (Sandbox) to Production



Tasks or Processes can only be promoted by an Administrator.

Renaming objects

When a task, process or datastore that has already been promoted is renamed, the copy in the next environment in the chain is also renamed. However, copies in more distant environments are not renamed. In our example above, assume a task has been promoted through the entire environment chain. In the development environment, if the task is renamed, only versions in the Development and Test environments would take on the new name. The Acceptance (Sandbox) and Production versions would retain the old name until the next time the renamed object is promoted.

Parent topic: Initial Setup [page 7]

Related Information

Checklist: Setting Up [page 8]
Enabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 9]
Disabling SAP Cloud Integration for data services Consumption-Based License Model (CPEA) Subaccount [page 10]
Checklist: Planning [page 11]
What is a Project? [page 12]
Checklist: Moving Your Data [page 14]

3 Datastores

Datastores are the objects that connect SAP Cloud Integration for data services to your cloud and on-premise applications and databases. Through these connections, SAP Cloud Integration for data services can access metadata from and read and write data to your applications and databases.

Within the *Datastores* tab, you can create and manage datastores, which connect SAP Cloud Integration for data services to your applications and databases.

From this tab, you can:

- Create and delete datastores
- Test the connection to a datastore
- View and edit a datastore's configuration options (*Configuration)*
- Browse a datastore's metadata objects (*File Formats* or *Tables*)
- Import and manage file format or table metadata objects for a datastore
- View data loaded to a table in a target datastore to ensure it is correct

Enable SNC to provide a secure connection between SAP BW and the remote function call (RFC) server for jobs that you launch from SAP BW.

Related Information

3.1 What are Datastores?

Datastores are the objects that connect SAP Cloud Integration for data services to your cloud and on-premise applications and databases. Through these connections, SAP Cloud Integration for data services can access metadata from and read and write data to your applications and databases.

SAP Cloud Integration for data services supports datastores that include the following types of applications and databases:

- SAP Business Suite applications
- SAP BW sources
- SAP HANA application clouds
- SAP HANA cloud applications such as SAP Integrated Business Planning and SuccessFactors BizX
- Applications that have pre-packaged or user-written adapters
- Databases
- File format groups
- SOAP and REST Web services

The specific information that a datastore can access depends on its connection configuration. When your database or application changes, make corresponding changes in the datastore as it does not automatically detect the new information.

Parent topic: Datastores [page 23]

Related Information

Create Datastores [page 25]
Datastore Types and Their Properties [page 26]
Import Metadata Objects [page 132]
View Data in a Datastore [page 133]
Create or Copy Datastore Configurations [page 134]
Create System Configurations [page 135]
Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]

What are File Formats? [page 34]
Create Datastores [page 25]

3.2 Create Datastores

Create a datastore for each application or database you want to connect to SAP Cloud Integration for data services.

1. In the web UI, click the *Datastores* tab.
2. Click the + icon in the upper left corner.
3. In the *New Datastore* dialog, configure the necessary fields.

   The configuration options change according to the datastore type you select. For a detailed description of all possible datastore types and their properties, see the corresponding topics under Datastore Types and Their Properties [page 26].
4. Click *Save* to create the datastore.

After the datastore is created and saved, click *Test Connection* to verify the connection between SAP Cloud Integration for data services and the datastore's database or application.

Once the connection works, you can import metadata objects from the database or application into the datastore.

Task overview: Datastores [page 23]

Related Information

Importable Object Types [page 26]
What are Datastores? [page 24]
Datastore Types and Their Properties [page 26]
Import Metadata Objects [page 132]
View Data in a Datastore [page 133]
Create or Copy Datastore Configurations [page 134]
Create System Configurations [page 135]
Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]
What are Datastores? [page 24]
Import Metadata Objects [page 132]
View Data in a Datastore [page 133]

3.2.1 Importable Object Types

Once you have defined the datastore and its various connection properties, you can begin to import different objects to the datastore from the underlying data source.

These objects include:

- Tables
  A table is a collection of related data held in a table format within an SAP or non-SAP system. It consists of columns and rows.
- Extractors
  An extractor is a pre-defined SAP program that gathers data from various tables in an SAP source system, which is typically SAP ECC, then processes this data to create specific business content for insertion into another SAP system such as SAP BW or SAP IBP.
- Functions
  An SAP Function (or Function Module) is a pre-written custom program that typically extracts data from an SAP system and writes this to output fields or tables that can be read by SAP Cloud Integration for data services.

3.3 Datastore Types and Their Properties

Each type of SAP Cloud Integration for data services datastore has options that you configure depending on the underlying data source to which you are connecting.

DB2 [page 28]
Create a DB2 datastore to connect to a DB2 database.

File Format Group [page 30]
Create a File Format Group datastore to connect to a flat file.

File Location [page 43]
A file location object defines the location and transfer protocol for remote file objects.

Google BigQuery [page 53]
SAP Cloud Integration for data services supports using a Google BigQuery connection with an ODBC driver.

HANA [page 66]
Create a HANA datastore to connect to a HANA database.

Microsoft SQL Server [page 67]
Create a Microsoft SQL Server to connect to a Microsoft SQL Server database.

MySQL [page 69]
Create a MySQL datastore to connect to a MySQL database.

OData Adapter [page 71]
An OData Adapter datastore can extract and load data using two types of authentication.

ODBC Data Sources [page 80]
To work with ODBC data sources, drivers need to be configured on the Agent side.

Oracle [page 81]

Create an Oracle datastore to connect to an Oracle database.

REST Web Service [page 82]
Create a REST Web Service datastore to connect to a REST Web Service.

SAP Business Suite Applications [page 87]
Create an SAP Business Suite Application datastore to connect to an SAP Business Suite Application.

SAP BW Source [page 98]
Create an SAP BW Source datastore to connect to an SAP BW Source database.

SAP BW Target [page 103]
Create an SAP BW Target datastore to connect to an SAP BW Target database.

SAP Cloud Platform (SCP) HANA [page 108]
Create an SAP HANA application cloud datastore of application type HANA to connect to SAP Cloud Platform (SCP) HANA.

SAP Datasphere [page 109]
You can create an SAP Datasphere datastore to connect to an SAP Datasphere service.

SAP HANA Database [page 112]
Create an SAP HANA Database datastore to connect to an SAP HANA database.

SAP Integrated Business Planning [page 116]
Create an SAP HANA application cloud datastore of application type Integrated Business Planning to connect to SAP Integrated Business Planning.

SAP Integrated Business Planning via WebSocket RFC [page 117]
To connect to an SAP IBP instance via WebSocket RFC, create an SAP Cloud Integration for data services datastore with the following options/parameters.

SAP Lumira Cloud [page 120]
Create an SAP Lumira Cloud datastore to connect to an SAP Lumira Cloud database.

SOAP Web Service [page 121]
Create a SOAP Web Service datastore to connect to a SOAP-based web service.

SuccessFactors Adapter [page 124]
A SuccessFactors Adapter datastore can extract and load data to and from SuccessFactors using two types of authentication.

Sybase ASE [page 127]
Create a Sybase ASE datastore to connect to a Sybase ASE database.

Sybase IQ [page 128]
Create a Sybase IQ datastore to connect to a Sybase IQ database.

Teradata [page 130]
Create a Teradata datastore to connect to a Teradata database.

Workforce Analytics [page 131]
Create a Workforce Analytics datastore to connect to a Workforce Analytics database.

Parent topic: Datastores [page 23]

Related Information

What are Datastores? [page 24]
Create Datastores [page 25]
Import Metadata Objects [page 132]
View Data in a Datastore [page 133]
Create or Copy Datastore Configurations [page 134]
Create System Configurations [page 135]
Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]

3.3.1 DB2

Create a DB2 datastore to connect to a DB2 database.

DB2 database datastores support a number of specific configurable options. Configure the datastore to match your DB2 database.

Option	Possible values	Description
DB2 version	DB2 UDB `<version number>`	The version of your DB2 client. This is the version of DB2 that the datastore accesses.
Use Data Source (ODBC)	Yes No	Select to use a DSN to connect to the database. By default, this option is set to *Yes*. To use a DSN connection, you must also specify the *ODBC data source name*. If you set this option to *No*, you must also specify the *Database server name*, *Database name*, and *Port number* for a DSN-less connection.
ODBC data source name	Refer to the requirements of your database	The ODBC data source name (DSN) defined for connecting to your database. This option is required if *Use Data Source (ODBC)* is set to *Yes*.
Database server name	Refer to the requirements of your database	The DB2 database server name. This option is required if *Use Data Source (ODBC)* is set to *No*.
Database name	Refer to the requirements of your database	The name of the database defined in DB2. This option is required if *Use Data Source (ODBC)* is set to *No*.

Option	Possible values	Description
Port number	Integer	The number of the database port. This option is required if *Use Data Source (ODBC)* is set to *No*.
User name	Alphanumeric characters and underscores	The user name of the account through which the software accesses the database.
Password	Alphanumeric characters, underscores, and punctuation	The password of the account through which the software accesses the database.
Bulk loader directory	Directory path	The location where command and data files are written for bulk loading. You can also enter a variable for this option.
Bulk loader user name	Alphanumeric characters and underscores or blank	The name used when loading data with the bulk loader option. For bulk loading, you might specify a different user name. For example, specify a user who has import and load permissions.
Bulk loader password	Alphanumeric characters, underscores, and punctuation, or blank	The password used when loading with the bulk loader option.
DB2 server working directory	Directory path	The working directory for the load utility on the computer that runs the DB2 server. This option is required whenever the DB2 server and the SAP Data Services Agent run on separate host machines.
FTP host name	Computer name, fully qualified domain name, or IP address	If this field is left blank or contains the name of the SAP Data Services Agent host system, the software assumes that DB2 and the software share the same host system, and that FTP is unnecessary. When FTP is not required, all other FTP-related fields can remain blank.
FTP login user name	Alphanumeric characters and underscores, or blank	Required to use FTP.
FTP login password	Alphanumeric characters, underscores, and punctuation, or blank	Required to use FTP.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Aliases	-	Enter the alias name and the owner name to which the alias name maps.

3.3.2 File Format Group

Create a File Format Group datastore to connect to a flat file.

File Format Group datastores support a number of specific configurable options. The options defined in a file format group are inherited by all the individual file formats that it contains. Configure the file format group to match the data in the flat files that you want the software to access while it executes tasks.

Option	Possible values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the file format group.
Description	Any text	The description of the file format group.
Agent	The list of agents that have been defined in the agents tab	Specifies the agent that should be used to access this data source.
Location	At Agent (default) and any defined file location objects	At Agent is on the local machine. Any FTP or SFTP file location objects that you set up using the File Locations datastore are also listed here. **i Note** Test connection is always enabled for the file format group datastore, but it is useful only when *Location* is *At Agent*.
Root directory	Path name on the SAP Data Services Agent host system	The directory where the source or target files are located. **i Note** The SAP Data Services Agent must also be configured to have access to the directory that contains the source or target files. For more information, see the *Agent Guide*.

Option	Possible values	Description
Adaptable Schema	Yes No	Indicates whether the schema of the file formats are adaptable or fixed. • *Yes* indicates that the schema is adaptable. The actual file can contain fewer or more columns than indicated by the file format. If a row contains fewer columns than expected, the software loads null values into the columns missing data. If a row contains more columns than expected, the software ignores the additional data. • *No* indicates that the schema is fixed. The software requires the number of columns in each row to match the number of columns specified in the file format. The default is *No*. If you select *Yes*, you must ensure that the selected column delimiter and new-line characters do not appear inside the actual data.
Parallel process threads	Integer	Specifies the number of threads for parallel processing, which can improve performance by maximizing CPU usage on the SAP Data Services Agent host system.
Escape Character	Any character sequence or empty	A special character sequence that causes the software to ignore the normal column delimiter. Characters following the escape character sequence are never used as column delimiters. For example, suppose you specify a forward slash as the escape character and a comma as the column delimiter. Then, you must have a forward slash to have a comma appear inside a field.
Null indicator	<Null> or any other character sequence	Special character sequence that the software interprets as NULL data.
Date Format	yyyy.mm.dd or other combinations	The date format for reading or writing date values to and from the file.
Time Format	hh24:mi:ss or other combinations	The time format for reading or writing time values to and from the file.
Date-time Format	yyyy.mm.dd hh24:mi:ss or other combinations	The date-time format for reading or writing date-time values to and from the file.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.

Option	Possible values	Description
Code page	-	Specify the character encoding of character data in the datastore.

SFTP options

> **Note**
>
> If you want to connect to a datastore using SFTP, it is recommended that you do so using the File Location datastore's SFTP option instead of File Format Group's SFTP option. The File Format Group SFTP option may be deprecated in the future. See File Location [page 43].

File format group datastores can also be configured to connect to a server using the SSH File Transfer Protocol (SFTP). When you use SFTP, the SAP Data Services Agent reads or writes the data file through an SSH connection to the host defined in the SFTP options.

> **Note**
>
> When a file is transferred to an external server using SFTP, a copy of the file remains in the Agent root directory.

Option	Possible values	Description
Enable SFTP	Yes No	Enables or disables SFTP connectivity for the file format group.
SFTP host	Alphanumeric characters and periods	The fully-qualified hostname of the SFTP server.
SFTP port	Integer	The port the SAP Data Services Agent uses to connect to the SFTP host.
Use proxy	Yes No	Specifies whether or not to use a proxy configuration when connecting to the SFTP host. The proxy configuration is defined in the SAP Data Services Agent. For more information, see the *Agent Guide*.
Verify SFTP host	Yes No	Specifies whether to verify the identity of the SFTP server host.
Verification method	Host public key fingerprint Known hosts file	The method to use to verify the identity of the SFTP host. **Note** When you use known hosts file verification, the SFTP host is verified against the known hosts file configured on the SAP Data Services Agent host machine.

Option	Possible values	Description
Host public key fingerprint	MD5 checksum	The 128-bit MD5 checksum of the SFTP host's public key.
User name	Alphanumeric characters	The user name used to connect to the SFTP host.
Authentication method	Password Public key	The authentication method used to connect to the SFTP host.
Password	Alphanumeric characters	The password used to connect to the SFTP host. Required only when using the password authentication method.
Private key file name	Folder path and file name	The full folder path and file name of the private key file located on the SAP Data Services Agent host system. We recommend you to keep the keys inside the folder path `<DS_COMMON_DIR>/conf/keys/sftp`. If you do so, map the folder inside your Agent Configuration > Configure Directories as instructed in Managing Allowlisted Directories. Required only when using the public key authentication method. **i Note** SAP Cloud Integration for data services supports key files generated only in the OpenSSH format. Tools such as `ssh-keygen` can create key files in this format. Other tools, such as PuTTY, may not use the OpenSSH format, and the generated key files will be incompatible.
Decryption passphrase	Alphanumeric characters	The passphrase used to decrypt the private key file. Required only when using the public key authentication method.

Option	Possible values	Description
Public key file name	Folder path and file name	The full folder path and file name of the private key file located on the SAP Data Services Agent host system. We recommend you to keep the keys inside the folder path `<DS_COMMON_DIR>/conf/keys/sftp`. If you do so, map the folder inside your Agent Configuration > Configure Directories as instructed in Managing Allowlisted Directories. Required only when using the public key authentication method. **i Note** SAP Cloud Integration for data services supports key files generated only in the OpenSSH format. Tools such as `ssh-keygen` can create key files in this format. Other tools, such as PuTTY, may not use the OpenSSH format, and the generated key files will be incompatible.

Related Information

What are File Formats? [page 34]
PGP Management
File Format Options [page 37]
Duplicate a Data Flow [page 166]

3.3.2.1 What are File Formats?

A file format is a set of properties that describes the metadata structure of a flat data file. File formats allow the software to access flat data files on an SAP Data Services Agent host system, and read from or write to those files while the software executes a task or process.

File format groups

Within the software, file formats are organized in a specialized type of datastore called a file format group. In each file format group, you can define any number of individual file formats. Each file format may describe a specific file, or be a generic description that can be used for multiple data files.

Creating file formats

You have three options to create file formats:

Option	Description
Create from sample	Create a file format based on a comma-separated values (CSV) file on your local system.
Create from tables	Create a file format based on an existing table or file in a datastore. You can choose multiple tables in a selected data-store to create multiple file formats all at once.
Create from scratch	If neither a file nor a table is available, you can create a file format from scratch.

After you create a file format, you can modify its properties.

> **i** Note
>
> The source files for File Format datastores need to be placed into a folder that is defined for the SAP Cloud Integration for data services Agent. For more information, see Managing Allowlisted Directories.

Using XML file templates

An XML template is a special type of file format that you can use to write structured, hierarchical data to an XML file on the SAP Data Services Agent host system.

When you want to write to an XML file, you must use a Target XML Map transform as the final step in your data flow. Unlike other file formats, XML templates do not have any column or option definitions. Instead, the hierarchical structure is inherited from the output schema of the Target XML Map transform.

Using XSD Schema XML file

An XSD Schema XML file is another special type of file format that you can use to read and write structured, hierarchical data from and to an XML file on the SAP Data Services Agent host system.

You can import XSD metadata document files, and use this XSD as definition for your XML source and target files, in jobs. XML documents are hierarchical. Their valid structure is stored in a file format group and can be mixed with flat files (XML template is already there).

The format of the XML data file is always specified by one or more XML Schema documents (XSD). When multiple XSDs are used, they should be combined in a zip archive. When an XSD or XSD archive is imported, the software creates a hierarchical schema based on the schema from the XSD.

> **Note**
>
> If there is more than one element available within the XML schema, then select a name in the namespace drop-down list to identify the imported XML Schema.

Related Information

File Format Options [page 37]
File format error handling [page 39]
File Format Group [page 30]

3.3.2.1.1 File Format Options

File formats support a number of specific configurable options. Configure the file format to match the structure of the flat file that you want the software to access while it executes tasks or processes.

Option	Possible values	Description
Name	Alphanumeric characters, underscores, global variables	The name of the object. The name appears in the *File Formats* tab of a file format group datastore and in data flows that use the file format. **i Note** Each file format name should be globally unique within an environment landscape such as Sandbox or Production. You cannot have the same file format name in a different file format group. **→ Tip** Global variables can be used as file names. For example, if a file name includes a date stamp (`Product_20170428.csv`, `Product_20170429.csv`, and so on), a pre-load script could contain a statement that creates the value for the global variable. The script might include the following statement: `$G_FILENAME = 'File_Product_' || to_char(sysdate(), 'YYYYMMDD') || '.csv';`
Description	Any text	The description of the file format.
Column Delimiter	Tab Semicolon Comma Space Other	The character sequence that indicates the end of one column and the beginning of the next. If a column delimiter of "Other" is chosen, SAP Cloud Integration for data services will recognize this delimiter, however it will not create datastore columns automatically. You will need to manually define your datastore columns.

Option	Possible values	Description
Newline Style	Default Windows Unix	The character sequence that indicates when one row of data ends and the next begins. If you choose Default, the newline style will match the operating system of the SAP Data Services Agent host system that is used while executing the task or process.
Text Qualifier	Single quotation marks (') Double quotation marks (") None	Denotes the start and end of a text string. All characters (including those specified as column delimiters) between the first and second occurrence of this character are considered to be a single text string. **i Note** Data in columns cannot include the column delimiter unless you also specify a text delimiter. For example, if you specify a comma as the column delimiter, none of the data in the file can contain commas. However, if you specify a comma as the column delimiter and a single quote as the text delimiter, commas are allowed in strings in the data.
Skip top rows	Integer	The number of rows that are skipped when reading the file. You can specify a non-zero value when the file includes comments or other non-data information.
First row contains column headers	Selected Unselected	Indicates whether the first row of data in the file contains the column names and should be skipped when reading the file. The software uses this option in addition to the *Skip top rows* option. When you select this option, the software does not read data from the first row, and uses the data in the first row to determine the file's column names.

Option	Possible values	Description
File Header	A string containing a combination of the following options: • Column delimiter (**COLDELIM**) • Column names • Global variables • Plain text	The format of the header row to prepend to the output. **Note** **COLDELIM**, column names, and global variables must be surrounded by square brackets []. For example, **Benefits[COLDELIM][$G_LOAD_DATE]**. In this example, **$G_LOAD_DATE** is a global variable defined in the execution properties.
File Footer	A string containing a combination of the following options: • Column delimiter (**COLDELIM**) • Column names • Global variables • Plain text	The format of the footer row to append to the output. **Note** **COLDELIM**, column names, and global variables must be surrounded by square brackets []. For example, **Total[COLDELIM][ROWCOUNT]**. In this example, **ROWCOUNT** is a column defined in the file format as a decimal column that is used only in the file footer or header.

Related Information

File Format Group [page 30]

3.3.2.1.2 File format error handling

To specify how the software handles errors and warnings when processing data from the file format, set options in the *Error Handling* group in the File Format editor.

Access the *Error Handling* group when you create or edit a file format.

Table 1: Error Handling option descriptions

File format option	Description
Error Handling group	
Log data conversion warnings	Specifies whether the software includes data type conversion warnings in the error log. • *Yes*: Includes data type conversion warnings in the error log. *Yes* is the default setting. • *No*: Does not include data type conversion warnings in the error log. Applicable for new, edit, and source modes.
Log row format warnings	Specifies whether the software includes row format warnings in the error log. • *Yes*: Includes row format warnings in the error log. *Yes* is the default setting. • *No*: Does not include row format warnings in the error log. Applicable for new, edit, and source modes.
Log warnings	Specifies whether the software logs warnings for unstructured file formats. • *Yes*: Logs warnings for unstructured file formats. *Yes* is the default setting. • *No*: Does not log warnings for unstructured file formats. **i Note** Option appears only when you select *Unstructured Text* for *Type*. The software includes warnings for the following situations: • No files in the specified directory • No files match specified filter • Skipping irregular files on UNIX. For example, a FIFO, symbolic link, character or block device, or UNIX socket. • Empty file and *Skip empty files* set to *Yes* Applicable for new, edit, and source modes.

File format option	Description
Maximum warnings to log	Specifies the maximum number of warnings the software logs. • Positive integer greater than 0. • *{no limit}* Applicable when you set *Log data conversion warnings* or *Log row format warnings* to *Yes*. Applicable for new, edit, and source modes.
Capture data conversion errors	Specifies whether the software captures data type conversion errors for flat file sources. • *Yes*: Captures data type conversion errors for flat file sources. • *No*: Does not capture data type conversion errors for flat file sources. *No* is the default setting. Applicable for new, edit, and source modes.
Capture row format errors	Specifies whether the software captures row format errors for flat file sources. • *Yes*: Captures row format errors for flat file sources. *Yes* is the default setting. • *No*: Does not capture row format errors for flat file sources. Applicable for new, edit, and source modes.
Capture file access errors	Specifies whether the software captures file access errors for flat file sources. • *Yes*: Captures file access errors for flat file sources. *Yes* is the default setting. • *No*: Does not capture file access errors for flat file sources. Applicable for new, edit, and source modes.
Capture string truncation errors	Specifies whether the software captures string truncation errors for flat file sources. • *Yes*: Captures string truncation errors for flat file sources. • *No*: Does not capture string truncation errors for flat file sources. *No* is the default setting. Applicable for new, edit, and source modes.

File format option	Description
Maximum errors to stop job	Specifies the maximum number of invalid rows the software processes before stopping the job. • Integer greater than 0. • *{no limit}*. *{no limit}* is the default setting. Applicable when *Capture data conversion errors* or *Capture row format errors* are set to *Yes*. Applicable for new, edit, and source modes.
Write error rows to file	Specifies whether the software writes invalid rows to an error file. • *Yes*: Writes error rows to error file. Also specify *Error file root directory* and *Error file name*. • *No*: Does not write error rows to error file. *No* is the default setting. Applicable for new, edit, and source modes.
Error file root directory	Specifies the location of the error file. • Directory path • Blank • Select a variable **i Note** If you enter a directory path for this option, enter only a file name for *Error file name* option. If you leave this option blank, enter the full path and file name in *Error file name*. Applicable only when you select *Yes* for *Write error rows to file*. Applicable for new, edit, and source modes.

File format option	Description
Error file name	Specify the file name for the error file. • File name if you only entered the directory path for *Error file root directory*. • File name including full path if you left *Error file root directory* blank. • Blank • Variable **i Note** Set the variable to a specific file with full path name. Use variables to specify file names that you cannot enter such as file names that contain multibyte characters. Applicable for new, edit, and source modes.

3.3.3 File Location

A file location object defines the location and transfer protocol for remote file objects.

! Restriction

Running a task that includes a file location object requires Data Services Agent version 1.0.11 Patch 34 or later.

The software supports the following protocols:

- FTP
- SFTP
- Azure Cloud Storage
- Azure Data Lake Storage

The software uses the remote and local server information and the file transfer protocols to move data between the local and remote server.

After following any of the protocols listed above, you can read and write data to or from a remote server by selecting it as the *Location* in your file format datastore.

Related Information

Create a File Location Object [page 44]
Associate a File Format Group with a File Location Object [page 52]

Create or Copy Datastore Configurations [page 134]

3.3.3.1 Create a File Location Object

Create a file location object and specify a file transfer protocol to set local and remote server locations for source and target files.

To specify these file transfer protocols, follow the steps below:

- FTP
- SFTP
- Azure Cloud Storage
- Azure Data Lake Storage Gen1 and Gen2

1. In the *Datastores* tab, click the (New Datastore) icon to create a new datastore configuration.
2. Complete the following fields, being sure to select **`File Location`** as the *Type*:

Option	Possible values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the *Datastores* tab and in tasks that use this datastore.
Description	Any text	Optional. Record special or identifying information about this datastore.
Type	A list of available datastore types, including *File Location*.	Selecting *File Location* allows you to choose a protocol of FTP, SFTP, Azure Cloud Storage, or Azure Data Lake Storage.
Agent	The list of agents that have been defined in the agents tab	Specifies the agent to use to access this data source.
Protocol	FTP, SFTP, Azure Cloud Storage, or Azure Data Lake Storage	This selection determines the remaining fields to populate.

3. Based on the *Protocol* you have selected, define the appropriate parameters shown in the sections below:
   - **FTP Options**

Option	Possible values	Description
Host Name	Computer name, fully qualified domain name, or IP address of the FTP server	Specifies the remote server name of the FTP server.
Port	21 (default)	Specifies the remote server port number of the FTP server.
User Name	Alphanumeric characters and underscores	Specifies the remote server user name of the FTP server.
Password	Alphanumeric characters and underscores, or blank	Specifies the remote server password associated with the FTP server.

Option	Possible values	Description
Connection Retry Count	Number	Specifies the number of times the software can try to connect to the server.
Connection Retry Interval	Number	Specifies the time in seconds between which the software waits to retry connecting to the server.
Local Directory	Path name on the SAP Data Services Agent host system	The directory where the source or target files are located. The SAP Data Services Agent must also be configured to have access to the directory that contains the source or target files. For more information, see the *SAP Data Services Agent Guide*.
Remote Directory	Relative path to the root directory of FTP or SFTP. Empty if the files are located at the root directory.	Optional. Specifies the file path to the remote server.

- SFTP Options

Option	Possible values	Description
Host Name	Computer name, fully qualified domain name, or IP address	Specifies the remote server name.
Port	22 (default)	Specifies the remote server port number.
Host Public Key Fingerprint	MD5 checksum	The 128-bit MD5 checksum of the SFTP host's public key.
Authorization Type	Password or Public Key	The authentication method used to connect to the SFTP host.
User Name	Alphanumeric characters and underscores	Specifies the user name for the specified remote server.
Password	Alphanumeric characters and underscores, or blank	Specifies the password related to the user for the remote server.
Private Key File Name	File name	The name of the private key file located in `<DS_COMMON_DIR>/conf/keys/sftp` on the SAP Data Services Agent host system. Required only when using the public key authentication method. **i Note** SAP Cloud Integration for data services supports key files generated only in the OpenSSH format. Tools such as `ssh-keygen` can create key files in this format. Other tools, such as PuTTY, may not use the OpenSSH format, and the generated key files will be incompatible.

Option	Possible values	Description
Decryption Pass Phrase	Alphanumeric characters	The passphrase used to decrypt the private key file. Required only when using the public key authentication method.
Public Key File Name	File name	The name of the public key file located in `<DS_COMMON_DIR>/conf/keys/sftp` on the SAP Data Services Agent host system. Required only when using the public key authentication method. **Note** SAP Cloud Integration for data services supports key files generated only in the OpenSSH format. Tools such as `ssh-keygen` can create key files in this format. Other tools, such as PuTTY, may not use the OpenSSH format, and the generated key files will be incompatible.
Connection Retry Count	Number	Specifies the number of times the software can try to connect to the server.
Connection Retry Interval	Number	Specifies the time in seconds between which the software waits to retry connecting to the server.
Local Directory	Path name on the SAP Data Services Agent host system	The directory where the source or target files are located. The SAP Data Services Agent must also be configured to have access to the directory that contains the source or target files. For more information, see the *SAP Data Services Agent Guide*.
Remote Directory	Relative path to the root directory of FTP or SFTP. Empty if the files are located at the root directory.	Optional. Specifies the file path to the remote server.

- **Azure Cloud Storage Options**

Option	Description
Account Name	Specifies the name for the Azure storage account in the Azure Portal.
Storage Type	Specifies the storage type to access. *Container*.

Option	Description
Authorization Type	Indicates whether you use an account-level or service-level storage access signature (SAS). If you use a service-level SAS, indicate whether you access a resource in a file (blob) or in a container service. • *Primary Shared Key*: Authentication for Azure Storage Services using an account-level SAS. Accesses resources in one or more storage services. • *File (Blob) Shared Access Signature*: Authentication for Azure blob storage services using a service-level SAS. Select to access a specific file (blob). • *Container Shared Access Signature*: Authentication for Azure container storage services using a service-level SAS. Select to access blobs in a container.
Shared Access Signature URL	Specifies the access URL that enables access to a specific file (blob) or blobs in a container. Azure recommends that you use HTTPS instead of HTTP. To access blobs in a container, include the following elements: `https://<storage_account_name>/<container_name>/<signature value>` To access a specific file (blob), include the following elements: `https://<storage_account_name>/<container_name>/<file_name>/<signature value>`
Account Shared Key	Specifies the Account Shared Key. Obtain a copy from the Azure portal in the storage account information. **Note** For security, the software does not export the account shared key when you export a data flow or file location object that specifies *Azure Cloud Storage* as the protocol.
Connection Retry Count	Specifies the number of times the computer tries to create a connection with the remote server after a connection fails. The default value is 10. The value cannot be zero. After the specified number of retries, the software issues an error message and stops the job.
Batch Size for Uploading Data	Specifies the maximum size of a data block per request when transferring data files. The limit is 4 MB. **Caution** Accept the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.

Option	Description
Batch Size for Downloading Data	Specifies the maximum size of a data range to be downloaded per request when transferring data files. The limit is 4 MB. **⚠ Caution** Accept the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.
Number of Threads	Specifies the number of upload and download threads for transferring data to Azure Cloud Storage. The default value is 1. When you set this parameter correctly, it could decrease the download and upload time for blobs.
Local Directory	Specifies the directory where the source or target files are located. The SAP Data Services Agent must also be configured to have access to the directory that contains the source or target files. For more information, see the *SAP Data Services Agent Guide*.
Remote Path Prefix	Optional. Specifies the file path for the remote server, excluding the server name. You must have permission to this directory. If you leave this option blank, the software assumes that the remote path prefix is the user home directory used for FTP. When an associated file format is used as a reader in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing. When an associated file format is used as a loader in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory. Container type storage is a flat file storage system and it does not support subfolders. However, Microsoft allows forward slashes with names to form the remote path prefix, and a virtual folder in the container where you upload the files. **Example** You currently have a container for finance database files. You want to create a virtual folder for each year. For 2021, you set the remote path prefix to: `2021/`. When you use this file location, all of the files upload into the virtual folder “2021”.

Option	Description
Container	Specifies the Azure container name for uploading or downloading blobs to your local directory. If you specified the connection information, including account name, shared key, and proxy information (if applicable), select *Container*. The software sends a request to the server for a list of existing containers for the specific account. Either select an existing container or specify a new one. When you specify a new one, the software creates it when you run a job using this file location object.
Proxy Host, *Proxy Port*, *Proxy User Name*, *Proxy Password*	Optional. Enter the same proxy information as when you configured the agent during installation.

- **Azure Data Lake Storage Gen1 Options**

Option	Description
Version	Select *Azure Data Lake Gen1*.
Data Lake Store Name	Name of the Azure Data Lake Store to access.
Service Principal ID	Obtain from your Azure administrator.
Tenant ID	Obtain from your Azure administrator.
Password	Obtain from your Azure administrator.
Connection Retry Count	Specifies the number of times SAP Cloud Integration for data services can try to connect to the server.
Batch Size for Uploading Data	Maximum size of a data block to upload per request when transferring data files. The default setting is 5 MB. ⚠ Caution Keep the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.
Batch Size for Downloading Data	Maximum size of a data range to download per request when transferring data files. The default setting is 5 MB. ⚠ Caution Keep the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.
Number of Threads	Number of parallel uploaders or downloaders to run simultaneously. The default value is 1.
Local Directory	Specifies the directory where the source or target files are located. The SAP Data Services Agent must also be configured to have access to the directory that contains the source or target files. For more information, see the *SAP Data Services Agent Guide*.

Option	Description
Remote Path Prefix	Directory path for your files in the Azure Data Lake Store. Obtain the directory path from Azure Data Lake Store Properties. **Example** If the directory in your Azure Data Lake Store Properties is `adl://<yourdatastoreName>.azuredatalakestore.net/<FolderName>/<subFolderName>`, the remote path prefix value is `<FolderName>/<subFolderName>`. Permission to access this directory is required.
Proxy Host, *Proxy Port*, *Proxy User Name*, *Proxy Password*	Optional. Enter the same proxy information as when you configured the agent during installation.

- **Azure Data Lake Storage Gen2 Options**

Option	Description
Version	Select *Azure Data Lake Gen2*.
Authorization Type	Select *Shared Key* or *Service Principal*.
Account Shared Key	When *Authorization Type* is set to *Shared Key*, enter the account shared key you obtain from your Azure Data Lake Store administrator.
Communication Protocol/Endpoint URL	Enter **`https`**. You can also enter the endpoint URL.
Service Principal ID	Obtain from your Azure Data Lake Store administrator.
Tenant ID	Obtain from your Azure Data Lake Store administrator.
Password	Obtain from your Azure Data Lake Store administrator.
Connection Retry Count	Specifies the number of times SAP Cloud Integration for data services should try to connect to the server.
Batch size for uploading data (MB)	Maximum size of a data block to upload per request when transferring data files. The default is 10 MB; Microsoft suggests setting this value within the range of 4 MB to 16 MB for better performance. **⚠ Caution** Keep the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.

Option	Description
Batch size for downloading data (MB)	Maximum size of a data range to download per request when transferring data files. The default is 10 MB; Microsoft suggests setting this value within the range of 4 MB to 16 MB for better performance. **⚠ Caution** Keep the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.
Number of threads	Number of parallel uploaders or downloaders to run simultaneously. The default value is 1.
Remote Path Prefix	Directory path for your files in the Azure Data Lake Store. Obtain the directory path from Azure Data Lake Store Properties. **Example** If the directory in your Azure Data Lake Store Properties is `adl://<yourdatalakeaccountName>.dfs.core.windows.net/<containerName>/<FolderName>/<subFolderName>`, the remote path prefix value is `<FolderName>/<subFolderName>`. Permission to access this directory is required.
Local Directory	Path to the local directory for your local Data Lake Store data. Permission to access this directory is required.
Container	May contain only lowercase letters, numbers, and hyphens, and must begin with a letter or a number. Each hyphen must be preceded and followed by a non-hyphen character. The name must also be between 3 and 63 characters long.
Proxy Host, *Proxy Port*, *Proxy User Name*, *Proxy Password*	Optional. Enter the same proxy information as when you configured the agent during installation.

4. Click *Save*.

   You have specified the file transfer protocol and can associate a file format group with one of the protocols above in order to read or write data to a local or remote location.

Related Information

Associate a File Format Group with a File Location Object [page 52]
File Location [page 43]
Create or Copy Datastore Configurations [page 134]

3.3.3.2 Associate a File Format Group with a File Location Object

Associate a file format group with an FTP, SFTP, Azure Cloud Storage, or Azure Data Lake Storage protocol in order to read or write data to a local or remote location.

To read or write data to a local or remote location and specify the type of data to be transferred, follow these steps:

1. In the *Datastores* tab, click the plus button to create a new datastore.

   > **i Note**
   >
   > You can also change the *Location* of an existing datastore in its *Configuration* details.

2. Enter the *Name* of the datastore. This name appears in the datastores tab and in tasks that use this datastore.
3. (Optional) Enter a *Description* of the datastore.
4. Select an *Agent* to use to access this data source.
5. In the *Type* list, select File Format Group.
6. In the *Location* list, specify your previously created File Location Object name, so SAP Cloud Integration for data services will know how to connect to your remote data source.
7. Click *Save*.

   You can now create tasks using the datastore to read or write data to a local or remote location.

Related Information

File Location [page 43]
Create a File Location Object [page 44]
Create or Copy Datastore Configurations [page 134]

3.3.4 Google BigQuery

SAP Cloud Integration for data services supports using a Google BigQuery connection with an ODBC driver.

> **i Note**
>
> If you plan to use a Google BigQuery datastore as a source, the target must be an SAP Integrated Business Planning (IBP) WebSocketRFC datastore.

Prerequisite: You must install the Simba ODBC driver on the agent machine. For more information, see Download and install the Simba ODBC driver [page 57].

To access tables from your Google BigQuery projects, create a Google BigQuery ODBC datastore using either a data source name (DSN) or a server name (DSN-less) connection.

Table 2: Google BigQuery datastore options

Option	Description
Name	The name of the object. Alphanumeric characters and underscores. This name appears in the datastores tab and in tasks that use the datastore.
Description	The description of the datastore.
Type	Select *Google BigQuery*.
Agent	The list of agents that have been defined in the *Agents* tab. Specifies the agent that should be used to access this data source.
Database Type	Select *Google BigQuery ODBC*.
Use Data Source (ODBC)	Select **Yes** or *No* to indicate whether you configured the driver with ODBC. **i Note** Before you configure this datastore, configure a DSN for the Simba ODBC driver for Google BigQuery using the ODBC Data Source Administrator for Windows or the SAP Data Services (DS) Connection Manager for Linux.
ODBC data source name	Select the DSN name from the dropdown list. Required when *Use Data Source (ODBC)* is set to **Yes**. **i Note** The dropdown list contains only existing DSNs. Before you configure this datastore, configure a DSN for the Simba ODBC driver for Google BigQuery using the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.

Option	Description

OAuth Mechanism	Specify how the ODBC driver authenticates access to Google BigQuery. Required when *Use Data Source (ODBC)* is set to **No**. Select one of the following options:
	• *User Authentication*: Authenticates as a user through a Google user account. Also populate *Refresh Token*.
	• *Service Authentication*: Authenticates as a service through a Google service account. Also complete the following options:
	  • *Email*
	  • *Key File Path*
	i Note
	Appears only for DSN-less connections. For DSN connections, you select the OAuth mechanism and complete the additional options in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.

Refresh Token	Enter the refresh token obtained from your Google BigQuery account. Required when *OAuth Mechanism* is set to **`User Authentication`**. **i Note** Appears only for DSN-less connections. For DSN connections, you enter the Refresh Token in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.
Email	Enter your Google Cloud Platform e-mail ID accessed from your Google BigQuery account. Required when *OAuth Mechanism* is set to **`Service Authentication`**. **i Note** Appears only for DSN-less connections. For DSN connections, you enter Email in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.

Option	Description
Key File Path	Browse to and select the location of the P12 or JSON file you generated from Google Cloud Platform and saved locally. Required when *OAuth Mechanism* is set to **`Service Authentication`**. **i Note** Appears only for DSN-less connections. For DSN connections, you enter the Private Key information in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.
Catalog	Enter the Google BigQuery project name. Required when *Use Data Source (ODBC)* is set to **`No`**. **i Note** Appears only for DSN-less connections. For DSN connections, you enter the Catalog in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.
Advanced group	
Use SSL encryption	Configurable when *Use Data Source (ODBC)* is set to **`No`**. **i Note** Applicable only for DSN-less connections. For DSN connections, you select TLS by completing the Trust Store information in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.
Encryption parameters	Configurable when *Use Data Source (ODBC)* is set to **`No`**. Click in the text box to open the *Encryption Parameters* popup dialog box. Complete one of the following two options: • *Use System Trust Store* • *Trusted Certificate* **i Note** Applicable only for DSN-less connections. For DSN connections, you enter the Trust Store information in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.
Use System Trust Store	Select to use the system trust store instead of the Google BigQuery trusted certificate.

Option	Description
Trusted Certificate	Select the location for the Google BigQuery trusted certificate PEM file from the *Browse* dialog box, or you can enter the location for your PEM trust store file.
Proxy host *Proxy port* *Proxy user name* *Proxy password*	Optional. Complete the proxy options when you use a proxy server. **i Note** Applicable only for DSN-less connections. For DSN connections, you enter the Proxy options in the ODBC Data Source Administrator for Windows or the DS Connection Manager for Linux.

Related Information

Google BigQuery ODBC datastore [page 56]
Download and install the Simba ODBC driver [page 57]
Configuring DSN on Windows [page 57]
Configuring DSN on Linux [page 59]
Configuring driver for DSN-less connection on Linux [page 62]
Use Google BigQuery ODBC datastore as a source [page 64]

3.3.4.1 Google BigQuery ODBC datastore

With a Google BigQuery ODBC datastore, make native ODBC calls to your Google BigQuery data sets to download and process data in SAP Cloud Integration for data services.

After you create the datastore, open the datastore to view data from your Google BigQuery account. Download table metadata from your Google BigQuery account to use as a source in SAP Cloud Integration for data services.

i Note

SAP Cloud Integration for data services and Google BigQuery ODBC datastores do not support nested or repeated records. When a column is either a nested or repeated datatype, the column is not imported when importing tables and is ignored by SAP Cloud Integration for data services.

To access the data in your Google BigQuery account, the datastore uses the Magnitude Simba ODBC driver for BigQuery, which supports the OAuth 2.0 protocol for authentication and authorization. Configure the Magnitude Simba ODBC driver to provide your credentials and authenticate the connection to the data using either a Google user account or a Google service account.

3.3.4.2 Download and install the Simba ODBC driver

Download and install the Simba ODBC driver for Google BigQuery, and configure the driver based on your Windows or Linux platform.

Find driver downloads for the Magnitude Simba driver for BigQuery and access to documentation on the Google Cloud website .

Select the link **Windows 64-bit (msi)** or **Linux 32-bit and 64-bit (tar.gz)** to start the installation.

After you install the driver, follow the instructions to configure the driver for either a data source name (DSN) connection or a server name (DSN-less) connection. Then create the Google BigQuery ODBC datastore.

Be sure to add the following line to the DBClientDrivers scope within dsConfig, which is located in the %DS_COMMON_DIR%\conf folder:

```
GOOGLE_BIGQUERY = Simba ODBC Driver for Google BigQuery
```

3.3.4.3 Configuring DSN on Windows

A data source name (DSN) connection enables SAP Cloud Integration for data services to connect to a Google BigQuery named project and dataset.

Before you configure a DSN for Google BigQuery, download and install the Simba ODBC driver for Google BigQuery.

Perform the following steps to configure a DSN for Windows:

1. Click the Windows **Start** icon, then search for and open the **ODBC Data Source Administrator**.
2. Open either the **User DSN** or **System DSN** tab as applicable and click **Add**.
3. Select **Simba ODBC Driver for Google BigQuery** and click **Finish**.

   The ODBC Data Source Administrator opens the **Simba ODBC Driver for Google BigQuery DSN Setup** dialog box.
4. Enter a unique name in **Data Source Name** and optionally enter text for **Description**.
5. Select the applicable authentication from the **OAuth Mechanism** dropdown list: **Service Authentication** or **User Authentication**.

   The type of OAuth mechanism you select determines the authentication options to complete. Use the information in the following tables for option descriptions based on the authentication that you select.

   Table 4: User Authentication

Option	Description
User Authentication	ODBC driver authenticates as a user through a Google user account.
Sign In	Opens a sign-in dialog for Google BigQuery. Sign into your Google BigQuery account to obtain a confirmation code.

Option	Description
Confirmation Code	Code that you obtain from Google when you sign in. SAP Cloud Integration for data services uses the code to generate a refresh token. **i Note** You can use the confirmation code once. Obtain a new confirmation code when you need another refresh token. However, when you save the refresh token in the DSN configuration, the driver can use the same refresh token each time you use this DSN to access the account.
Refresh Token	Google BigQuery requires a token to access a user account. The driver uses the refresh token each time it accesses your Google user account. After you paste the code into **Confirmation Code**, perform one of the following actions: • Click the **Refresh Token** text box. • Click `Tab` to move your cursor from **Confirmation Code** to **Refresh Token**. The driver automatically populates the field with a refresh token. **i Note** In place of a refresh token, you can choose to save the token to a `.json` key file and save the file locally. For complete information about using a `.json` key file instead of a refresh token, see the Simba ODBC driver documentation.

Table 5: Service Authentication

Option	Description
Service Authentication	ODBC driver authenticates as a service through a Google service account.
Email	Obtain your service account e-mail ID from your Google account.
Key File Path	Enter the path and file name of the saved key file. Generate a Google service account private key from your Google account and save the file locally as either a JSON or P12 file type.

6. Select the version number from the **Minimum TLS Version** dropdown list.

   Google BigQuery requires TLS. SAP Cloud Integration for data services supports only TLS version 1.2.

7. Specify the Trust Store CA certificate file to use.
   - To use the Windows Trust Store for the CA Certificates, select **Use System Trust Store**.
   - To use the `.pem` file that is installed with the Simba ODBC driver for Google BigQUery, accept the default address in **Trusted Certificates**.
   - To use your system Trust Store, enter the full path to the trusted certificates `.pem` file on your system.
8. Select the applicable Google BigQuery project name from the **Catalog (Project)** dropdown list.
9. Select the data set from the **Dataset** dropdown list.
10. Optional. If you use a proxy server connection, click **Proxy Options** and complete the options as applicable.
11. Optional. Click **Test**.
12. Click **OK** after the DSN tests successfully.

After you finish configuring the driver and creating a DSN, create a Google BigQuery ODBC datastore using the options for a DSN connection.

Related Information

Download and install the Simba ODBC driver [page 57]

3.3.4.4 Configuring DSN on Linux

The DSN configuration on Linux requires the same information as on Windows, but you use the DS Connection Manager utility for configuration.

Perform the following tasks before you configure the DSN for Linux:

- Download the Simba ODBC driver for Google BigQuery.
- Log in as the root user and run the installation file with the applicable command. For example, for SUSE Linux, run the following command:

  ```
  zypper install simbagooglebigquery-<Version>-<Release>.x86_64.rpm
  ```
- Either use the command line for DS Connection Manager or install the GTK+12 library to use a graphical user interface.

Perform the following steps to configure a DSN connection on Linux for Google BigQuery ODBC datastore:

1. Open a command prompt and enter a command to open DS Connection Manager.

   DS Connection Manager is located in `$LINK_DIR/bin` by default.

   ```
   $ $LINK_DIR/bin/DSConnectionManager.sh
   ```

   The Start Menu of the DS Connection Manager opens displaying the options as follows:

   ```
   ***********************************
   SAP Data Services Connection Manager
   ***********************************
   ------------------Start Menu-----------------
   Connection Manager is used to configure Data Sources or Drivers.
   ```

```
1:  Configure Data Sources
2:  Configure Drivers
q:  Quit Program
Select one command: '1'
```

2. Enter **'1'** to configure data sources and click `Enter`.

   The list of database types appears.

3. Enter the index number that corresponds to Google BigQuery for the prompt **Specify database index #**.

   The DS Connection Manager presents prompts related to Google BigQuery.

4. Complete the following prompts as described in the following table.

   Table 6: DSN Connection Prompts

Prompt	Description
Specify the DSN name from the list or add a new one	Enter a unique name for the data source name.
Specify the UNIX ODBC Lib Path	Enter the path of the Unix ODBC driver manager library files. The Unix ODBC driver manager library files are in `$USER_DIR/unixODBC-232/lib`.
Specify the Driver	Enter the path and name of the Simba ODBC Google BigQuery driver file. The driver file is in the location where you installed the driver.
Specify the Google BigQuery OAuth Mechanism [0:Service Authentication/1:User Authentication]	Enter the index number that corresponds to the applicable OAuth Mechanism. Complete the prompts related to the authentication type you chose.

   The following table contains the options to complete when you select service authentication.

   Table 7: Service Authentication

Prompt	Description
Specify the Google BigQuery Email	Type the service account e-mail ID.
Specify the Google BigQuery Private Key	Type the full path to the P12 or JSON key file that you generate and download from your Google project.

   The following table contains the options to complete when you select user authentication.

   Table 8: User Authentication

Prompt	Description
Specify the Google BigQuery Refresh Token	Google BigQuery requires a token to access a user account. The driver uses the refresh token each time it accesses your Google user account. For instructions to obtain an access token, see "Retrieving a Refresh Token" in the Simba documentation.

5. Continue entering information for the prompts described in the following table:

   Table 9: DSN Connection Prompts

Prompt	Description
Specify the Google BigQuery catalog	Enter the Gooogle BigQuery project name.

Prompt	Description
Specify the Google BigQuery Proxy option	Optional. Enter **1** to enable the options. Enter **0** to disable the options so they do not appear. When you enter 1, complete the following Proxy options: • Proxy Host • Proxy Port • Proxy Username • Proxy Password
Specify the Google BigQuery Trusted Certificates	Enter the location and file name for the Google BigQuery trusted certificate file. The trusted certificates are for the TLS protocol, which is required for a Google BigQuery connection. If you leave this option blank, SAP Cloud Integration for data services uses the default certificate file in the driver installation directory: `/lib/cacerts.pem`. The exact file path varies based on the version of the driver installed driver.

DS Connection Manager uses the information you just entered to test the connection. DS Connection Manager shows one of the following messages:

- **Test connection failed.**
- **Successfully added database source.**

6. Press `Enter` after a successful connection message.
7. Enter **'q'** to quit.

Example

The following is an example of the DS Connection Manager prompts for configuring a DSN for the Simba ODBC driver for Google BigQuery. The example shows options for the OAuth mechanism, Service Authentication.

```
*********************************
Configuration for Google BigQuery
*********************************
The ODBC ini file is $ODBCINI
Specify the DSN name from the list or add a new one:
<DSN_Name>
Specify the Unix ODBC Lib Path:
/odbc/unixODBC-232/lib
Specify the Driver:
/<SIMBA_GBQ_DRIVER_INSTALL_DIR>/SimbaBigQuery/simba/
googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
Specify the Google BigQuery Oauth Mechanism[0: Service Authentication/
1:User Authentication]: '0'
0
Specify the Google BigQuery Email:''
<gserviceaccount e-mail address>.com
Specify the Google BigQuery Private Key:''
/<SIMBA_GBQ_DRIVER_INSTALL_DIR>/SimbaBigQuery/simba/googlebigqueryodbc/key/
privatekey.p12
Specify the Google BigQuery Catalog:''
<Google project name>
```

```
Specify the Google BigQuery Proxy option[0:Disabled/1:Enabled]:'0'
1
Specify the Google BigQuery Proxy Host:''
<proxy_host_name>
Specify the Google BigQuery Proxy Port:''
<proxy_port>
Specify the Google BigQuery Proxy Username:''
<proxy_username>
Specify the Google BigQuery Proxy Password:''
<proxy_password>
Specify the Google BigQuery Trusted Certificates:''
/<SIMBA_GBQ_DRIVER_INSTALL_DIR>/SimbaBigQuery/simba/
googlebigqueryodbc/lib/64/cacerts.pem
Testing connection...
Successfully added database source.
Press Enter to go back to the Main Menu.
```

The DS Connection Manager adds the Simba ODBC driver for Google BigQuery and DSN information to the ODBC INI file in `$ODBCINI` and the driver information to the ODBC INI file in `$ODBCINST`.

After you complete the steps to configure the DSN on Linux using the DS Connection Manager, create a Google BigQuery ODBC datastore using the options for a DSN connection.

3.3.4.5 Configuring driver for DSN-less connection on Linux

Configure the Simba ODBC driver for Google BigQuery using the SAP Data Services (DS) Connection Manager when you use a server name (DSN-less) connection on Linux.

Perform the following tasks before you configure the driver for Linux:

- Download either the RPM file or the Tarball file for the driver as applicable for the bit size of your SAP Cloud Integration for data services application.
- Log in as the root user and run the installation file with the applicable command. For example, for SUSE Linux, run the following command:

  ```
  zypper install simbagooglebigquery-<Version>-<Release>.x86_64.rpm
  ```

- Either use the command line for DS Connection Manager or install the GTK+12 library to use a graphical user interface. For complete information about the Connection Manager and the GTK+12 library, see the *Data Services Administrator Guide*.

To add the driver information to `$ODBCINST`, perform the following steps.

1. Open a command prompt and open DS Connection Manager that is located by default in `$LINK_DIR/bin`.

   For example:

   ```
   $ $LINK_DIR/bin/DSConnectionManager.sh
   ```

   The Start Menu of the DS Connection Manager opens displaying the options as follows:

   ```
   ************************************
   SAP Data Services Connection Manager
   ************************************
   ------------------Start Menu-----------------
   ```

```
Connection Manager is used to configure Data Sources or Drivers.
1:  Configure Data Sources
2:  Configure Drivers
q:  Quit Program
Select one command: '1'
```

2. Enter **2** to configure the driver and click `Enter`.

   The list of database types appears.
3. Enter the index number that corresponds to Google BigQuery for the prompt **Specify database index**.
4. Enter the driver name for the prompt **Specify the Driver Name**.
5. Enter the location and file name for the Simba ODBC driver for the prompt **Specify the Driver**.
6. Enter the path for the ODBC Library for the prompt **Specify the Unix ODBC Lib Path**.
7. Select the index related to the OAuth mechanism for the prompt **Specify the Google BigQuery Oauth Mechanism[0:Service Authentication/1:User Authentication]**.
8. Complete the options related to the OAuth mechanism you chose.

   For Service Authentication:

   Table 10: Service Authentication

Prompt	Description
Specify the Google BigQuery Email	Type the service account e-mail ID.
Specify the Google BigQuery Private Key	Type the full path to the P12 or JSON key file that you generate and download from your Google project.

   For User Authentication:

   Table 11: User Authentication

Prompt	Description
Specify the Google BigQuery Refresh Token	Google BigQuery requires a token to access a user account. The driver uses the refresh token each time it accesses your Google user account. For instructions to obtain an access token, see "Retrieving a Refresh Token" in the Simba documentation.

9. Enter the Google BigQuery project name for the prompt **Specify the Google BigQuery Catalog**.
10. Enter **1** to enable or **0** to disable for the prompt: **Specify the GoogleBigQuery Proxy option**.

    If you enter **1** for Enabled, enter Proxy information for the prompts.
11. Enter the location and file name for the Google BigQuery trusted certificate file for the prompt **Specify the Google BigQuery Trusted Certificates**.

    > **i** Note
    >
    > The trusted certificates are for the TLS protocol, which is required for a Google BigQuery connection. If you leave this option blank, SAP Cloud Integration for data services uses the default certificate file in the driver installation directory: `/lib/cacerts.pem`. The exact file path varies based on the version of the driver installed driver.

    DS Connection Manager uses the information you just entered to test the connection. DS Connection Manager shows one of the following messages:

- **Test connection failed.**
- **Successfully added database source.**

12. Press `Enter` after a successful test message.
13. Enter **'q'** to quit and close the DS Connection Manager.

> **Example**
>
> The following is an example of the DS Connection Manager prompts for configuring the Simba ODBC driver for Google BigQuery. The example shows options for a Service Authentication OAuth Mechanism. We've bolded the prompts for illustration purposes:

```
**********************************
Configuration for Google BigQuery
**********************************
The ODBC inst file is $ODBCINST
Specify the Driver Name:
GBQdriver
Specify the Driver:
/<SIMBA_GBQ_DRIVER_INSTALL_DIR>/SimbaBigQuery/simba/
googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
Specify the Unix ODBC Lib Path:
/odbc/unixODBC-232/lib
Specify the Google BigQuery Oauth Mechanism[0: Service Authentication/
1:User Authentication]: '0'
1
Specify the Google BigQuery Refresh Token:''
<refresh_token>
Specify the Google BigQuery Catalog:''
<GoogleProjectName>
Specify the Google BigQuery Proxy option[0:Disabled/1:Enabled]:'0'
1
Specify the Google BigQuery Proxy Host:''
<proxy_host_name>
Specify the Google BigQuery Proxy Port:''
<proxy_port>
Specify the Google BigQuery Proxy Username:''
<proxy_username>
Specify the Google BigQuery Proxy Password:''
<proxy_password>
Specify the Google BigQuery Trusted Certificates:''
/<SIMBA_GBQ_DRIVER_INSTALL_DIR>/SimbaBigQuery/simba/
googlebigqueryodbc/lib/64/cacerts.pem
Testing connection...
Successfully added driver.
Press Enter to go back to the Main Menu.
```

The DS Connection Manager adds the driver information to $ODBCINST.

Create a Google BigQuery ODBC datastore and complete the options that correspond with the DSN-less connection.

3.3.4.6 Use Google BigQuery ODBC datastore as a source

Use imported Google BigQuery tables as source objects in a data flow. When using a Google BigQuery datastore as a source, the target must be an SAP Integrated Business Planning (IBP) WebSocketRFC datastore.

To configure the Google BigQuery source table for SAP Cloud Integration for data services processing, create a data flow and click the source object to open the editor. The following information appears and is not editable:

- Table name
- Table owner
- Datastore name
- Database type

Set the editable source options as described in the following table as applicable.

Table 12: Google BigQuery Source tab options

Option	Description
Make port	Select to make the source table an embedded data flow port.
Join rank	Indicates the rank of this source relative to other tables joined in the data flow. SAP Cloud Integration for data services joins tables with higher join ranks before it joins tables with lower join ranks. Enter a non-negative integer. Default value is 0.
Cache	Indicates whether SAP Cloud Integration for data services reads data from the source and loads it into memory or pageable cache. • *Yes*: Always caches the source unless it is the outer-most source in a join. *Yes* is the default setting. • *No*: Never caches the source. → Tip Because SAP Cloud Integration for data services reads an inner table of a join for each row of an outer source, consider caching a source when you use it as an inner source in a join. The value for *Cache* specified in the Query transform editor *FROM* tab overrides any cache specified here in the source *Cache* option. Therefore, if your data flow contains a Query transform, specify the cache only in the Query transform editor.
Array fetch size	Indicates the number of rows retrieved from a source table in a single request. The default value is 1000 rows. The maximum is 5000 rows. Enter a higher number to reduce the number of requests to the source table, which decreases network traffic and improves performance.

Related Information

Conversion of Google BigQuery Datatypes [page 66]

3.3.4.6.1 Conversion of Google BigQuery Datatypes

When importing a table from Google BigQuery to SAP Cloud Integration for data services, the system replaces certain Google BigQuery datatypes with those compatible with the SAP Cloud Integration for data services environment, as shown in the following table:

Google BigQuery Datatype	SAP Cloud Integration for data services Datatype
BIGNUMERIC	decimal(77,38)
BOOLEAN	integer
BYTES	long(blob)
DATE	date
DATETIME	datetime
FLOAT	double
GEOGRAPHY	varchar
INTEGER	decimal(19,0)
JSON	varchar
NUMERIC	decimal(38,9)
TIME	time
TIMESTAMP	datetime

Struct datatypes are ignored and are not imported into SAP Cloud Integration for data services.

All other Google BigQuery datatypes are supported.

3.3.5 HANA

Create a HANA datastore to connect to a HANA database.

HANA datastores support a number of specific configurable options. Configure the datastore to match your HANA configuration.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.

Option	Possible Values	Description
Type	SAP HANA application cloud	Select the type of datastore to which you are connecting.
Application type	HANA	Specifies the application that should be used to access this datastore.
Instance	Alphanumeric characters and underscores	Name of the HANA application.

3.3.6 Microsoft SQL Server

Create a Microsoft SQL Server to connect to a Microsoft SQL Server database.

Microsoft SQL Server database datastores support a number of specific configurable options. Configure the datastore to match your Microsoft SQL Server database.

Prerequisites if using Azure Active Directory authentication:

- You must have installed SQL Server ODBC Driver 18 (Microsoft Windows) or DataDirect ODBC Driver V8.0 SP2 (Linux) on the Agent machine.
- You must have enabled TLS 1.2 or above on the Agent machine, which is enabled by default in several Microsoft Windows versions.

> ⚠ Caution
>
> If you are using Azure PaaS with agents that are older than the 2309 release, be aware that running a job uses the authentication method `SQL Server Authentication` despite your being able to select **`Active Directory – Password`** in *Authentication Method* for database subtype Azure PaaS. Pre-2309 agents do not recognize the new UI parameter *Authentication Method*. Since the user credentials are different, the job will fail with an error about incorrect credentials.

Option	Possible values	Description
Database Subtype	On Premise Azure PaaS Azure VM	Select the database subtype, which determines certain configuration options such as authentication method.
SQL Server version	Microsoft SQL Server `<version number>`	The version of your SQL Server client. This is the version of SQL Server that this datastore accesses.
Database server name	Computer name, fully qualified domain name, or IP address	The name of the host system where the SQL Server instance is located.
Database name	Refer to the requirements of your database	The name of the database to which the datastore connects.

Option	Possible values	Description
User name	Alphanumeric characters and underscores	The user name of the account through which SAP Cloud Integration for data services accesses the database. For SQL Server authentication, enter your database credentials. For Active Directory - Password authentication, enter your Azure credentials. For Windows Authentication, the *User name* field does not appear.
Password	Alphanumeric characters, underscores, and punctuation	The user's password. For SQL Server authentication, enter your database credentials. For Active Directory - Password authentication, enter your Azure credentials. For Windows authentication, the *Password* field does not appear.
Authentication Method	Windows Authentication SQL Server Authentication Active Directory - Password	The type of authentication used to connect to this datastore. For an On Premise or Azure VM database subtype, select *SQL Server Authentication* or *Windows Authentication*. For an Azure PaaS database subtype, select *SQL Server Authentication* or *Active Directory - Password*. **i Note** Be sure to enter the appropriate credentials as described above in *User name* and *Password*. For more information about how to use Windows authentication with Microsoft SQL Server, refer to Microsoft's SQL Server documentation.

Option	Possible values	Description
Use SSL encryption	Yes No	SSL encryption protects data that is transferred between the database server and the Agent. The default is *Yes*. For the On Premise and Azure VM database subtypes, you can choose to enable or disable SSL encryption. For the Azure PaaS database subtype, SSL encryption is automatically enabled and you cannot deselect the option. On Microsoft Azure, Microsoft SQL Server uses the default self-signing mechanism for secure connectivity. No additional action is necessary.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Aliases	-	Enter the alias name and the owner name to which the alias name maps.

For information about how to set up a Microsoft SQL Server Connection on Linux using a DataDirect driver for SAP Cloud Integration for data services Agent, see Knowledge Base Article 3202261.

3.3.7 MySQL

Create a MySQL datastore to connect to a MySQL database.

MySQL database datastores support a number of specific configurable options. Configure the datastore to match your MySQL Server database.

Option	Possible values	Description
MySQL Version	MySQL `<version number>`	The version of your MySQL client. This is the version of MySQL that the datastore accesses.
Use Data Source (ODBC)	Yes No	Select to use a DSN to connect to the database. By default, this option is set to *Yes*. To use a DSN connection, you must also specify the *ODBC data source name*. If you set this option to *No*, you must also specify the *Database server name*, *Database name*, and *Port number* for a DSN-less connection.

Option	Possible values	Description
ODBC data source name	Refer to the requirements of your database	The ODBC data source name (DSN) defined for connecting to your database. This option is required if *Use Data Source (ODBC)* is set to *Yes*.
Database server name	Refer to the requirements of your database	The MySQL database server name. This option is required if *Use Data Source (ODBC)* is set to *No*.
Database name	Refer to the requirements of your database	The name of the database defined in MySQL. This option is required if *Use Data Source (ODBC)* is set to *No*.
Port number	Integer	The number of the database port. This option is required if *Use Data Source (ODBC)* is set to *No*.
User name	Alphanumeric characters and underscores	The user name of the account through which the software accesses the database.
Password	Alphanumeric characters, underscores, and punctuation	The password of the account through which the software accesses the database.
Additional connection information	Alphanumeric characters and underscores or blank	Information for any additional parameters that the data source supports (parameters that the data source's ODBC driver and database support). Uses the format: `<parameter1=value1; parameter2=value2>`
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Date format	`yyyy.mm.dd` or other combinations	The date format supported by the data source.
Time format	`hh24:mi:ss` or other combinations	The time format supported by the data source.
Date-time format	`yyyy.mm.dd hh24:mi:ss` or other combinations	The date-time format supported by the data source.
Decimal separator	Period Comma	The character that the data source uses to separate the decimal portion of a number.

Option	Possible values	Description
Data type conversion support	No Automatic ODBC syntax SQL-92 syntax	If there is a data type mismatch in an expression, the software automatically generates an explicit convert function call.
NVL support	No Automatic ODBC syntax	If the input value is NULL, the software replaces with the specified value.
Ifthenelse support	Yes No	Allows conditional logic in mapping and selection operations.
Additional session parameters	A valid SQL statement or multiple SQL statements delimited by semicolons.	Additional session parameters specified as valid SQL statements.
Aliases	-	Enter the alias name and the owner name to which the alias name maps.

3.3.8 OData Adapter

An OData Adapter datastore can extract and load data using two types of authentication.

Authentication Options

You can use basic authentication or OAuth 2.0 authentication.

For basic authentication, create the datastore using the appropriate fields as described in OData Adapter Options [page 72].

For OAuth 2.0 authentication, do the following:

1. Register your client application to obtain a Client ID or API Key value and an X.509 certificate, both of which are used by the adapter for authentication. See Registering Your OAuth2 Client Application.
2. Create the datastore using the appropriate fields as described in OData Adapter Options [page 72].

Related Information

OData Adapter Options [page 72]
OData pagination [page 80]
Connecting to Secure Web Services by Manually Adding Certificates

3.3.8.1 OData Adapter Options

Create an OData Adapter datastore to connect to applications using OData services.

OData Adapter datastores support a number of specific options. Configure the datastore to match your adapter configuration.

Option	Possible values	Description
Endpoint URI	URI	The root endpoint URI for the OData data source.
Authentication Type	Basic OAuth 2.0	Specifies the authentication method to use when connecting to OData. • *Basic*: Uses Username and Password for authentication. • *OAuth 2.0* When you select *OAuth 2.0*, you need an endpoint token. The service uses the token to call the endpoint. For example, you would need a token from the Azure Active Directory (AD) v2.0 endpoint to call Microsoft Graph API v4 under its own identity. The following list outlines the basic steps to configure a service and obtain a token. This list uses Microsoft Graph API v4, which requires OData version V4, as an example. 1. Register your application in the Azure Portal. 2. Configure permissions for Microsoft Graph for your application. 3. Get administrator consent. 4. Get an access token. 5. Use the access token to call Microsoft Graph. **! Restriction** Perform steps 1 through 3 before configuring the datastore. To find specific instructions for the steps, go to the Microsoft Graph API Web page.
User Name	Alphanumeric characters and underscores	The user name of the account through which the software accesses the OData data source.
Password	Alphanumeric characters and underscores, or blank	The user's password. Applicable when you select one of the following: • *Basic* in *Authentication Type* • *Password* in *Grant Type* after selecting *OAuth 2.0* in *Authentication Type*

Option	Possible values	Description
Grant Type	SAML 2.0 Bearer Client credentials Password	Specifies the grant type. When **V2** is selected in *OData Version*, *SAML 2.0 Bearer* is selected by default and is greyed out so that you cannot change the selection. When either **V4** or **Auto** is selected in *OData Version*, you can change the grant type to one of the available options. For Microsoft Graph API, select *Client credentials*.
Client ID	Alphanumeric characters and dashes	Specifies the unique application (client) ID. Also known as an API Key value. For example, for Azure AD this ID is assigned when you click **Register** in the **Register an application** page in the Microsoft Azure portal. Applicable only when you select *OAuth 2.0* in *Authentication Type*.
Token Endpoint	URL	Specify the token endpoint to get the access token. For example, SAP Cloud Integration for data services uses the Azure AD v2.0 `/token` token endpoint to communicate with the Microsoft platform. Applicable only when you select *OAuth 2.0* in *Authentication Type*.
Client Secret	Alphanumeric characters	Specifies the password that the application uses to authenticate with the Microsoft identity platform. For example, you would obtain the client secret when you register your application on the Microsoft Azure Portal. Applicable only when you select *Client credentials* in *Grant Type*.
Company ID		Specifies a unique company ID that identifies the adapter client instance. Applicable only when you select *SAML 2.0 Bearer* in *Grant Type*.
Private Key PEM File	Path	Location where the agent can find the `<file_name>.pem` X.509 private key that the system uses to sign the SAML assertion. It can be the private key of a self-signed X.509 certificate or the private key of a generated X.509 certificate. Applicable only when you select *SAML 2.0 Bearer* in *Grant Type*.
Resource	URI	Specifies the URI of the Web API resource you want to access. This field is optional.

Option	Possible values	Description
Scope	URL	Specifies the scope (permissions) applicable for the request. For example, you would set permissions when you register your application on the Microsoft Azure Portal. The value passed for the scope parameter in this request consists of the following elements: • The application ID URI assigned when you registered the application • The default suffix `.default` **Example** For Microsoft Graph, the value is `https://graph.microsoft.com/.default`. This value requests tokens from the Azure AD v2.0 endpoint for the application resources for which you have permission. Applicable only when you select *OAuth 2.0* in *Authentication Type*.
Default Base64 binary field length	Integer	The default length for base64 binary fields, in kilobytes.
Depth	Integer	Specifies whether the OData data contains navigation properties. • *1*: OData data does not contain navigation properties. • *2*: OData data contains navigation properties. Read about navigation properties in your OData documentation.
OData Version	V2 V4 AUTO	Specifies the OData version. • *V2* • *V4* • *AUTO*: SAP Cloud Integration for data services detects the OData version based on the `Edmx Version` value obtained from the endpoint's metadata. If your endpoint defines the wrong version or contains an undefined version, you may see a connection error. **Note** SAP Cloud Integration for data services does not support job migration between OData V2 and V4 because each version uses different metadata. Also, SAP Cloud Integration for data services does not support OData V3. The OData adapter uses the Apache Olingo library that supports V2 and V4. For more information about OData libraries, see http://www.odata.org/libraries/ .

Option	Possible values	Description
URL Suffix	Alphanumeric characters	The URL suffix for OData endpoints, which routes requests to the correct client of the SAP ERP system. For example, `sap-client=001`. **⚠ Caution** Do not include a question mark (?). Do not include an ampersand (&) unless there are two or more query parameters. For example, `sap-client=001&authentication=Basic`. This parameter is optional if the endpoint does not require it. This field does not have a length limitation. Applicable only when you select *V4* or *Auto* in *OData Version*.
Require CSRF Header	no yes	Require the use of Cross-Site Request Forgery (CSRF) tokens. Default value is no.
OData Metadata Header	full minimal none	The OData.metadata parameter will be applied to the Accept header of an OData request to indicate how much control information the system includes in a response. Default value is Full. **⚠ Caution** For customers using OData V2, prior to agent version 2206 the *OData Metadata Header* option was set to the default of Full in the ATL although the header was not used. After upgrading to agent version 2206, in which the header is now supported for OData V2, customers using OData V2 should verify that the *OData Metadata Header* option in your datastores is set appropriately for your business needs. Also, if you call an OData V2 service in an SAP system, you must set OData Metadata Header in the OData datastore to **None** to avoid the SAP error "The server is refusing to process the request because the entity has an unsupported format.".

Data flow options

When you use an OData adapter datastore as a data flow source or target, there are additional options available. The following options are available in the Adapter Options tab in the data flow editor:

Table 13: OData as a Source

Option	Possible values	Description
Batch Size	Integer	The number of data rows to process as a single batch. Must be a value between 1 and 99999. Default: 1. Therefore, all new data flows are set to non-batch processing unless you change the batch size.
Column delimiter		The character sequence used to separate data between columns. Default: **/127**
Row delimiter		The character sequence used to separate data between rows. Default: **/007**
Null indicator		This indicates the value that is defined as a null value. Default: **/31**
Top Count	Integer	This is the standard $top OData option to limit the result set and only select the first N entries. **i Note** The top count does not support global variables.
Skip Count	Integer	This is the standard $skip OData option to skip the first N entries and only select entries starting from N+1. **i Note** The skip count does not support global variables.
As of Date	Date	This is a specific SuccessFactors OData option. The input date must be of the format **YYYY-MM-DD**.
From Date	Date	This is a specific SuccessFactors OData option. The input date must be of the format **YYYY-MM-DD**.
To Date	Date	This is a specific SuccessFactors OData option. The input date must be of the format **YYYY-MM-DD**.

Option	Possible values	Description
Number of concurrent threads	Integer	Controls the number of concurrent threads that SAP Cloud Integration for data services uses to load data. Default: **1**

Table 14: OData as a Target

Option	Possible values	Description
Batch Size	Integer	The number of data rows to process as a single batch. Must be a value between 1 and 99999. Default: 1. Therefore, all new data flows are set to non-batch processing unless you change the batch size.
Column delimiter		The character sequence used to separate data between columns. Default: **/127**
Row delimiter		The character sequence used to separate data between rows. Default: **/007**

Option	Possible values	Description
Loader Action		Specifies how SAP Cloud Integration for data services loads generated data to a target that contains existing data. • *Create*: Creates a new entity in the given target entity set. This functions in both batch mode and non-batch mode. **ℹ Note** If you load to Microsoft Graph API object, *Create* is the only option to select. • *Update*: Modifies an existing entity in the target using update semantics. This functions in both batch mode and non-batch mode. • *Merge*: Modifies an existing target entity using merge semantics. This functions in both batch mode and non-batch mode. • *Upsert*: Modifies an existing target entity and adds new entities if they do not already exist. **! Restriction** Because each OData adapter uses a different third-party API per OData version, there is not a method to send upsert requests to the OData service. Therefore, for the *Upsert* option, SAP Cloud Integration for data services uses the following workflow: • OData version 4: OData adapter sends an update request. If the update request fails, it creates and sends a request. • OData version 1 and 2: OData adapter sends a create request. If the create request fails, it sends a merge request. If the create request and the merge request fail to process, SAP Cloud Integration for data services generates an error message. • *Upsert function*: Modifies an existing entity in the target and adds new entities when the entity does not exist.

Option	Possible values	Description
		> **Note** For use with OData version 2 and SuccessFactors only. For SucessFactors, unlike the *Upsert* option, the *Upsert function* option sends the function by HTTP request to SuccessFactors. • *Upsert (IF-MATCH=*)*: For batch requests, modifies existing entities in the target and adds new entities when they do not already exist. *Batch Size* must be greater than 1. Supported in V2. > **Note** *Upsert (IF-MATCH=*)* is supported in release 2206 and higher. • *Delete*: Deletes an existing entity in the target. This functions in both batch mode and non-batch mode. • *Create link*: Creates a new related entity link between two entities in the target. • *Update link*: Updates related entity links between two entities in the target by navigation property. • *Delete link*: Deletes related entity links between two entities in the target by navigation property.

Option	Possible values	Description
Audit		Specifies whether to log data for auditing. • *True*: Logs the status for each row and creates audit files. Stores audit files in `<DS_COMMON_DIR>\log\LoaderAudit`. The format of the file is `<JobName>_<Datastore_Name>_<Table_Name>_<Process_ID>_<Thread_ID>.dat`. • *False*: Returns an error to the user interface if OData server throws an error. Does not check for the row statuses. **i Note** Selecting *False* may improve performance. Therefore, if you do not need auditing data, select *False*.

3.3.8.2 OData pagination

With an OData Adapter, SAP Cloud Integration for data services uses server-side pagination.

Server-side pagination utilizes the $skiptoken in the `odata.nextLink` annotation that comes as part of the response and indicates that a response is only a subset of the requested collection of entities or collection of entity references. It contains a URL that allows retrieving the next subset of the requested collection. The `nextlink` annotation will keep coming until there is next set of data and which will indicate stop requesting for more data.

SAP Cloud Integration for data services uses the *Batch size* value to determine how much data to send to the target at a time. A batch size from 2 through 99999 indicates batch processing.

Related Information

OData Adapter [page 71]

3.3.9 ODBC Data Sources

To work with ODBC data sources, drivers need to be configured on the Agent side.

For more information, see Configuring ODBC data sources in Linux in the SAP Data Services Agent Guide.

3.3.10 Oracle

Create an Oracle datastore to connect to an Oracle database.

Oracle database datastores support a number of specific configurable options. Configure the datastore to match your Oracle database.

Option	Possible values	Description
Oracle version	`<version number>`	The version of your Oracle client. This is the version of Oracle that this datastore accesses.
Use TNS name	Yes No	Whether to use TNS to connect to the database. By default, this option is set to *Yes*. To use a TNS connection, you must specify a *Database connection name*. If you set this option to *No*, you must also specify the *Database server name*, *SID*, and *Port number*for a TNS-less connection. If you select *Yes*,
Database connection name	Refer to the requirements of your database	An existing Oracle Transparent Network Substrate (TNS) name through which the software accesses sources and targets defined in this datastore. This option is required when you set *Use TNS name* to *Yes*.
Database server name	Computer name, fully qualified domain name, or IP address	The name of the host system where the Oracle Server instance is located. This option is required when you set *Use TNS name* to *No*.
System Identifier (SID)	Refer to the requirements of your database	The System ID for the Oracle database. This option is required when you set *Use TNS name* to *No*.
Port number	Integer	The port number used to connect to the Oracle Server. This option is required when you set *Use TNS name* to *No*.
User name	Alphanumeric characters and underscores	The user name of the account through which the software accesses the database.
Password	Alphanumeric characters, underscores, and punctuation	The user's password.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.

Option	Possible values	Description
Code page	-	Specify the character encoding of character data in the datastore.
Aliases	-	Enter the alias name and the owner name to which the alias name maps.
Default precision for Oracle Number	1 <= precision <= 96	The total number of digits in the value.
Default scale for Oracle Number	0 <= scale <= precision	The number of digits to the right of the decimal point.

3.3.11 REST Web Service

Create a REST Web Service datastore to connect to a REST Web Service.

REST Web Service datastores support a number of specific configurable options. Configure the datastore to match your REST-based web service.

SAP Cloud Integration for data services does not support using web services or RFC function calls as a source in the job's data flow. However, if you want to call one of them as a source, you have to set it up in the middle of a data flow. Also, you must set up a dummy source with any datastore because data flows require a defined source and target. You can choose any source you like, then use `Row_Gen` to trigger the data flow to iterate the row for function call. Additionally, you can use a web services datastore as a target.

Option	Possible values	Description
WADL Path	URL Local path	Specifies the location of the WADL file that describes the REST-based web service.
Display response in history	Yes No	Specifies whether to display the response from the web service in the *Web Service Response* tab in the history. The stored web service response will be cleared when the history is cleared.
Authorization schema	Basic Basic + CSRF Authorization Header OAuth 1.0 OAuth 2.0	Specifies the authorization schema to use when connecting to the web service. You must have knowledge of the provider to which you are connecting in order to obtain some configuration information, such as the consumer key. For OAuth 2.0-based authentication, the software supports only client_credentials and password-based grant types. It does not support the 3-legged model.
User name	Alphanumeric characters and underscores, or blank	The user name for basic authentication. This option is required only when basic authentication is needed to connect to the web service provider.

Option	Possible values	Description
Password	Alphanumeric characters and underscores, or blank	The password for basic authentication. This option is required only when basic authentication is needed to connect to the web service provider.
Password type	Plain Text Digest	The password type for basic authentication. This option is required only when basic authentication is needed to connect to the web service provider.
CSRF Fetch URL	URL	The URL of the CSRF token. This option is required when CSRF (Cross-Site Request Forgery protection) is needed to connect to the web service provider.
CSRF Fetch URL Method	GET POST	The preferred method to use to retrieve the CSRF token. This option is required when CSRF (Cross-Site Request Forgery protection) is needed to connect to the web service provider.
CSRF Header Key	Alphanumeric characters and underscores	The header key to use for CSRF protection. The default is X-CSRF-Token. This option is required when CSRF (Cross-Site Request Forgery protection) is needed to connect to the web service provider.
CSRF Header Value	Alphanumeric characters and underscores	The header value to use for CSRF protection. The default is Fetch. This option is required when CSRF (Cross-Site Request Forgery protection) is needed to connect to the web service provider.
Header-based API key or token	Alphanumeric characters and underscores	The API key or token to use for header-based authorization. This option is required only when an authorization header is needed to connect to the web service provider.
Consumer Key *Consumer Secret*	Alphanumeric characters and underscores	The OAuth 1.0 consumer key and secret (equivalent to a role account user name and password). You can obtain this information from the web service provider. This option is required when OAuth 1.0 authentication is needed to connect to the web service provider.

Option	Possible values	Description
Token Key *Token Secret*	Alphanumeric characters and underscores	The OAuth 1.0 token key and secret. This information allows single user authorization. You can obtain this information from the web service provider. These options are required only when OAuth 1.0 authentication is needed to connect to the web service provider and the provider does not support 2-legged authentication.
Request Token URL *Access Token URL*	URL	The URL for requesting a temporary authorization token and the URL for retrieving the final token. These options are required only when OAuth 1.0 authentication is needed to connect to the web service provider. Leave these options empty if the server uses only the consumer key and secret to access the protected resource.
Credentials Location	Both Header Body	This configuration option is available for OAuth 2.0 and allows you to choose where the authentication is added in the request. by selecting one of the following options: • Both (default) - Adds the client ID and client secret to both the authorization header and body of the request **i Note** Certain REST endpoints may only accept authentication in either the header or body, so selecting this option may cause an authentication failure. • Header - Adds the client ID and client secret to only the authorization header of the request • Body - Adds the client ID and client secret to only the authorization body of the request
Client ID *Client Secret*	Alphanumeric characters and underscores	The OAuth 2.0 client ID (represents your application) and client secret (security key). You can obtain this information from the web service provider. These options are required only when OAuth 2.0 authentication is needed to connect to the web service provider.

Option	Possible values	Description
Access Token	Alphanumeric characters and underscores	The location (API endpoint) of the OAuth 2.0 temporary token. This allows you to access protected resources. This option is required only when OAuth 2.0 authentication is needed to connect to the web service provider.
Refresh Token	Alphanumeric characters and underscores	The OAuth 2.0 refresh token. This option is required only when OAuth 2.0 authentication is needed to connect to the web service provider.
Grant Type	Client credentials Password	The type of grant access you want to use to obtain an access token. • Client credentials (default): Use your own credentials in order to obtain an access token. • Password: Use the resource owner's username and password to obtain an access token. This option is required only when OAuth 2.0 authentication is needed to connect to the web service provider.
Signature Method	HMAC-SHA1 Plain Text	The signature method to use for HTTP requests.
Preferred Method	Header String (POST) Query String (GET)	The method that you want to use to test trusted authentication.
Additional Headers	Alphanumeric characters and underscores	Allows you to include additional parameters in the webservices request. Enter one or more key/value pairs. Multiple parameters must be separated by an ampersand (&). For example: `resource=https://graph.facebook.com/oauth/access_token&scope=something` Supported only with OAuth 2.0 authorization schema.
XML recursion level	Positive integer	The number of passes the software should run through the XSD to resolve names. The default is 0.
Use proxy	Yes No	Specifies whether to use a proxy to connect to the web service provider.

Option	Possible values	Description
Standard HTTP Header Fields	A semi-colon separated list of header fields	A list of the fields and values that are the same and fixed for all web service functions in the web service datastore. The values for standard fields also remain the same for all web service calls in a data flow.
Dynamic Base URL	URL	The base URL comprised of the protocol, server name, port number, and path of the service that listens to RESTful web service requests. You can obtain this information from the WADL file under the element resource and attribute base. This is useful when you have multiple servers serving the same resources and you want to dynamically create the URL during the call. **Note** You must populate *Dynamic Base URL* if you are using more than one system configuration. Otherwise, the system connects to the server from which the WEB_SERVICE_FUNCTION was imported. Changing the default configuration does not affect the URL; you must add a Dynamic Base URL for this to work.
Preferred Mediatype	Application/XML Application/JSON	The preferred media type.
Server Certificate File	Path and filename	The path and filename of the .pem server certificate file on the Agent host system. Acquire the REST web services server certificate file from the REST web service provider and download it to this path. The path can be anywhere, however it must be configured on the Agent's allowlist.
Client Certificate File	Path and filename	The path and filename of the .pem client certificate file on the Agent host system. Contact your Security Administrator for this client certificate.
Client Key File	Path and filename	The path and filename of the .pem private key for the client certificate on the Agent host system.
Passphrase	Alphanumeric and special characters, or blank	The passphrase used to generate the private key file.

Data flow options

When you use a web services datastore as a data flow target, there are additional options available. The following options are available in the *Web Service Response* tab in the data flow editor:

Option	Possible values	Description
Response File Location	File path	The path to the template XML file on the SAP Data Services Agent host system where the response from the web service will be stored.
Delete and re-create file	Selected Unselected	Specifies whether to delete the existing response file each time the web service is called.

Related Information

Configuring Client Certificate Authentication for a REST Web Service Datastore [page 87]
Connecting to Secure Web Services by Manually Adding Certificates

3.3.11.1 Configuring Client Certificate Authentication for a REST Web Service Datastore

Administrators can configure client certificate authentication for REST Web Service datastores.

When creating a new REST Web Service datastore or editing the configuration of an existing REST Web Service datastore, perform the following steps:

1. In *Client Certificate File*, enter the path and filename of the .pem client certificate file on the Agent host system.
2. In *Client Key File*, enter the path and filename of the .pem private key for the client certificate.
3. In *Passphrase*, enter the passphrase used to generate the private key file.
4. Save your entries.

3.3.12 SAP Business Suite Applications

Create an SAP Business Suite Application datastore to connect to an SAP Business Suite Application.

Datastores for SAP Business Suite Applications support a number of specific options. Configure the datastore to match your SAP Application configuration.

Table 15: SAP Business Suite Applications options

Option	Possible values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.

Option	Possible values	Description
Type	SAP Business Suite Applications	Select the type of datastore to which you are connecting.
Agent	The list of agents that have been defined in the agents tab	Specifies the agent that should be used to access this data source.
Application server	Computer name, fully qualified domain name, or IP address	The name of the remote SAP application computer (host) to which the software connects.
Authentication	Password SNC	Specifies the authentication type used to connect to the datastore.
User name	Alphanumeric characters and underscores	The name of the account through which the software accesses the SAP application server.
Password	Alphanumeric characters and underscores, or blank	The user's password. **i Note** If you have problems connecting to a system that uses SAP_BASIS version 6.40 and earlier, the system might be expecting an uppercase password. To prevent this issue, install the appropriate kernel patch as described in SAP Note 792850, "Preparing ABAP systems to deal with incompatible passwords".
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.

Option	Possible values	Description
ABAP execution option	Generate and execute Execute preloaded	Select the task execution strategy. Your choice affects the required authorizations. *Generate and Execute:* The ABAP created by the task resides on the same computer as the SAP Data Services Agent and is submitted to SAP using the /BODS/RFC_ABAP_INSTALL_AND_RUN function. Select this option if the task changes between scheduled executions. This is the recommended option for non-production environments, such as sandbox or development. *Execute Preloaded:* ABAP resides on the SAP application server and is submitted using Data Services RFC function modules. Select this option if the task does not change between scheduled executions. This is the recommended option for production environments where the generated code from a non-production environment such as a sandbox or development environment has been reviewed and is uploaded to the production server.
Client number	000-999	The three-digit client number. Defaults to 800.
System number	00-99	The two-digit system number. Defaults to 00.
ODP Context	Refer to the requirements of the application	The context in the ODP framework describes a non-local SAP repository that maps its metadata in the ODP framework. The context can be compared with a schema in a database. When setting up SAP BW/4HANA as a source, set this option to *BW*.
Routing string	Refer to the requirements of the application	Enter the SAP routing string used to connect to SAP systems through SAProuters.
Execute in background (batch)	Yes No	Specify whether the generated ABAP programs created by SAP application data flows defined with this datastore will execute in batch mode on the SAP server. Batch mode operation is slower than the normal console mode; however, choose batch mode if the application is too long to run during the console mode time frame. Defaults to No.
Target host	Computer name, fully qualified domain name, or IP address	If you chose to execute ABAP programs in the background, specify the target computer (host).
Job class	A B C	If you chose to execute ABAP programs in the background, specify the job class.

Option	Possible values	Description
Security profile		By default, SAP Cloud Integration for data services does not use an SAP security profile. You can associate a security profile with a datastore so that data flows that access SAP application sources defined by the datastore include appropriate authorization checking. Specify any security profile defined in SAP (a predefined profile or a profile you defined).
RFC trace level	Brief Verbose Full	*Brief*: Error messages are written to the trace log. (Default) *Verbose*: The trace entries are dependent on the SAP program being traced. *Full*: In addition to entries traced by verbose value, data blocks are also traced. **i Note** You must specify a location on your Agent system where you want to store the RFC trace log file. To specify the location: 1. On your Agent system, navigate to the `conf` directory under `<DS_COMMON_DIR>`. 2. Open the `DSConfig.txt` file in a text editor. 3. In the `AL_Engine` section, after the line "`AL_EngineMiscOptions =`", add the following line: `SAP_RFC_TRACE_DIR = <rfc trace log directory>` Where <rfc trace log directory> is a directory on your agent system. For example, `SAP_RFC_TRACE_DIR = c:\temp`
RFC destination	SAPDS or `<Destination name>`	For the *RFC* data transfer method, enter a TCP/IP RFC destination. You can keep the default name of SAPDS and create a destination of the same name in the source SAP system, or you can enter a destination name for an existing destination.
Use sapnwrfc.ini	Yes No	Select to use an `sapnwrfc.ini` file, which overrides the datastore settings. Place the `sapnwrfc.ini` file in the current directory of the process being executed (`%LINK_DIR%/bin`). Defaults to No.

Option	Possible values	Description
Destination	Refer to the requirements of the application	If using an `sapnwrfc.ini` file, enter the destination name to reference.
Load balance	Yes No	Select *Yes* to enable load balancing, which helps to run tasks successfully in case the application server is down or inaccessible. See Extract Data from a Load-Balanced SAP Application System [page 93].
MS host	Computer name, fully qualified domain name, or IP address	Specify the message server host name. Overrides the setting in `sapnwrfc.ini`.
MS port	Refer to the requirements of the application	Specify this parameter only if the message server does not listen on the standard service `sapms<SysID>` or if this service is not defined in the services file and you need to specify the network port directly. Overrides the setting in `sapnwrfc.ini`.
Server group	`<User input>` Public Space	Optionally specify the group name of the application servers. Default: Public. Overrides the setting in `sapnwrfc.ini`.
System ID	Refer to the requirements of the application	Name of the SAP system. Overrides the setting in sapnwrfc.ini.
Upload attribute: Status	P - SAP Standard Production Program K - Customer Production Program S - System Program T - Test Program	Indicates whether the program is a test program, a system program, or a production program. Default is *T - Test program*. The parameter can have only the value code or the value code and description, separated by a space.
Upload attribute: Application	Refer to the drop-down list for available options	Indicates the application area to which the program belongs (Basis, General Ledger, Sales, and so on). The default value is *S - Basis*. The parameter can have only the value code or the value code and description, separated by a space.
Upload attribute: Development class (Package)	Refer to the drop-down list for available options	Indicates the name under which related objects in the ABAP Workbench are grouped together in a package. Default is $TMP. The program is created as a local (non-transportable) object.
Upload attribute: Request ID	Refer to the drop-down list for available options	Indicates the Change and Transport System (CTS) request ID. The default value is blank. This option is populated by the Data Services Agent if a non-local program object is created in SAP.
Upload attribute: Task ID	Refer to the drop-down list for available options	Indicates the CTS task ID. The default value is blank. This option is populated by the Data Services Agent if a non-local program object is created in SAP.

> **i Note**
>
> When creating a task and the source is either a Business Suite Application datastore or a BW Source datastore, you cannot use a BW Target datastore as the target.

Related Information

Extractor Options [page 92]
Extract Data from a Load-Balanced SAP Application System [page 93]
RFC-enabled Function Calls [page 94]

3.3.12.1 Extractor Options

You can maintain good system performance when extracting data during a transform from an SAP ODP source object by setting options on the *Extractor Options* tab. This can be a BW extractor or a generic extractor created by the IBP add-on in S/4HANA or SAP ECC.

The *Extractor Options* tab appears when you edit a data flow and then click on an input field.

Extractor options apply to any ODP source directly or through an embedded data flow.

The following table describes the extractor options:

Extractor option	Description
Package size	Indicates the maximum number of rows the extractor reads from the source and loads into memory at one time. Once the system processes and loads these rows to the target, it reads the next set of rows. By limiting the number of rows, less memory is used. Default is 1,000.
Initial load	For changed-data capture (delta loads), indicates whether to reload all the data on a subsequent run. • Yes: Returns a full extraction • No: Returns only data that has changed since the last run If you change the filter properties for an ODP source, reset the job by running it with the source option *Initial load* set to Yes. Then you perform subsequent runs with *Initial load* set to No.

Extractor option	Description
Extract from datetime	Indicates a specific date and time for when to extract changed data. Select a predefined global variable of type `datetime`. If the datetime value is the same as the value from the last execution, or falls before the value from the last execution, the system repeats the last changed-data extraction. If the datetime value is later than the value from the last execution, the system returns the new data. **Example** Yesterday the job ran with a datetime value of 2020.01.28 03:00:00, however there was a problem in the last execution. To reload the same data again, keep the datetime value the same. To get new changes, use a later value, for example 2020.01.29 03:00:00.
Parallel process threads	Specifies the number of threads used to read the data. For example, if you have four CPUs on your Agent machine, enter 4 to maximize performance. **Note** We recommend that you don't use this option. Setting a value can cause the software to go into recovery mode after the first iteration, resulting in sending the same rows repeatedly.

3.3.12.2 Extract Data from a Load-Balanced SAP Application System

Connect to and extract data from a load-balanced SAP application system.

Restriction

This concept applies only if you are using Data Services Agent version 1.0.11 Patch 34 or later.

To enable extracting from a load-balanced SAP application system, configure the SAP application datastore to connect to a load-balanced SAP application system and point it to the message server. Use an ABAP data flow in your SAP Cloud Integration for data services job.

SAP Cloud Integration for data services does not support failover. Therefore, if your message server goes down, your SAP Cloud Integration for data services job fails.

3.3.12.3 RFC-enabled Function Calls

Use RFC-enabled functions in SAP Cloud Integration for data services jobs to retrieve information from and apply information to SAP applications.

SAP Cloud Integration for data services supports select RFC-enabled function calls for SAP application datastores. RFC-enabled function calls can be used to read data from or load data to an SAP application datastore.

RFC functions can be called and used in query transformations. The transformation passes input values to the RFC functions and then produces the function return values as the output.

> **i Note**
>
> - RFC-enabled function calls can only be used as transforms and cannot be used as a target datastore.
> - RFC function parameters can be scalar or other types, such as exporting tables, without nested structures. All non-scalar parameters are shown as both input and output parameters.

RFC-enabled functions enable you to make up the input from tables. Specify the top-level table, top-level columns, and any tables nested one-level down relative to the tables listed in the FROM clause. If the RFC includes a structure as an input parameter, you must specify the individual columns that make up the structure.

RFC-enabled functions enable you to:

- Return a specific response based on specific input that you provide to the function
- Apply data to or retrieve data from more than one SAP table at a time

RFC functions can require input values for some parameters; SAP supplies default values for other inputs, and some can be left unspecified. You must determine the requirements of the function to prepare the appropriate inputs.

> **i Note**
>
> To avoid returning errors from RFC calls, format input as required by SAP. For example, all character data must be in uppercase; some values require padding to fill out the length of the data type.

A data flow may contain several steps that call a function, retrieve results, then shape the results into the columns and tables required for a response.

Related Information

Import and Use RFC-enabled Function Calls [page 95]
RFC-enabled Function Call Return Codes [page 95]

3.3.12.3.1 Import and Use RFC-enabled Function Calls

Import and use RFC-enabled function calls in SAP Cloud Integration for data services jobs to retrieve information from and apply information to SAP applications.

To import and use RFC-enabled function calls, follow these steps:

1. Navigate to the *Datastores* tab in the web UI of SAP Cloud Integration for data services.
2. Select an SAP application datastore from the list of datastores on the left-hand side.
3. Select the *Import Object By Name* icon under *Tables*.
4. Select **`Function`** in the *Type* drop-down list.
5. Enter the name of your RFC function in the *Name* field.
6. Click *OK*.

   You can now use your RFC-enabled function call in between query transformations by dragging and dropping the *Web Services or Function Call* transformation in the data flow editor.

3.3.12.3.2 RFC-enabled Function Call Return Codes

You can call SAP application RFC-enabled function calls, including Business Application Programming Interface (BAPI) functions, from queries inside data flows.

To make an RFC function available to call from SAP Cloud Integration for data services data flows, import the metadata for the function from the SAP application server using an SAP Applications datastore connection. Be aware that the requirements for RFCs and BAPIs, and therefore their metadata, may be different between versions of SAP applications.

If you design data flows with BAPI calls against one version of an SAP application, then change datastores to a later version of SAP, SAP Cloud Integration for data services allows this without the need to reimport the BAPI. Any new parameters added to the function call, including additional columns in table parameters, are added automatically to the call and filled with NULL values. Thus SAP Cloud Integration for data services allows you to design jobs that are portable between SAP systems.

For a SAP Cloud Integration for data services job to execute a RFC function, the login indicated by the datastore into which you imported the function must include the appropriate permissions required to execute the functions.

After you import the metadata for a SAP function, the function is listed in the Functions category of the SAP Applications datastore. You will also see the function in the function wizard listed under the datastore name.

SAP Cloud Integration for data services supports tables as input and output parameters for SAP RFC and BAPI functions. The function import process automatically includes the metadata for tables included as function parameters.

To specify a table as an input parameter to a function, the table must be an input to a query, either as a top-level input or nested under the top-level. The table must also be available in the FROM clause of the context where you call the function. SAP Cloud Integration for data services maps columns in the input schema by name to the columns in the table used as the function input parameter. You need only supply the columns that are required by the function. At validation, if SAP Cloud Integration for data services encounters type mismatches between supplied columns and the function signature, it attempts to convert the given type to the

expected type. For type mismatches that it cannot resolve, SAP Cloud Integration for data services produces validation errors.

One of the values that a transform can return is AL_RFC_RETCODE. This column contains a flag that identifies the success or failure of the function call. The possible values for AL_RFC_RETCODE are as follows:

Value	Description	Returned by
BOBJ_DI_RFC_OK	The RFC call succeeded. This value is replaced by the return value from the RFC call.	Data Services
BOBJ_DI_RFC_ENABLE_ERROR	The function is not RFC enabled.	Data Services
BOBJ_DI_RFC_NOT_ENABLED	The function is not RFC enabled.	Data Services
BOBJ_DI_RFC_CALLRECEIVE_ERROR	The function call in SAP returned an error.	Data Services
BOBJ_DI_RFC_R3_CONN_EXCEPTION	The SAP Applications datastore connection cannot be created, because of a connection error, invalid user, password, system number, or host name.	Data Services
BOBJ_DI_RFC_CALL_ERROR	The connection completes, but the call fails in SAP.	Data Services
BOBJ_DI_RFC_GET_RESULT_ERROR	Data Services cannot obtain the result of the function call from SAP.	Data Services
BOBJ_DI_RFC_COMMIT_ERROR	Data Services cannot commit the work because the BAPI_TRANSACTION_COMMIT call returned an error.	Data Services
BOBJ_DI_RFC_BAPI_NOT_SAME	The Data Services repository contains a different BAPI function than in SAP.	Data Services
RFC_OK	The function call succeeded. Look for the results or errors that it returns.	SAP application
RFC_FAILURE	The function call returned an error. If the function is a BAPI, details for the cause of the error are available in the RETURN structure available as an output from the function.	SAP application
RFC_EXCEPTION	The function call returned an error. If the function is a BAPI, details for the cause of the error are available in the RETURN structure available as an output from the function.	SAP application

Value	Description	Returned by
RFC_SYS_EXCEPTION	The function call returned an error and closed the connection to Data Services. If the function is a BAPI, details for the cause of the error are available in the RETURN structure available as an output from the function.	SAP application
RFC_CALL	The function call was received by SAP. If this value is left, the function failed to return a success flag after starting.	SAP application
RFC_INTERNAL_COM	An internal communication error occurred within SAP.	SAP application
RFC_CLOSED	The SAP application closed the connection and cancelled the function call.	SAP application
RFC_EXECUTED	The SAP application already executed the function call.	SAP application
RFC_MEMORY_INSUFFICIENT	The SAP application does not have enough memory available to process the function call.	SAP application
RFC_NO_TID	The transaction ID is not available to SAP.	SAP application
RFC_RETRY	The SAP application did not process data yet. SAP will retry the function call.	SAP application
RFC_NOT_FOUND	The SAP application could not find the function.	SAP application
RFC_CALL_NOT_SUPPORTED	The SAP application does not support the function call.	SAP application
RFC_NOT_OWNER	The login in the Data Services datastore cannot connect to SAP.	SAP application
RFC_NOT_INITIALIZED	The Data Services RFC library did not initialize properly.	SAP application
RFC_SYNCHRONIZE	The SAP application is busy processing a synchronous call.	SAP application
RFC_SYSTEM_CALLED	Data Services is busy executing a call from SAP.	SAP application
RFC_VERSION_MISMATCH	The version of the function call from Data Services is incompatible with the function expected by SAP.	SAP application

BAPIs are a type of RFC-enabled function calls. The RETURN structure for BAPIs varies between releases of SAP applications:

Field	Description
TYPE	blank — success S — success E — error W — warning I — information A — abort
CODE	Error message numbers
MESSAGE	Error message text in the language chosen at login

This TYPE value is blank or NULL depending on the current setting of the Server option *Convert SAP null to null*. Check this option by choosing *Tools* > *Options* in the Designer. In particular when calling BAPI functions, the data you provide through the BAPI call might be different from the data that you use to test a BAPI directly in the SAP GUI interface. The SAP application interface automates data handling, where the BAPI operation undercuts the interface level.

Consider the following issues:

- All character values must be uppercase
- Padding values
- Assumed decimal values (QTY)
- Codes are language-specific
- Automatic type conversion
- SAP application version-specific behavior

To determine the data requirements of various SAP application functions, you can read the function requirements in the SAP GUI transaction screens:

- BAPI list by functional area: `bapi`
- BAPI and RFC source and input and output parameters: `se37`

You can also determine appropriate values, such as the language-specific code values, by looking at the table where the data is ultimately stored.

3.3.13 SAP BW Source

Create an SAP BW Source datastore to connect to an SAP BW Source database.

SAP BW database datastores support a number of specific configurable options. Configure the datastore to match your SAP BW configuration.

Option	Possible values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	SAP applications	Select the type of datastore to which you are connecting.
Agent	The list of agents that have been defined in the agents tab	Specifies the agent that should be used to access this data source.
Application server	Computer name, fully qualified domain name, or IP address	The name of the remote SAP application computer (host) to which the software connects.
Authentication	Password SNC	Specifies the authentication type used to connect to the datastore.
User name	Alphanumeric characters and underscores	The name of the account through which the software accesses the SAP application server.
Password	Alphanumeric characters and underscores, or blank	The user's password. **Note** If you have problems connecting to a system that uses SAP_BASIS version 6.40 and earlier, the system might be expecting an uppercase password. To prevent this issue, install the appropriate kernel patch as described in SAP Note 792850, "Preparing ABAP systems to deal with incompatible passwords".
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.

Option	Possible values	Description
ABAP execution option	Generate and execute Execute preloaded	Select the job execution strategy. Your choice affects the required authorizations. *Generate and Execute:* The ABAP created by the job resides on the same computer as the SAP Data Services Agent and is submitted to SAP using the /BODS/RFC_ABAP_INSTALL_AND_RUN function. Select this option if the job changes between scheduled executions. → Tip This is the recommended option for sandbox or development systems. *Execute Preloaded:* ABAP resides on the SAP application server and is submitted using Data Services RFC function modules. Select this option if the job does not change between scheduled executions. → Tip This is the recommended option for production environments where the generated code from the sandbox has been reviewed and is uploaded to the production server.
Client number	000-999	The three-digit client number. Defaults to 800.
System number	00-99	The two-digit system number. Defaults to 00.
Routing string	Refer to the requirements of the application	Enter the SAP routing string used to connect to SAP systems through SAProuters.
Execute in background (batch)	Yes No	Specify whether the generated ABAP programs created by SAP application data flows defined with this datastore will execute in batch mode on the SAP server. Batch mode operation is slower than the normal console mode; however, choose batch mode if the application is too long to run during the console mode time frame. Defaults to No.
Target host	Computer name, fully qualified domain name, or IP address	If you chose to execute ABAP programs in the background, specify the target computer (host).
Job class	A B C	If you chose to execute ABAP programs in the background, specify the job class.

Option	Possible values	Description
Security profile		By default, SAP Cloud Integration for data services does not use an SAP security profile. You can associate a security profile with a datastore so that data flows that access SAP application sources defined by the datastore include appropriate authorization checking. Specify any security profile defined in SAP (a predefined profile or a profile you defined).
RFC destination	SAPDS or `<Destination name>`	For the *RFC* data transfer method, enter a TCP/IP RFC destination. You can keep the default name of SAPDS and create a destination of the same name in the source SAP system, or you can enter a destination name for an existing destination.
RFC trace level	Brief Verbose Full	*Brief*: Error messages are written to the trace log. (Default) *Verbose*: The trace entries are dependent on the SAP program being traced. *Full*: In addition to entries traced by verbose value, data blocks are also traced. **Note** You must specify a location on your Agent system where you want to store the RFC trace log file. To specify the location, do the following: 1. On your Agent system, navigate to the `conf` directory under `<DS_COMMON_DIR>`. 2. Open the `DSConfig.txt` file in a text editor. 3. In the `AL_Engine` section, after the line "`AL_EngineMiscOptions =`", add the following line: `SAP_RFC_TRACE_DIR = <rfc trace log directory>` Where <rfc trace log directory> is a directory on your agent system. For example, `SAP_RFC_TRACE_DIR = c:\temp`
Use sapnwrfc.ini	Yes No	Select to use an `sapnwrfc.ini` file, which overrides the datastore settings. Place the `sapnwrfc.ini` file in the current directory of the process being executed (`%LINK_DIR%/bin`). Defaults to No.

Option	Possible values	Description
Destination	Refer to the requirements of the application	If using an `sapnwrfc.ini` file, enter the destination name to reference.
Load balance	Yes No	Select *Yes* to enable load balancing, which helps to run tasks successfully in case the application server is down or inaccessible.
MS host	Computer name, fully qualified domain name, or IP address	Specify the message server host name. Overrides the setting in `sapnwrfc.ini`.
MS port	Refer to the requirements of the application	Specify this parameter only if the message server does not listen on the standard service `sapms<SysID>` or if this service is not defined in the services file and you need to specify the network port directly. Overrides the setting in `sapnwrfc.ini`.
Server group	`<User input>` Public Space	Optionally specify the group name of the application servers. Default: Public. Overrides the setting in `sapnwrfc.ini`.
System ID	Refer to the requirements of the application	Name of the SAP system. Overrides the setting in `sapnwrfc.ini`.
Upload attributes		
Status	P - SAP Standard Production Program K - Customer Production Program S - System Program T - Test Program	Indicates whether the program is a test program, a system program, or a production program. Default is *T - Test program*. The parameter can have only the value code or the value code and description, separated by a space.
Application	Refer to the drop-down list for available options	Indicates the application area to which the program belongs (Basis, General Ledger, Sales, and so on). The default value is *S - Basis*. The parameter can have only the value code or the value code and description, separated by a space.
Development class (Package)	Refer to the requirements of the application	Indicates the name under which related objects in the ABAP Workbench are grouped together in a package. Default is $TMP. The program is created as a local (non-transportable) object.
Request ID	Refer to the requirements of the application	Indicates the Change and Transport System (CTS) request ID. The default value is blank. This option is populated by the Data Services Agent if a non-local program object is created in SAP.
Task ID	Refer to the requirements of the application	Indicates the CTS task ID. The default value is blank. This option is populated by the Data Services Agent if a non-local program object is created in SAP.

> **i Note**
>
> When creating a task and the source is either a Business Suite Application datastore or a BW Source datastore, you cannot use a BW Target datastore as the target.

Related Information

Configuring SAP Business Suite Connectivity
Configuring SAP Business Suite connectivity

3.3.14 SAP BW Target

Create an SAP BW Target datastore to connect to an SAP BW Target database.

SAP BW database datastores support a number of specific configurable options. Configure the datastore to match your SAP BW configuration.

Option	Possible values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Agent	The list of agents that have been defined in the agents tab	Specifies the agent that should be used to access this data source.
Type	SAP BW Target	Select the type of datastore to which you are connecting.
Application server	Computer name, fully qualified domain name, or IP address	The name of the remote SAP application computer (host) to which the service connects.
Authentication	Password SNC	Specifies the authentication type used to connect to the datastore.
User name	Alphanumeric characters and underscores	The name of the account through which the service accesses the SAP application server.
Password	Alphanumeric characters and underscores, or blank	The user's password.
SNC library	Full file path and name of SNC security library	Enter the full path and name of the third-party security library to use for SNC communication (authentication, encryption, and signatures).
SNC name of Data Services	Refer to the requirements of the application	Enter the SNC name that the SAP system uses to identify .

Option	Possible values	Description
SNC name of SAP system	Refer to the requirements of the application	Enter the SNC name of the SAP system for this connection.
SNC quality of protection	Max Available Authentication Integrity Privacy	With *Max Available*, the system obtains the maximum quality of protection supported by the target SAP system. This value is configured in the SAP Application Server profile parameter snc/data_protection/max. It could be configured to be Authentication, Integrity, or Privacy. With *authentication*, the system verifies the identity of the communication partners, which in this case is SAP Cloud Integration for data services. This is the minimum protection level offered by SNC. No actual data protection is provided. With *integrity*, the system detects any changes or manipulation of the data that might have occurred between the two end points of a communication. With *privacy*, the system encrypts the messages being transferred to make eavesdropping useless. Privacy protection also includes integrity protection of the data. This is the maximum protection level offered by SNC.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Client number	000-999	The three-digit client number. Defaults to 800.
System number	00-99	The two-digit system number. Defaults to 00.
Routing string	Refer to the requirements of the application	Enter the SAP routing string used to connect to SAP systems through SAP routers.
RFC destination	SAPDS or `<Destination name>`	For the *RFC* data transfer method, enter a TCP/IP RFC destination. You can keep the default name of SAPDS and create a destination of the same name in the source SAP system, or you can enter a destination name for an existing destination.

Option	Possible values	Description
RFC trace level	Brief Verbose Full	*Brief*: Error messages are written to the trace log. (Default) *Verbose*: The trace entries are dependent on the SAP program being traced. *Full*: In addition to entries traced by verbose value, data blocks are also traced. **i Note** NOTE: You must specify a location on your Agent system where you want to store the RFC trace log file. To specify the location, do the following: 1. On your Agent system, navigate to the `conf` directory under `<DS_COMMON_DIR>`. 2. Open the `DSConfig.txt` file in a text editor. 3. In the `AL_Engine` section, after the line "`AL_EngineMiscOptions =`", add the following line: `SAP_RFC_TRACE_DIR = <rfc trace log directory>` Where <rfc trace log directory> is a directory on your agent system. For example, `SAP_RFC_TRACE_DIR = c:\temp`.
Load balance	Yes No	Select *Yes* to enable load balancing, which helps run tasks successfully in case the application server is down or inaccessible.
MS host	Computer name, fully qualified domain name, or IP address	Specify the message server host name. Overrides the setting in `sapnwrfc.ini`.
MS port	Must be a number that does not start with 0 (zero).	The port of the message server host name.
Server group	Public Space	Optionally specify the group name of the application servers. Default: Public. Overrides the setting in `sapnwrfc.ini`.
System ID	Refer to the requirements of the application	Name of the SAP system. Overrides the setting in `sapnwrfc.ini`.

To use the BW target datastore, you must configure the RFC destination with the Program ID defined. See the SAP Business Suite connectivity information in the *SAP Data Services Agent Guide*.

Data Flow Options

When you are setting up a data flow for a BW Target datastore, you can use the following options:

Option	Possible values	Description
Rows per commit	Positive integer Default: 1000	Enter the maximum number of rows loaded to a target table before saving the data. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Column comparison	*Compare by position* Default: *Compare by name*	Specifies how the service maps the input columns to persistent cache table columns. • *Compare by position*: Disregards the column names and maps source columns to target columns by position. • *Compare by name*: Maps source columns to target columns by name.

Option	Possible values	Description
Number of loaders	Positive integer Default: 1	Specifies the number of loaders the service uses. There are different types of loading: • Single loader loading: Loading with one loader. • Parallel loading: Loading with two or more loaders. When parallel loading, each loader receives the number of rows indicated in the *Rows per commit* option, and processes the rows in parallel with the other loaders. **Example** If *Rows per commit* = 1000 and *Number of Loaders* = 3: • First 1000 rows go to the first loader • Second 1000 rows go to the second loader • Third 1000 rows go to the third loader • Fourth 1000 rows go the first loader Each loader performs the necessary processing and, when complete, sends the data in a packet to the SAP BW system. Due to differences in processing, the loaders may not send the packets to the SAP BW system in sequential order. **Note** Parallel loading is not supported for a hierarchy BW data source.

> **i Note**
>
> When creating a task and the source is either a Business Suite Application datastore or a BW Source datastore, you cannot use a BW Target datastore as the target.

Related Information

Loading to a BW Target [page 108]
Configuring SAP Business Suite Connectivity
Set Up the Communication between BW and Agent
Route Strings
Set Up the Communication between BW and Agent

3.3.14.1 Loading to a BW Target

When loading to a BW target, you can load up to 5,000 records per info package, which is the default value.

3.3.15 SAP Cloud Platform (SCP) HANA

Create an SAP HANA application cloud datastore of application type HANA to connect to SAP Cloud Platform (SCP) HANA.

SCP HANA datastores support a number of specific configurable options. Configure the datastore to match your SCP HANA configuration.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	SAP HANA application cloud	Select the type of datastore to which you are connecting.
Application Type	HANA Cloud Platform HANA	Specifies the application that should be used to access this data source.
Account Name	Alphabets	HANA Cloud Application account name

Option	Possible Values	Description
Schema ID	Alphanumeric characters and underscores	HANA Cloud Application HANA Schema ID
DB User Name	Follow the database requirements	Optional. Username to activate the database that is exposed through SAP Cloud Platform
DB User Password	Follow the database requirements	Optional. Password to activate the database that is exposed through SAP Cloud Platform
Access Token	Alphanumeric characters	Specifies the Access Token that was generated when providing schema access for HCI-DSoD. This field is used to activate schema in the REST API call to Neo Persistency Service. Access Token field is not saved as a part of the application connection properties. See grand-schema-access.

3.3.16 SAP Datasphere

You can create an SAP Datasphere datastore to connect to an SAP Datasphere service.

Use the information in this table to set the appropriate options.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	Database	Select the type of datastore to which you are connecting, which is *Database* for an SAP Datasphere datastore.
Database	SAP HANA	The type of SAP HANA database
Agent	The agents that have been defined in the agents tab	Specifies the agent that should be used to access this data source.
HANA version	HANA 1.x or HANA 2.x	Select the version of the HANA datastore to which you are connecting.

Option	Possible Values	Description
Use Data Source (ODBC)	Yes No	Indicates whether to use a DSN to connect to the database. By default, this option is set to *Yes*. To use a DSN connection, you must also specify the *ODBC data source name*. If you set this option to *No*, you must also specify the *Database server name* and *Port number* for a DSN-less connection.
ODBC data source name	Refer to the requirements of your database	The ODBC data source name (DSN) defined for connecting to your database. This option is required if *Use Data Source (ODBC)* is set to *Yes*.
Database server name	Refer to the requirements of your database	The HANA database server name. This option is required if *Use Data Source (ODBC)* is set to *No*.
Port number	Integer	The number of the database port. This option is required if *Use Data Source (ODBC)* is set to *No*.
User name	Alphanumeric characters and underscores	The name of the account through which the software accesses the SAP application server.
Password	Alphanumeric characters, underscores, and punctuation	The password of the account through which the software accesses the database.
Additional connection information	Alphanumeric characters and underscores or blanks	Information for any additional parameters that the data source supports (parameters that the data source's ODBC driver and database support). Uses the format: `<parameter1=value1; parameter2=value2>`

Option	Possible Values	Description
Use SSL encryption	yes	SSL encryption protects data as it is transferred between the database server and the Agent. Make sure this is set to *Yes* for your SAP Datasphere datastore. Selecting *Yes* displays the *Validate Server Certificate* field.
Validate Server Certificate	no	Indicates whether to use server certificate authentication. For an SAP Datasphere datastore, set this option to *No*.
Use Client Certificate Authentication	yes no	Indicates whether to use client certificate authentication. The default is *No*. Setting this to *Yes* requires you to enter a certificate keystore and hides the *User name* and *Password* fields under *Credentials*. Setting this to *No* indicates authentication is done using the *User name* and *Password* fields under *Credentials*.
Certificate Keystore	Alphanumeric characters, underscores, and punctuation	Name of the certificate keystore PSE file that contains the client and/or server identities. This file is located either in `SECUDIR` or in a path you specify, which should be validated against your AllowedList. Required when *Validate Server Certificate* or *Use Client Certificate Authentication* is set to *Yes*.
Rows per commit	Positive integer Default: 1000	Enter the maximum number of rows loaded to a target table before saving the data. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.

Option	Possible Values	Description
Overflow file directory	Directory path	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.
Additional session parameters	A valid SQL statement or multiple SQL statements delimited by semicolon	A valid SQL statement or multiple SQL statements delimitated by semicolon.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Alias name	Alphanumeric characters and underscores	Enter the alias name. Required when loading/writing from HANA Cloud into an SAP Datasphere target using an IBP connection.
Owner name	Alphanumeric characters and underscores	Enter the owner name to which the alias name maps. Required when loading/writing from HANA Cloud into an SAP Datasphere target using an IBP connection.

3.3.17 SAP HANA Database

Create an SAP HANA Database datastore to connect to an SAP HANA database.

SAP HANA database datastores support a number of specific configurable options. Configure the datastore to match your SAP HANA configuration.

> **i Note**
>
> HANA modeling views such as attribute views, analytical views, and calculation views from a SAP Cloud Integration for data services (HANA schema) datastore, can be used as a data source.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	Database	Select the type of datastore to which you are connecting.

Option	Possible Values	Description
Database	SAP HANA	A type of SAP HANA Database
Agent	The agents that have been defined in the agents tab	Specifies the agent that should be used to access this data source.
HANA version	HANA 1.x or HANA 2.x	Select the version of the HANA datastore to which you are connecting.
Use Data Source (ODBC)	Yes No	Select to use a DSN to connect to the database. By default, this option is set to *Yes*. To use a DSN connection, you must also specify the *ODBC data source name*. If you set this option to *No*, you must also specify the *Database server name*, *Database name*, and *Port number* for a DSN-less connection.
ODBC data source name	Refer to the requirements of your database	The ODBC data source name (DSN) defined for connecting to your database. This option is required if *Use Data Source (ODBC)* is set to *Yes*.
Database server name	Refer to the requirements of your database	The HANA database server name. This option is required if *Use Data Source (ODBC)* is set to *No*.
Port number	Integer	The number of the database port. This option is required if *Use Data Source (ODBC)* is set to *No*.
User name	Alphanumeric characters and underscores	The name of the account through which the software accesses the SAP application server.
Password	Alphanumeric characters, underscores, and punctuation	The password of the account through which the software accesses the database.
Additional connection information	Alphanumeric characters and underscores or blanks	Information for any additional parameters that the data source supports (parameters that the data source's ODBC driver and database support). Uses the format: `<parameter1=value1; parameter2=value2>`

Option	Possible Values	Description
Use SSL encryption	yes no	SSL encryption protects data as it is transferred between the database server and the Agent. The default is *Yes*. Selecting *Yes* displays the *Validate Server Certificate* and *Server Certificate Hostname* fields
Validate Server Certificate	yes no	Indicates whether to use server certificate authentication. The default is *No*. Setting this to *Yes* requires you to provide a certificate keystore.
Server Certificate Hostname	Alphanumeric characters and underscores	Specifies the hostname used to verify server's identity. The host name specified here verifies the identity of the server instead of the host name with which the connection was established. For example, if a connection is established to the same host, it might be established to localhost instead of the actual host name in the certificate. **i Note** This parameter should be used only if you absolutely require it for your use case, such as in the example given above, since it bypasses the security of validating the established connection. In most cases, it would not be used.

Option	Possible Values	Description
Use Client Certificate Authentication	yes no	Indicates whether to use client certificate authentication. The default is *No*. Setting this to *Yes* requires you to enter a certificate keystore and hides the *User name* and *Password* fields under *Credentials*. Setting this to *No* indicates authentication is done using the *User name* and *Password* fields under *Credentials*.
Certificate Keystore	Alphanumeric characters, underscores, and punctuation	Name of the certificate keystore PSE file that contains the client and/or server identities. This file is located either in `SECUDIR` or in a path you specify, which should be validated against your AllowedList. Required when *Validate Server Certificate* or *Use Client Certificate Authentication* is set to *Yes*.
Rows per commit	Positive integer Default: 1000	Enter the maximum number of rows loaded to a target table before saving the data. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Overflow file directory	Directory path	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.
Additional session parameters	A valid SQL statement or multiple SQL statements delimited by semicolon	A valid SQL statement or multiple SQL statements delimitated by semicolon.

Related Information

Configuring X.509 Certificate Authentication for an SAP HANA Database Datastore [page 116]

3.3.17.1 Configuring X.509 Certificate Authentication for an SAP HANA Database Datastore

Administrators can configure X.509 certificate authentication for SAP HANA database source and target datastores.

Prerequisite: Agent version 2203 or higher

You can set up certificate authentication for all HANA database datastore types and for both ODBC- and server-based connections. See SAP HANA Database [page 112] for information about their options.

A datastore can have both client and server certificate authentication functioning simultaneously, or only one of them as needed.

Server Certificate Authentication

If ODBC is **not** used, follow these steps to set up server certificate authentication. If ODBC is used, all configuration is done in the HANA ODBC driver.

1. While creating or modifying an SAP HANA database datastore, set *Use SSL encryption* to **Yes**.
2. Set *Validate Server Certificate* to **Yes**.

   > **i** Note
   >
   > Enter a hostname only when the hostname in the certificate is different than the one from the connection. For example, when the connection is established to the localhost and the certificate contains the actual hostname. Populate this field only if a failure occurs that was caused by a known hostname change.

3. Enter the certificate keystore file name in *Certificate Keystore*.
4. Save your entries.

Client Certificate Authentication

To set up client certificate authentication, perform these steps:

1. While creating or modifying an SAP HANA database datastore, set *Use Client Certificate Authentication* to **Yes**.
   The user name and password in the *Credentials* section become hidden since authentication will be derived from the client certificate.
2. Do one of the following:
   - If *Use Data Source(ODBC)* is set to **Yes**, configure the keystore location in the ODBC driver on the client side.
   - If *Use Data Source(ODBC)* is set to **No**, enter the certificate keystore filename in *Certificate Keystore*.
3. Save your entries.

3.3.18 SAP Integrated Business Planning

Create an SAP HANA application cloud datastore of application type Integrated Business Planning to connect to SAP Integrated Business Planning.

Integrated Business Planning datastores support a number of specific configurable options. Configure the datastore to match your Integrated Business Planning configuration.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	SAP HANA application cloud	Select the type of datastore to which you are connecting.
Application type	Integrated Business Planning	Specifies the application that should be used to access this datastore.
Instance	Alphanumeric characters and underscores	Name of the Integrated Business Planning application.

3.3.19 SAP Integrated Business Planning via WebSocket RFC

To connect to an SAP IBP instance via WebSocket RFC, create an SAP Cloud Integration for data services datastore with the following options/parameters.

Starting with release 2209, when you use a WebSocket RFC connection you can create tasks and build data flows using IBP datastores as both your source and target. This functionality is supported only for WebSocket RFC connections. To take advantage of this IBP to IBP functionality, we strongly recommend that you migrate your connection type to WebSocket RFC if you have not done so already.

Option	Description
Type	Required: SAP HANA application cloud.
Application Type	Required. Specifies the application that should be used to access this datastore. For IBP WebSocket RFC, it is *Integrated Business Planning*.
Connection Type	Visible and required only when migrating from JDBC to WebSocket RFC. If you began using SAP Cloud Integration for data services directly with a WebSocket RFC connection, this field is not visible.
Instance	Required. This is the name of the specific SAP IBP instance that you want to connect to. The Operations team can provide “n” instances of IBP to a customer. Select the appropriate instance from the drop-down. Once selected, the instance's host name and port number information will display.
Agent	Required. The agent that all Integrated Business Planning interactions will use. For WebSocket RFC connections, the system displays agents compatible with Patch 39 or higher.

Option	Description

Authentication	Required. Indicates whether to authenticate by password or by certificate.
	• If you select *Password*, enter the following credentials:
	  • *User name*: (Required) The alias of the user name that IBP WebSocket RFC uses for the connection.
	  • *Password*: The password for the alias and username.
	• If you select *Certificate*, populate the PSE filename.
	For more information, see the SAP Integrated Business Planning for Supply Chain topic Defining the Communication Arrangement.

PSE filename	Required. The file name including the .pse extension of the Personal Security Environment (PSE) file, which contains the certificates for TLS communication. The file should always be on SECUDIR. For more information, see Setting Up a WebSocket RFC Connection.
TLS Trust All	Required. When enabled, the server certificate is not verified and all TLS entities are trusted. This option is mostly enabled for troubleshooting purposes and should not be enabled in production. Therefore, the recommended setting in production is *No*.
Host Name	Displays based on the instance selected above.
Port	Displays based on the instance selected above.
Batch Size	
Reader Batch Size (MB)	Size in megabytes of the batch used for reading data from IBP. Default size is 20MB.
Loader Batch Size (MB)	Size in megabytes of the batch used for loading data to IBP. Default size is 20MB.
Compression Type	Data compression method. Possible values are the following: • *LAN* - (Default) Uses fast LZ4 compression, which is the best option in fast networks. • *WAN* - Uses slow but better zLib compression, which may be better in slow networks. • *Off* - No data compression occurs, which is useful when troubleshooting problems because you can see the data in plain text in the RFC trace.
Proxy Settings	

Option	Description
Use Proxy	Required. Enable or disable proxy use. Possible values are *Yes* or *No*. When set to *Yes*, the proxy information is set up in the agent configuration. For more information, see Configuring the Secure Agent Connection.
Connection Settings	
Number of Connection Retries	The number of times to retry the connection before generating an error. Default is 1.
Interval between Retries (ms)	The time interval between two tries. For example, a connection retry or job status check. Default is 10000 milliseconds.
RFC Trace Level	The level of detail written to the RFC trace logs. Possible values are the following: • *Brief* - (Default) Error messages are written to the trace log. • *Verbose* - Includes additional statistical data compared to the Brief level; the volume of trace entries are dependent on the SAP program being traced. • *Full* - In addition to entries traced by Verbose, data blocks are also traced.

Related Information

Reimporting Objects for an SAP Integrated Business Planning Instance That Uses a WebSocket RFC Connection [page 119]

3.3.19.1 Reimporting Objects for an SAP Integrated Business Planning Instance That Uses a WebSocket RFC Connection

If you have an SAP Integrated Business Planning instance that uses a WebSocket RFC connection, the system alerts you if you attempt to reimport an object when its data structure has changed since the last import.

After you click *Import* on the Import Objects window, a dialog appears listing any objects that have undergone data structure changes, meaning that columns have been added or removed. You can choose whether to continue importing all listed objects or to cancel the import.

- If you cancel, you can then reselect which objects to import if, for example, you do not want to reimport the modified objects.
- If you continue with the import process, meaning you want to import the changed objects, you must manually update all tasks that use any of the listed objects. Follow these steps:

1. Select a task.
2. Open the task in edit mode.
3. Select a data flow.
4. Open the data flow in edit mode.
5. Double-click on any transform.
6. Click *Close*.
7. Click *Done*.
8. Repeat steps 3 through 7 for any additional data flows in the task.
9. Click *Done*.

Related Information

SAP Note 3276886

3.3.20 SAP Lumira Cloud

Create an SAP Lumira Cloud datastore to connect to an SAP Lumira Cloud database.

SAP Lumira Cloud datastores support a number of specific configurable options. Configure the datastore to match your SAP Lumira Cloud application configuration.

> **i Note**
>
> This datastore is only available on SAP Cloud Platform.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	SAP HANA application cloud	Select the type of datastore to which you are connecting.
Application type	SAP Lumira Cloud	Specifies the application that should be used to access this datastore.
Instance	Alphanumeric characters and underscores	Name of the SAP Lumira Cloud application.

Limitations:

- Tables can only be imported by browsing the schema and cannot be imported by name.
- View data is not available for tables.
- Lumira datastore can only be used as target in tasks or processes.

3.3.21 SOAP Web Service

Create a SOAP Web Service datastore to connect to a SOAP-based web service.

SOAP Web Service datastores support a number of specific configurable options. Configure the datastore to match your SOAP-based web service.

> **! Restriction**
>
> If you will connect to a SOAP web service that uses SSL, before you create the SOAP Web Service datastore, you must import the certificate and place the keystore on your agent machine to verify the client. These steps are necessary to enable two-factor authentication. See Importing Certificates in the *SAP Data Services Agent Guide* This applies only when using Data Services Agent version 1.0.11 patch 34 or later.

SAP Cloud Integration for data services does not support using web services or RFC function calls as a source in the job's data flow. However, if you want to call one of them as a source, you have to set it up in the middle of a data flow. Also, you must set up a dummy source with any datastore because data flows require a defined source and target. You can choose any source you like, then use `Row_Gen` to trigger the data flow to iterate the row for function call. Additionally, you can use a web services datastore as a target.

Option	Possible values	Description
WSDL Path	URL URI	Specifies the location of the external web service to accept a connection and return WSDL. When creating the datastore, the WSDL path must be accessible from the agent machine. If the WSDL path is entered incorrectly or is inaccessible for other reasons, the system will not create the datastore.
Display response in history	Yes No	Specifies whether to display the response from the web service in the *Web Service Response* tab in the history. **i Note** The stored web service response is cleared when the history is cleared.
User name	Alphanumeric characters and underscores, or blank	The user name for HTTP basic authentication. This option is required only when basic authentication is needed to connect to the web service provider.
Password	Alphanumeric characters and underscores, or blank	The password for HTTP basic authentication. This option is required only when basic authentication is needed to connect to the web service provider.

Option	Possible values	Description
WSS Username	Alphanumeric characters and underscores, or blank	The user name to use for WS-Security. This option is required only if the WS-Security communications protocol is needed to connect to the web service provider.
WSS Password	Alphanumeric characters and underscores, or blank	The password to use for WS-Security. This option is required only if the WS-Security communications protocol is needed to connect to the web service provider.
WSS Password Type	PlainText Digest	The password type to use for WS-Security. This option is required only if the WS-Security communications protocol is needed to connect to the web service provider.
WSS Time to live	Positive integer 0	The time for WS-Security protected messages to live. The default is 0. Any positive number will add a timestamp to the message. This option is required only if the WS-Security communications protocol is needed to connect to the web service provider.
WSS Policy file path	File path	The path to the WS-Security policy file on the SAP Data Services Agent host system. The default path is `<LINK_DIR>/ext/webservice-c/policy.xml`.
Socket timeout in milliseconds	Positive integer	The maximum number of milliseconds the web service client will wait to receive the response from the web service provider.
Axis2/c configuration file path	File path	The path to your Axis2/c configuration file (`axis2.xml`) on the SAP Data Services Agent host system. If a path is not specified, the default path is `<LINK_DIR>/ext/webservice-c/`.
Use proxy	Yes No	Specifies whether to use a proxy to connect to the web service provider.
XML recursion level	Positive integer	The number of passes the software should run through the XSD to resolve names. The default is 0.
SSL Pem File	Path and filename	The path and filename of the `.pem` file (private key or certificate) on the Agent host system.

Option	Possible values	Description
Keystore path	File path	The location of the keystore used to establish an SSL connection. This option is required only when client authentication is required for SSL connection. See Configuring Client Authentication for SOAP Web Services. **! Restriction** This option applies only when using Data Services Agent version 1.0.11 patch 34 or later.
Keystore password	Alphanumeric characters and underscores, or blank	The password of the keystore used to establish an SSL connection. This option is required only when client authentication is required for SSL connection. See Configuring Client Authentication for SOAP Web Services. **! Restriction** This option applies only when using Data Services Agent version 1.0.11 patch 34 or later.
Standard HTTP Header Fields	A semi-colon separated list of header fields	A list of the fields and values that are the same and fixed for all web service functions in the web service datastore. The values for standard fields also remain the same for all web service calls in a data flow.
Dynamic HTTP Header Fields	A semi-colon separated list of header fields	A list of the fields and maximum value lengths that may be different for each function in the web service datastore. The values for dynamic fields can also change for each web service call in a data flow.

Data flow options

When you use a web services datastore as a data flow target, there are additional options available. The following options are available in the *Web Service Response* tab in the data flow editor:

Option	Possible values	Description
Response File Location	File path	The path to the template XML file on the SAP Data Services Agent host system where the response from the web service will be stored.
Delete and re-create file	Selected Unselected	Specifies whether to delete the existing response file each time the web service is called.

Related Information

Connecting to Secure Web Services by Manually Adding Certificates
Connecting to secure web services

3.3.22 SuccessFactors Adapter

A SuccessFactors Adapter datastore can extract and load data to and from SuccessFactors using two types of authentication.

Authentication Options

You can use basic authentication or OAuth 2.0 authentication.

For basic authentication, create the datastore using the appropriate fields as described in SuccessFactors Adapter Options [page 124].

For OAuth 2.0 authentication, do the following:

1. Register your client application to obtain a Client ID or API Key value and an X.509 certificate, both of which are used by the adapter for authentication. See Registering Your OAuth2 Client Application.
2. Create the datastore using the appropriate fields as described in SuccessFactors Adapter Options [page 124].

Related Information

SuccessFactors Adapter Options [page 124]
Authentication Using OAuth 2.0
Configuring the SuccessFactors Adapter

3.3.22.1 SuccessFactors Adapter Options

Create a SuccessFactors Adapter datastore to connect to SuccessFactors.

SuccessFactors Adapter datastores support a number of specific options. Configure the datastore to match your adapter configuration. Be aware that some of the fields you must populate depend on which authentication type you select, as described in the following table.

Option	Possible values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	Adapter	Select the type of datastore to which you are connecting.
Adapter Type	SuccessFactors Adapter	Select the type of adapter you are using.
Agent	The list of agents that have been defined in the agents tab.	Specifies the agent used to access this data source.
Endpoint URI	URI	Specifies the URL where your service can be accessed by a client application.
Authentication Type	Basic OAuth 2.0	Specifies the authentication method to use when connecting to SuccessFactors. • *Basic*: Uses username and password for authentication. • *OAuth 2.0* A more secure way to authenticate without having to provide a password. When you select *OAuth 2.0*, you need an endpoint token. The service uses the token to call the endpoint. For more information, see Registering Your OAuth2 Client Application.
Company ID	Alphanumeric characters	Specifies a unique company ID that identifies the SuccessFactors client instance.
User Name	Alphanumeric characters and underscores	The user name of the account through which the software accesses SuccessFactors.
Password	Alphanumeric characters and underscores, or blank	The user's password. Applicable only when you select Basic as the authentication type.
Grant Type	SAML 2.0 Bearer	The credential used by the client to obtain an access token. Applicable only when you select OAuth 2.0 as the authentication type. Read-only.
Client ID	Alphanumeric characters and dashes	Specifies the unique application (client) ID. Obtained when you register your client application. Applicable only when you select OAuth 2.0 as the authentication type.
Token URL	URL	Used by the client to obtain an access token by presenting its authorization grant or refresh token. Applicable only when you select OAuth 2.0 as the authentication type.

Option	Possible values	Description
Private Key PEM File	Path	Location where the agent can find the `<file_name>.pem` X.509 private key that the system uses to sign the SAML assertion. It can be the private key of a self-signed X.509 certificate or the private key of an X.509 certificate generated by SAP SuccessFactors. Put the .pem file (obtained during app registration) in %DS_COMMON_DIR%\ext\SFSF\Certificate.pem. If you don't have this folder, create it and provide all the necessary permissions. You also have the option to use an alternate location. Be aware that no matter where you put the file, you need to specify the absolute path of the file in the datastore. Once you provide all of the parameters, the adapter authenticates using OAuth 2.0 and you receive an access token to log into SFAPI. Applicable only when you select OAuth 2.0 as the authentication type.
Default Base64 binary field length	Integer	The default length for base64 binary fields, in kilobytes.

Data Flow Options

When you use a SuccessFactors adapter datastore as a data flow source or target, there are additional options available. The following options are available in the Adapter Options tab in the data flow editor:

Option	Possible values	Description
Batch Size	Integer	The number of data rows to process as a single batch. Default: 200
Column delimiter		The character sequence used to separate data between columns. Default: **`/127`**
Row delimiter		The character sequence used to separate data between rows. Default: **`/007`**

Option	Possible values	Description
Constrain by		The character sequence used to build simple logical expression in “constrainable’ fields. The syntax used is similar to SFQL and it is used for ADHOC queries only. It covers: • As of date and Date Range fields • Only single level AND expressions • Represents required ad-hoc report parameters

3.3.23 Sybase ASE

Create a Sybase ASE datastore to connect to a Sybase ASE database.

Sybase ASE datastores support a number of specific configurable options. Configure the datastore to match your Sybase ASE configuration.

Option	Possible values	Description
Sybase version	`<version number>`	The version of your SAP ASE client. This is the version of SAP Sybase that this datastore accesses.
Database server name	Computer name	Enter the name of the computer where the SAP ASE instance is located. **i Note** For LINUX Agents, when logging in to a SAP Sybase repository in the UI, the case you type for the database server name must match the associated case in the SYBASE_Home\interfaces file. If the case does not match, you might receive an error because the Agent cannot communicate with the repository.
Database name	Refer to the requirements of your database	Enter the name of the database to which the datastore connects.
User name	Alphanumeric characters and underscores	Enter the user name of the account through which the software accesses the database.
Password	Alphanumeric characters, underscores, and punctuation	Enter the user's password.
Overflow file directory	Directory path	Enter the location of overflow files written by target tables in this datastore. A variable can also be used.
Rows per commit	Positive integer Default: 1000	Enter the maximum number of rows loaded to a target table before saving the data. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.

Option	Possible values	Description
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Server code page	-	Specify the character encoding of character data in the datastore.
Additional session parameters	A valid SQL statement or multiple SQL statements delimited by semicolon	A valid SQL statement or multiple SQL statements delimitated by semicolon.
Aliases	-	Enter the alias name and the owner name to which the alias name maps.

3.3.24 Sybase IQ

Create a Sybase IQ datastore to connect to a Sybase IQ database.

Sybase IQ datastores support a number of specific configurable options. Configure the datastore to match your Sybase IQ configuration.

Option	Possible values	Description
Sybase IQ version	Currently supported versions	Select the version of SAP Sybase IQ that this datastore accesses. Displayed options in the rest of the datastore editor vary depending on the version selected.
Use Data Source (ODBC)	Yes No	Select to use a DSN to connect to the database. By default, this option is set to *Yes*. To use a DSN connection, you must also specify the *ODBC data source name*. If you set this option to *No*, you must also specify the *Database server name*, *Database name*, and *Port number* for a DSN-less connection.
ODBC data source name	Refer to the requirements of your database	Type the data source name defined in the ODBC Administrator for connecting to your database. This option is required if *Use data source (ODBC)* is set to *Yes*.
Database server name	Computer name or IP address	Type the computer name or IP address. This option is required if *Use data source (ODBC)* is set to *No*.

Option	Possible values	Description
Database name	Refer to the requirements of your database	Type the name of the database defined in SAP Sybase IQ. This option is required if *Use data source (ODBC)* is set to *No*.
Port number	Integer	Type the number of the database port. This option is required if *Use data source (ODBC)* is set to *No*.
Server name	Refer to the requirements of your database	Type the SAP Sybase IQ database server name. This option is required if *Use data source (ODBC)* is set to *No*.
User name	Alphanumeric characters and underscores	Enter the user name of the account through which the software accesses the database.
Password	Alphanumeric characters, underscores, and punctuation	Enter the user's password.
Rows per commit	Positive integer Default: 1000	Enter the maximum number of rows loaded to a target table before saving the data. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Overflow file directory	Directory path	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Server code page	-	Specify the character encoding of character data in the datastore.

Option	Possible values	Description
Enable linked remote servers	Yes No	This option lets you use the INSERT...LOCATION SQL statement for a data flow that uses SAP Sybase IQ as the loader and SAP ASE or SAP Sybase IQ as the reader. The Data Services engine pushes down the SQL statement for the SAP Sybase IQ server location. Type Yes to use remote servers that have already been linked. To define a remote server, use the CREATE SERVER SQL statement in SAP Sybase IQ. To set up the remote login for users, use the CREATE EXTERNLOGIN SQL statement. For detailed information about the SQL statements, see the relevant SAP Sybase IQ product documentation.
Additional session parameters	A valid SQL statement or multiple SQL statements delimited by semicolon	Additional session parameters specified as valid SQL statement(s).
Aliases		Enter the alias name and the owner name to which the alias name maps.

3.3.25 Teradata

Create a Teradata datastore to connect to a Teradata database.

Teradata datastores support a number of specific configurable options. Configure the datastore to match your Teradata configuration.

Option	Possible values	Description
Teradata version	Teradata `<version number>`	The version of your Teradata client. This is the version of Teradata that the datastore accesses.
Use Data Source (ODBC)	Yes No	Select to use a DSN to connect to the database. By default, this option is set to *Yes*. To use a DSN connection, you must also specify the *ODBC data source name*. If you set this option to *No*, you must also specify the *Database server name*, *Database name*, and *Port number* for a DSN-less connection.
ODBC data source name	Refer to the requirements of your database	The ODBC data source name (DSN) defined for connecting to your database. This option is required if *Use Data Source (ODBC)* is set to *Yes*.

Option	Possible values	Description
Database server name	Refer to the requirements of your database	The Teradata database server name. This option is required if *Use Data Source (ODBC)* is set to *No*.
Database name	Refer to the requirements of your database	The name of the database defined in Teradata. This option is required if *Use Data Source (ODBC)* is set to *No*.
Port number	Integer Default: 8888	The number of the database port. This option is required if *Use Data Source (ODBC)* is set to *No*.
User name	Alphanumeric characters and underscores	The user name of the account through which the software accesses the database.
Password	Alphanumeric characters, underscores, and punctuation	The password of the account through which the software accesses the database.
Rows per commit	Positive integer Default: 1000	Enter the maximum number of rows loaded to a target table before saving the data. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Overflow file directory	Directory path	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.
Language	SAP-supported ISO three-letter language codes or <default>	Select the language from the possible values in the drop-down list. The <default> option sets the language to the system language of the SAP Data Services Agent host system.
Code page	-	Specify the character encoding of character data in the datastore.
Server code page	-	Specify the character encoding of character data in the datastore.
Log directory	Directory path	The directory in which to write log files.
Additional session parameters	A valid SQL statement or multiple SQL statements delimited by semicolon	A valid SQL statement or multiple SQL statements delimitated by semicolon.
Aliases	-	Enter the alias name and the owner name to which the alias name maps.

3.3.26 Workforce Analytics

Create a Workforce Analytics datastore to connect to a Workforce Analytics database.

Workforce Analytics datastores support a number of specific configurable options. Configure the datastore to match your Workforce Analytics configuration.

Option	Possible Values	Description
Name	Alphanumeric characters and underscores	The name of the object. This name appears in the datastores tab and in tasks that use the datastore.
Description	Any text	The description of the datastore.
Type	SAP HANA application cloud	Select the type of datastore to which you are connecting.
Application type	Workforce Analytics	Specifies the application that should be used to access this datastore.
Instance	Alphanumeric characters and underscores	Name of the Workforce Analytics application.

3.4 Import Metadata Objects

Importing metadata objects adds the table and file names from your source and target databases and applications to your datastores.

1. In the *Datastores* area, select a datastore.
2. Open the *Tables* or *File Format* tab (which one appears depends on the datastore type).
3. Do one of the following:
   - If the datastore has a *Tables* tab, click *Import Objects* or *Import Object by Name* and select the tables whose metadata you want to import. (To import a web service object, the web service must be up and running.)
   - If it has a *File Formats* tab, click *Create File Format* and select the option you want to create.

Task overview: Datastores [page 23]

Related Information

What are Datastores? [page 24]
Create Datastores [page 25]
Datastore Types and Their Properties [page 26]
View Data in a Datastore [page 133]
Create or Copy Datastore Configurations [page 134]
Create System Configurations [page 135]
Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]
What are File Formats? [page 34]
Reimporting Objects for an SAP Integrated Business Planning Instance That Uses a WebSocket RFC Connection [page 119]

3.5 View Data in a Datastore

After a task or process finishes running, you can view the data in its target datastore to ensure that the results are as you expected.

You can view data only in SAP HANA application cloud datastores that are in non-production environments. You cannot view data in source datastores or data in a production environment.

> **i Note**
>
> If you do not see the View Data icon in your target datastores, contact SAP Support and request that they activate View Data functionality on your target application. When you contact SAP Support, refer to the component **LOD-HCI-DS**.

1. In the *Datastores* tab, select the datastore that contains the data you want to view.
2. In the datastore's *Tables* tab, select a table.
3. Click *View Data* ().
4. (Optional) In the *View Data* window, define filter conditions to limit the number of rows that are displayed in the Data table.
   a. In the Filter pane, click the plus icon to add a filter condition.
   b. Select the column you want to filter on.
   c. Select the appropriate operator.
   d. Type a value.

      > **i Note**
      >
      > When filtering on a quoted string (varchar), you do not need to include the quotation marks in the *Value* field.

   e. (Optional) Add additional conditions.

      When defining additional conditions, ensure you select the appropriate operator (and/or) to use when evaluating the filter conditions.
5. (Optional) Select which columns you want to display. By default, the first 20 columns are displayed.

Task overview: Datastores [page 23]

Related Information

What are Datastores? [page 24]
Create Datastores [page 25]
Datastore Types and Their Properties [page 26]
Import Metadata Objects [page 132]
Create or Copy Datastore Configurations [page 134]
Create System Configurations [page 135]

Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]

3.6 Create or Copy Datastore Configurations

A datastore configuration represents a set of configurable options (including connection name, user name and password) and their values. A single datastore may have several different configurations, with each configuration used in a specific scenario or environment. For example, a datastore may have separate configurations for development and test environments.

> **! Restriction**
>
> Datastore configurations are not supported for file format groups.

If a datastore has more than one configuration, select a **default configuration**. The default configuration is always used for browsing and importing datastore objects. In cases where a system configuration has not been specified when scheduling or executing a task or process, the software uses the default datastore configuration.

You can create a new datastore configuration from scratch or copy an existing configuration and then modify it.

> **i Note**
>
> The copied configuration is identical to the original, except passwords are not copied.

1. In the *Datastores* tab, select an existing datastore and click *Configuration*.
2. In the datastore configuration toolbar, do one of the following:
   - Click the plus button () to create a new datastore configuration from scratch.
   - Click the copy button () to copy an existing configuration.
3. Specify a meaningful name for the new configuration and click *OK*.
4. Define (or modify) values for the appropriate options.
5. Click *Save*.

You can group datastore configurations from several different datastores into a system configuration.

Task overview: Datastores [page 23]

Related Information

What are Datastores? [page 24]
Create Datastores [page 25]
Datastore Types and Their Properties [page 26]
Import Metadata Objects [page 132]

View Data in a Datastore [page 133]
Create System Configurations [page 135]
Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]
Create System Configurations [page 135]

3.7 Create System Configurations

A system configuration is a set of datastore configurations that are used by a task or process during execution to connect to source and target datastores.

For example, within the Sandbox you want to execute a task or process using development systems and later using test systems. Using the appropriate datastore configurations, you could create a development system configuration and a test system configuration.

At least one datastore that has multiple datastore configurations.

1. In the Datastores tab, click *System Configurations*.
2. Click the *Create New System Configuration* button.
3. Use the buttons to add or remove datastore configurations from the system configuration.
4. Click *Save*.

When you run or schedule a task or process, use the *System Configuration* dropdown list to choose the configuration that contains the datastore configurations you want to use.

Task overview: Datastores [page 23]

Related Information

What are Datastores? [page 24]
Create Datastores [page 25]
Datastore Types and Their Properties [page 26]
Import Metadata Objects [page 132]
View Data in a Datastore [page 133]
Create or Copy Datastore Configurations [page 134]
Find Where an Object is Used [page 136]
Enable Secure Network Communications (SNC) in BW [page 136]
Create or Copy Datastore Configurations [page 134]

3.8 Find Where an Object is Used

A datastore cannot be deleted if its associated contents are in use. Find where an object is used by viewing its dependencies.

1. In the *Datastores* tab, select a specific datastore.

   The list of objects in the datastore displays in the right.
2. Select an object such as a table, file format, or web service function from the list.
3. Click the where used icon () to view the dependencies of the object.

Task overview: Datastores [page 23]

Related Information

What are Datastores? [page 24]
Create Datastores [page 25]
Datastore Types and Their Properties [page 26]
Import Metadata Objects [page 132]
View Data in a Datastore [page 133]
Create or Copy Datastore Configurations [page 134]
Create System Configurations [page 135]
Enable Secure Network Communications (SNC) in BW [page 136]

3.9 Enable Secure Network Communications (SNC) in BW

Enable SNC to provide a secure connection between SAP BW and the remote function call (RFC) server for jobs that you launch from SAP BW.

Prerequisites:

- Verify that SAP Cloud Integration for data services has the 64-bit SNC library installed.
- Download the SAPGUI_WIN32 package, which is the SAP Front End UI, if not installed already, to log on to the SAP system to perform tasks like importing the host certificate and exporting the server certification.

1. Open a command prompt as an administrator.
2. Execute `cd %link_dir%/bin`.
3. Generate the host certificate PSE by running the following command:

   ```
   sapgenpse.exe gen_pse -p PSE_name.pse -x PSE_password "CN=host_name, O=SAP, C=US"
   ```

   The distinguished name consists of the following case-sensitive elements:

- CN = <Common_Name>
- O = <Organization>
- C = <Country>

> **Example**
>
> ```
> sapgenpse.exe gen_pse -p hostname.pse -x abc1234 "CN=hostname, O=SAP, C=US"
> ```

Result: The PSE certificate is created under `ProgramData > SAP > DataServicesAgent > ssl > sec`.

4. On the same cmd as the previous step, create the login credential for the newly created PSE by running the following command:

   ```
   sapgenpse.exe seclogin –p PSE_name.pse -x PSE_password -o PSE_username
   ```

   Refer to the syntax definitions in step 3 [page 136].

   > **Example**
   >
   > ```
   > sapgenpse.exe seclogin -p hostname.pse -x "abc1234" -o XYZ6789
   > ```

   Result: The credential file cred_v2 is created under `ProgramData > SAP > DataServicesAgent > ssl > sec`.

5. On the same cmd as the previous step, export the host certificate by running the following command:

   ```
   sapgenpse.exe export_own_cert –o %ds_common_dir%\ssl\sec\PSE_name.crt -p
   PSE_name.pse -x PSE_password
   ```

   Refer to the syntax definitions in step 3.

   > **Example**
   >
   > ```
   > sapgenpse.exe export_own_cert –o %ds_common_dir%\ssl\sec\hostname.crt -p
   > hostname.pse -x XYZ6789
   > ```

6. In the SAP Logon application, update the BW/4HANA server with the agent host name certificate by doing the following:
   1. Select the BW/4HANA server or create a new entry for the server if necessary by performing the following steps:
      1. Select a connection type of *Custom Application Server*.
      2. Select *User Specified System* and click *Next*.
      3. Select *Custom Application Server*.
      4. Enter a description, the application server name, the instance number, and the system ID, then click *Finish*.
   2. Log on to the server by doing the following:
      1. Double-click the created connection.
      2. Enter the username and password.
   3. On the SAP Easy Access page, enter **STRUST** in all capital letters, then select *Enter* to access SAP Trust Manager.
   4. Locate and expand `SNC SAPCryptolib`, then click on the host server certificate beneath it.
   5. Click the *Display / Change*  icon in the upper left to go into Change mode.

6. Import the host `<PSE_name>.crt` certificate to the BW/4HANA server by doing the following:
    1. Click the *Import Certificate*  icon at the bottom of the window.
    2. Locate the host certificate .crt file in the directory to which you extracted it in step 5, then click *Open* and *Continue*.

        > **Note**
        >
        > Click *Allow* if you receive a security warning about file access.

    3. Click *Add to Certificate List* to add the imported certificate to the list of certificates.
    4. Click *Save*. The message "Certificate added to PSE" appears in the lower left of the window.
7. Export the BW/4HANA server certificate to update the host certificate by performing these steps:
    1. Double-click the *Subject* field.
    2. Click the *Export Certificate*  icon in the lower left of the window.

        > **Note**
        >
        > Confirm that the information you will export is related to the server certificate, not the PSE file you created.

    3. In *File path*, change the prepopulated file name, but be sure to maintain a .crt extension. This name cannot be the same as the one you just imported. Also, make this certificate name unique so you do not overwrite it if you export other certificates.

        > **Example**
        >
        > `BWServerB42Certificate.crt`

    4. In *File Format*, select *Base64*.
    5. Click the green *Confirm* checkmark. Click *Allow* if you receive a security warning about file access.
    Result: The .crt file is created under `ProgramData > SAP > DataServicesAgent > ssl > sec`.
    6. Click *Save*.
8. Point the host to the server by doing the following:
    1. In the SAP Logon application, enter transaction */nSU01*.
    2. Enter the username you use to log into your SAP system, then press *Enter*. This is not the user you use to log into the host machine.
    3. Click the *Display*  icon.
    4. Navigate to the *SNC* tab.
    5. Click the *Change*  icon in the upper left of the window.
    6. In the *SNC Name* field, insert the information you added when you created the certificate in the following format: `p:CN=<your CN>, O=<your O>, C=<your C>`.

        > **Example**
        >
        > In step 3 you executed `sapgenpse.exe gen_pse -p local_machine.pse -x password "CN=local_machine, O=SAP, C=US"`. Therefore, in SNC Name you would enter `p:CN=local_machine, O=SAP, C=US`.

    7. Make sure that *Allow password logon for SAP GUI (user-specific)* is selected.

9. Update the PSE with the server certificate by going back to the command prompt and in the folder `%link_dir%\bin` running the following command:

   ```
   sapgenpse.exe maintain_pk -a %ds_common_dir%\ssl\sec\server_certificate.crt -p
   PSE_name.pse -x PSE_password
   ```

   > **Example**
   >
   > ```
   > sapgenpse.exe maintain_pk -a %ds_common_dir%
   > \ssl\sec\BWServerB42Certificate.crt -p local_machine.pse -x password
   > ```

10. Go into the datastore and set up SNC authentication by doing the following:
    1. Select *SNC* as the authentication type.
    2. Provide the SNC library, the SNC name of Data Services, and the SNC name of the SAP system, as follows:
       - *SNC library*
         Enter the full path and name of the third-party security library to use for SNC communication (authentication, encryption, and signatures), which in a standard agent installation is `C:\Program Files\SAP\DataServicesAgent\bin\sapcrypto.dll`.
         You must add the folder C:\Program Files\SAP\DataServicesAgent\bin as a configured directory on your agent machine.
       - *SNC name of Data Services*
         This is the PSE of the certificate of the Agent. This is the information you entered in step 8.f.

         > **Example**
         >
         > p:CN=ccus1vmwin083, O=SAP, C=US

       - *SNC name of SAP system*
         This is the certificate of the appserver, which was created when IT installed the server. It must be in the following format: p:<subject>.

         > **Example**
         >
         > p:CN=B42, OU=SAP-BI, O=SAP, C=FR

         Gather this information as follows:
         1. In the SAP Logon application, log on to the server.
         2. At the command prompt of the SAP Logon application, type **STRUST** in all capital letters, then select *Enter* to access SAP Trust Manager.
         3. View the certificate list.
         4. For *SNC name of SAP system*, on the *Own Certificate* window click in the *Subject* field at the top, then copy the contents of the Subject field in the lower portion of the window. You populate the *SNC name of SAP system* field with this value.

Task overview: Datastores [page 23]

Related Information

What are Datastores? [page 24]

4 Tasks, Processes, and Projects

Tasks, Processes, and Projects allow you to define how data flows are put together and executed.

Related Information

4.1 What is a Task?

A task is a collection of one or more data flows that extract, transform, and load data to specific targets, and the connection and execution details that support those data flows. You can create tasks from scratch or from predefined templates.

Tasks contain the following information:

- Name, description, and project they belong to (*Details* tab).
- Source and target datastores to be used in the task's data flows (*Connections* tab).
- One or more data flows (*Data Flows* tab).
- Scripts and global variables applicable to all data flows in the task (*Execution Properties* tab).

Tasks must be created and tested before being promoted to production. Once in production, tasks can be run ad-hoc or on a schedule.

You can manage tasks from the *Projects* tab, where they are grouped under their parent project.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

Add Tasks to a Project [page 143]
Change Execution Order for Targets [page 143]
Moving Tasks (Export and Import) [page 144]
Scripts and Global Variables [page 145]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
What is a Project? [page 12]
What is a Data Flow? [page 164]

4.1.1 Add Tasks to a Project

There are multiple ways to add tasks to a project, such as importing, replicating, and creating from scratch or a predefined template.

Method	Procedure
Creating a new task	Select a project and click *Create Task*.
Importing a task	Moving Tasks (Export and Import) [page 144]
Replicating an existing task	Replicate a Task or Process [page 152]

4.1.2 Change Execution Order for Targets

When a task runs, its data flows are executed in the order in which their target objects appear in the data flows table. The data flows belonging to the target object at the top of the table are run first, and then those of the next target object in the table, and so on.

You can change the execution order of the data flows by reordering the target object in the data flows table.

> **i** Note
>
> If you want to execute data flows in parallel or to execute data flows from several tasks, consider using a process.

1. If the task is not already open for editing, from the *Projects* tag, select the task and click *Edit*.
2. In the *Data Flows* tab, select any target object and click *Actions* > *Manage target order*.
3. In the dialog, select a target object and use the arrow keys to move it.
4. When your target objects are in the desired order, click *Save*.

Related Information

What is a Process? [page 145]
Create a Process [page 147]

4.1.3 Moving Tasks (Export and Import)

You can move a single task or all tasks in a project by exporting and then importing to a different organization or new datacenter.

4.1.3.1 Export Tasks

You can export either a single task or all tasks in a project.

> **Note**
>
> When exporting an entire project, only the tasks are exported. Any processes that are part of the project are not exported.

1. Select the individual task or project containing the tasks you want to export.
2. Click *More Actions* > *Export*.

   A file is saved to your local Downloads directory. Single tasks are exported to a flat file in XMl format and saved with a `.xml` file extension. All tasks in a project are exported in a zip file.

4.1.3.2 Import Tasks

After exporting a single task or all the tasks in a project, complete the move by importing into a new organization or datacenter.

- You must have the Administrator role to import tasks.
- Tasks are imported into a project. Determine the project where you want to import the tasks. If needed, create a new project.
- File format group datastores must be created before the import process. The individual file formats are imported into the specified datastore.
  Any other required datastores are created during the import process. After importing, configure any new datastores with appropriate agent, connection, and logon credentials.

> **Note**
>
> Importing a task does not overwrite an existing datastore configuration.

1. Select the project where you want to import the single exported task or group of tasks in an exported project and click *More Actions* > *Import.*
2. Browse to the location where you saved the exported task or project.

   If you exported a single task, the file has a `.xml` extension.

   If you exported a project, the file has a `.zip` extension.
3. If you are importing file formats, specify the file format group datastore and select the *File Format Group* checkbox.

> **Note**
>
> You can safely ignore other fields in the dialog.

4. Click *OK*.

4.1.4 Scripts and Global Variables

Scripts and global variables can be used in tasks and processes.

Scripts and global variables are described in the following topics:

- Scripts [page 223]
  Scripts are used to call functions and assign values to variables in a task or process.
- Set Global Variables [page 235]
  Global variables are symbolic placeholders. When a task or process runs, these placeholders are populated with values.

4.2 What is a Process?

A process is an executable object that allows you to control the order in which your data is loaded.

A single process can include data flows from more than one task, project or datastore. Using the process editor, you can graphically specify the order in which you want the data to load and optimize the loading through parallel execution when data flows are independent of each other. When executing parallel data flows, SAP Cloud Integration for data services coordinates the parallel data flows, then waits for all data flows to complete before starting the next sequential step.

> **Note**
>
> In a process, SAP Cloud Integration for data services includes each data flow by reference; it does not make a separate copy. Changes that are made to a data flow (within its parent task) are automatically reflected in all processes that reference the data flow.
>
> Processes cannot be exported.

A process can include the following objects:

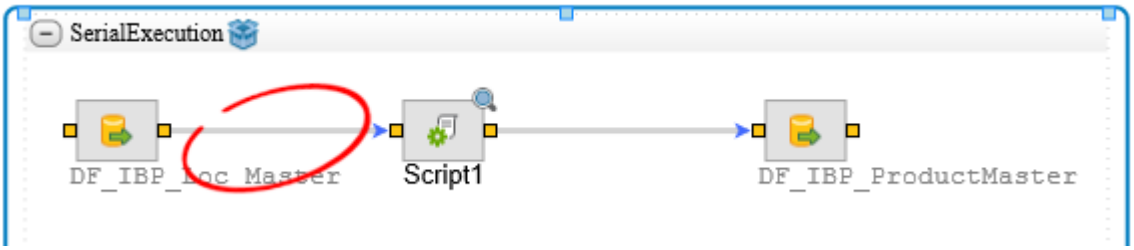
- data flows
- groups
- scripts
- annotations

Groups

Groups can contain data flows and scripts. Within a group, connections between objects are optional.

Independent data flows can be run in parallel to optimize loading efficiency. To be considered independent, data flows must not be required to run in a specific order nor rely on each other for any other reason. Data flows are run in parallel if they are contained in a group object, but not connected. This is illustrated in the following screenshot:

Data flows that must be executed in a specific order must be connected sequentially. It is optional to include sequential data flows in a group object, but you may choose to do so if that aids your data loading requirements. The data flow and script sequence in the following screenshot is executed sequentially because of the connections.



Scripts

A process can include scripts to call functions or assign values to global variables.

Scripts must be defined within a process. By design, scripts are not automatically referenced or copied from a data flow's parent task.

> → Tip
>
> You can copy a script from a task, paste it into a script object in a process, and then edit it as needed.

Global variables

Global variables are symbolic placeholders. When a task or process runs, these placeholders are populated with values. The values may be defined in the *Execution Properties* or set during an ad-hoc run.

When you drag and drop a data flow into a process, SAP Cloud Integration for data services also copies any global variables defined in the data flow's parent task. At the end of the design phase, the process execution properties include all the global variables that are defined in the parent tasks of all the data flows referenced in the process. Thus a process may include global variable definitions which are not used.

> **i Note**
>
> After a data flow has been referenced in a process, if the data flow is updated and new global variables are added to the parent task, the global variable list in the process is not automatically updated. To update the global variable list in the process editor, you must remove the data flow and then add it back.

Create a Process [page 147]
A process allows you to schedule data loads from multiple sources into multiple targets in an efficient and automated way. A process can reference data flows from tasks that are in different projects.

Process Design [page 149]
Thoughtful process design allows you to increase data loading efficiency.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
Scripts [page 223]
Set Global Variables [page 235]

4.2.1 Create a Process

A process allows you to schedule data loads from multiple sources into multiple targets in an efficient and automated way. A process can reference data flows from tasks that are in different projects.

Each data flow you plan to include in the process must be tested and work as expected within the context of its parent task.

Create a process, and then use the process editor to add data flows and include scripts.

1. Select the project to which you want to add the new process and click *Create Process*.
2. Enter a name for the process and, optionally, a description.
3. As needed for your situation, do one of the following:
   - If you are loading data to SAP Integrated Business Planning (IBP), ensure that the *Load to SAP Integrated Business Planning (requires post-processing)* box is checked (default) and select the target IBP datastore where you want to load your application data.
     Later, when you add a data flow to the process, the list of available target datatores includes only the selected target datastore and all datastores that are not of the type "Integrated Business Planning" such as file format group or web services.

     > **i Note**
     >
     > Within a process, you can load to only one Integrated Business Planning target datastore. This is due to post-processing actions which occur after the data is loaded.
     >
     > In addition to the IBP datastore, within the same process you can also load to target datastores which are not of the type Integrated Business Planning.

   - If you are loading data to any datastore other than Integrated Business Planning, deselect the *Load to SAP Integrated Business Planning (requires post-processing)* box.
4. Click *Create* > *Save and Edit Process*

   The process editor displays.
5. As planned in your process design, drag data flows, groups, and scripts from the tool palette and drop them on the canvas. Use connectors to indicate execution order.

Task overview: What is a Process? [page 145]

Related Information

Process Design [page 149]
Available Actions in Processes and Tasks [page 151]
Scripts [page 223]
Set Global Variables [page 235]

Add a data flow

1. Drag the data flow icon () from the object palette and drop it onto the canvas.
2. Select a target datastore.

   The result is a list of projects that contain tasks and data flows which load data to tables in the target datastore.
3. Expand the project and click the task which contains your desired data flow.

   A list of the tables and their associated data flows is displayed.

4. Select the desired data flow and click *OK*.

Add a group

Groups can contain data flows and scripts. Inside a group, connections between objects are optional.

1. Drag the group icon ( ) from the object palette and drop it onto the canvas.
2. Enter a name for the group.
3. Expand the group box by clicking on the + sign in the upper left corner.
4. Drag and drop script and/or data flow objects into the group as determined by your process design.
5. As needed, connect the objects.

   Data flows are executed in parallel if they are contained in a group object, but not connected.

Add a script

Use scripts to assign values to variables, call functions or define delta load properties.

1. Drag the script icon ( ) from the object palette and drop it onto the canvas.
2. Enter a name for the script.
3. Open the script editor by double-clicking the icon.
4. Type your script from scratch or copy an existing script from the data flow's parent task and paste it in the script editor.

   In the script, statements must end with a semicolon (;).
5. (Optional) Create any necessary global variables in the **Global Variables** area in the lower pane.
6. Click *OK* to save and close the script.

   The script is validated and a warning displays if there are any validation errors.

4.2.2 Process Design

Thoughtful process design allows you to increase data loading efficiency.

Planning

When designing a process, consider the following points:

- Review your data load strategy to identify areas where you can improve efficiency and reduce load time by loading data using a process instead of individual tasks.

- Note the names and locations (project and task) of the data flows you plan to include in the process.
- Before creating a process, make sure that the tasks and data flows you plan to refer to in the process load your data as expected. Consider the tasks and data flows as the foundation upon which a process is built.
- Understand and identify any dependencies between data flows. Does one data flow need to complete before the next begins? Can the data flows be included in a group and loaded in parallel?

Multiple datastore support

A process removes the single source and target datastore restriction that is imposed in tasks. Within a process, you can refer to data flows from more than one source datastore. You can also load data to targets in more than one target datastore.

> **!** Restriction
>
> Loading to more than one target application datastore is not supported for applications that require post-processing within the application after the data is loaded. These applications include:
>
> - SAP Integrated Business Planning
> - SuccessFactors Workforce Analytics
> - SAP Lumira

Process Promotion

Data flows cannot be promoted by themselves, only the parent tasks containing the data flows can be promoted. Since a process references the data flows (but does not make copies), SAP Cloud Integration for data services requires that the tasks containing the data flows referenced in a process be promoted before a process can be promoted. You can find the dependencies of a data flow by clicking the Where used icon ().

Additionally, it is possible for a data flow to be used in more than one process. Each process must be promoted individually. Ensure that you promote **all** processes that reference a data flow

Version support

SAP Cloud Integration for data services supports multiple versions of tasks and processes.

> ⚠ Caution
>
> After you roll back to a previous version of a task, it is recommended that you check all processes that reference the task's data flows to ensure that the references were maintained.

Parent topic: What is a Process? [page 145]

Related Information

Create a Process [page 147]

4.3 Available Actions in Processes and Tasks

Some actions are possible for both processes and tasks, but some actions are possible only for one or the other.

Action	Task	Process	Notes
Execute ad-hoc or scheduled	Yes	Yes	
Promote	Yes	Yes	Promote the tasks containing the data flows referenced in the process before promoting the process. The following icons may appear in the Promoted column on the *Projects* tab: ✓ The version of the task or process in this environment has been promoted to the next environment in the promotion path and the versions match. ⚠ The version of the task or process in this environment has been modified after being promoted and therefore does not match the version in the next environment in the promotion path. You must promote the modified task or process to the next environment for them to match.
Create or edit data flow	Yes	No	
Load content from more than one datastore	No	Yes	
Load content to more than one datastore	No	Yes	(Process) Each data flow can load content to a single datastore. A process can include multiple data flows and each data flow can load to a different datastore. **i Note** Loading into more than one application datastore is not supported for Integrated Business Planning, Workforce Analytics, and Lumira.
Define the execution order of data flows	Yes	Yes	(Task) Execution order can be defined only for data flows within a single task.
Preview data with the design-time data viewer	Yes	No	
Define global variables	Yes	Yes	

Action	Task	Process	Notes
Create scripts to call functions or assign values to global variables	Yes	Yes	
Group data flows so they execute in parallel	No	Yes	
Version	Yes	Yes	

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
Working in Multiple Environments [page 20]

4.4 Replicate a Task or Process

You can replicate an existing task or process to the same or different project.

To replicate a task or process, select the task in the Projects tab and choose *Replicate* from the *More Actions* menu.

When you replicate a task, copies of the task and all data flows that it contains are created and added to the target project you select as the replication target.

When you replicate a process, copies of the process (including references to data flows), scripts and execution properties are created and added to the target you select as the replication target.

The replicated task or process is named `<original_name>_copy_<#>`

> **Note**
>
> You cannot replicate tasks or processes in the Production environment.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
Duplicate a Data Flow [page 166]

4.5 Edit a Task or Process

Changes to a task or process are made in a Sandbox environment by administrators and developers and then promoted to the next environment in the promotion path. Note that you cannot edit tasks and processes directly in a Production environment.

To edit a task or process, select it in the *Projects* tab and click *Edit*. Make the necessary changes to the task, process, or data flow, then save your updates.

If a user in View mode moves among the tabs of a **task** while it is being edited, the system displays a message that the task may have changed. Closing the data flow and refreshing the list on the *Projects* tab shows the updated task.

If a user in View mode moves among the tabs of a **process** while it is being edited, the user sees the current version of the process, including the changes.

You may see the following icons in the Promoted column:

✓ The version of the task or process in this environment has been promoted to the next environment in the promotion path and the versions match.

⚠ The version of the task or process in this environment has been modified after being promoted and therefore does not match the version in the next environment in the promotion path. You must promote the modified task or process to the next environment for them to match.

Therefore, after editing a task or process, move the modified version to the next environment in your promotion path when you are ready by promoting it on the *Projects* tab. Promote the tasks within a process before promoting the process itself.

Keep in mind the following items when modifying a task or process:

- When you change the name of a task or process that has already been promoted, the name change is immediately sent to the next environment in your promotion path, even when there are other changes to that task or process that require promotion.
- A change to the description of a task or process is not flagged with the  icon. If you want the description in your environments to match, you should repromote the task or process.
- If your environment uses suborgs, you should make changes to tasks and processes in the highest org and promote the changes through your org structure. Making a change in an org that is midway through your org structure increases your risk of inconsistent behavior because the change would not appear in the higher level orgs.

Unlocking a Task or Process

If a task or process that you need to modify is currently being edited by another administrator or developer, it will appear locked. Administrators can choose *Unlock* from the *More Actions* menu and, after accepting the confirmation messages, can edit the task or process. Unlocking must be used with caution however, as users simultaneously saving changes can cause conflicts. Unlock a task or process only if you cannot unlock it another way and when you know that the other person editing the task or process will not save any changes.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
User Roles [page 392]
Working in Multiple Environments [page 20]

4.6 Promoting a Task or Process

Promotion is the application lifecycle management tool in SAP Cloud Integration for data services. It allows you to copy and move a task or process from one environment to the next available environment, for example, from Sandbox to Production.

The application lifecycle often involves multiple environments, with each environment used for a different development phase. SAP Cloud Integration for data services comes with two environments, Sandbox and Production.

- Use the Sandbox environment to create and edit objects.
- Once the design phase is done, promote tasks and processes to the Production environment where no further modification is allowed on the objects to be executed.

Only a user with the Administrator role can promote a task or process.

You can modify tasks and processes in Sandbox after they have been promoted. Most changes do not affect the already-promoted version in the Production environment until they are promoted; changing the name of a task or process, however, directly takes effect in the next environment in the promotion path.

You may see the following icons in the Promoted column:

 The version of the task or process in this environment has been promoted to the next environment in the promotion path and the versions match.

 The version of the task or process in this environment has been modified after being promoted and therefore does not match the version in the next environment in the promotion path. You must promote the modified task or process to the next environment for them to match.

Therefore, after editing a task or process, move the modified version to the next environment in your promotion path when you are ready by promoting it on the *Projects* tab. Promote the tasks within a process before promoting the process itself. For more information, see Edit a Task or Process [page 153].

If no projects exist in the Production environment when you promote a task or process from Sandbox to Production, the system creates a new project in Production called `Default` and places the promoted task or process into this project.

Datastore configurations

When a task or process is promoted from Sandbox to Production for the first time, its datastore configuration information is automatically carried over to the Production repository. The Administrator needs to edit and verify the datastore configuration information in the Production repository to make sure the datastore is pointing to the correct productive repository.

When a task or process is modified in the Sandbox environment, it may be promoted again. The changes that the Administrator has made in the Production datastore configurations will remain unchanged. The Sandbox datastore configuration information will not overwrite the configuration information and all defined objects in the Production repository. However, if needed, a user can *Include source datastore configurations* and *Include target datastore configurations* when re-promoting a task or process to overwrite the Production datastore configurations with the Sandbox datastore configurations.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]

4.7 Versioning Tasks and Processes

A new version is created each time you promote a task or process. You can also create a custom version if needed.

Versions allow you to keep track of major changes made to a task or process. You can consult the version history and return to a previously promoted or saved version to roll back unwanted or accidental changes.

It is recommended that you give each version a unique name and a meaningful description. They can remind you of the changes you made to the task or process, help you decide whether you want to roll back to a previous version, and decide which version you want to roll back to.

> ⚠ Caution
>
> After you roll back to a previous version of a task, it is recommended that you check all processes that reference the task's data flows to ensure that the references were maintained.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]

Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
Roll Back to a Previous Version [page 157]

4.8 Roll Back to a Previous Version

If you are not satisfied with the changes you have made to a task or process in your current environment such as Sandbox, you can roll back to a previous version of the task.

1. Select the task or process, and click *More Actions* > *Manage Versions*.
2. Select the version that you want to roll back to, and click *Rollback*.

   If you are not sure which version is the one that you want to go back to, you can refer to the version name and description, or use the *View* function to check more details.
3. Click *Yes*.

   The checkmark in the *Latest* column will switch to the row of the version you just rolled back to.

   Please note that any future changes made to the task will be based upon this marked version. However, those changes will not be included in this marked version. In order to include the changes, you must create a new version either manually or by promoting the task to the next environment such as Production.

Task overview: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]

4.9 Change Data Capture (Delta Loads)

You can use change data capture techniques to identify changes in a source table which occur between two points in time. For example, to identify changes between the end point of an initial or last load and the current date.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Post-Processing for SAP Integrated Business Planning [page 160]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
What is a Task? [page 142]
Change Execution Order for Targets [page 143]

Functions

SAP Cloud Integration for data services provides functions that allow you to save data along with a timestamp and then later retrieve it.

The `save_data (<VARCHAR_name>, <VARCHAR_data>)` function creates a persistent variable with a name (which could be the task name or any other string) and any piece of data. This data could be the end date timestamp of the previous load. The maximum data size is 255 characters.

The `get_data (<VARCHAR_name>)` function retrieves the stored data.

Example

Consider a single task containing global variables that can be set at run time. This task can be used for an initial load and later for delta loads. You use preload and postload scripts to call the necessary functions. The functions set values for global variables that can be used to filter data by date range.

> **Note**
>
> The same logic can be applied in a process by placing the preload script before a data flow and the postload script after it.

The following global variables are used:

Global Variable	Use	Type
$G_STARTDATE	Specify the start date of the data range	datetime
$G_ENDDATE	Specify the end date of the data range	datetime
$G_RESET	Switch between a fresh initial load or delta load. For the initial load use a dummy start date of 1900-01-01.	varchar(1)

Preload script

Create the following preload script:

```
# Start date
if (get_data('<task_name>') = '' or $G_RESET = 'Y')
  $G_STARTDATE = to_date('1900-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss');
else
  $G_STARTDATE = to_date(get_data('<task_name>'),'yyyy-mm-dd hh24:mi:ss');
# End date
if ($G_ENDDATE is null)
  $G_ENDDATE = sysutcdate();
print('Using query period from [$G_STARTDATE] to [$G_ENDDATE]');
```

Postload script

Create the following postload script:

```
print('Saving enddate for next startdate: [$G_ENDDATE]');
save_data('<task_name>',to_char($G_EDATE,'yyyy-mm-dd hh24:mi:ss'));
```

* Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

4.10 Post-Processing for SAP Integrated Business Planning

SAP Cloud Integration for data services tasks load data to staging tables in SAP Integrated Business Planning. A stored procedure within SAP Integrated Business Planning then performs post-processing validation checks and loads the data to the appropriate application tables.

Validation checks include:

- When loading transaction data, check that the corresponding master data is already loaded.
- Check for invalid special characters. For example, special characters such as ', <, or > are not allowed in product or customer names.
- Check master data records to ensure that duplicate records are not loaded.

In SAP Cloud Integration for data services you can define when you want the post-processing to occur and how SAP Cloud Integration for data services reports post-processing errors.

To define post-processing properties for Integrated Business Planning tasks or processes:

1. From the *Projects* tab, expand the project that contains the task or process that loads data to Integrated Business Planning.
2. Select the appropriate task or process and click *Edit*.
3. In the task or process, click *Execution Properties*.
4. In the *Post-Processing for Integrated Business Planning* section, set the appropriate values:

Option	Description
Status check duration (hours)	Amount of time that SAP Cloud Integration for data services periodically checks the status of the post-processing operation running in Integrated Business Planning. If no status (success or failure) is reported after the specified duration, SAP Cloud Integration for data services stops polling and logs an error indicating that the status check operation has stopped.
Begin post-processing	Specifies whether Integrated Business Planning should run the stored procedure after each data flow completes or after the entire task or process is executed. Your choice may be determined by the type of data being loaded into Integrated Business Planning. For example, master data may need to be loaded and processed before transactional data can be loaded successfully. A process may include multiple data flows and each data flow can load to a different target datastore. SAP Cloud Integration for data services detects the target object type and triggers post-processing only for targets in Integrated Business Planning datastores.
Treat 'Processed with Error' as success	Specifies how SAP Cloud Integration for data services reports errors returned by the post-processing. If the option is checked, after the data is loaded to the SAP Integrated Business Planning application tables then SAP Cloud Integration for data services reports that the task or process completed successfully. Any post-processing errors are reported in the logs, dashboard, and task statuses.

Option	Description
	By selecting this option, email notifications are sent only for actual data load failures, not for other post-processing errors.

5. Click *Done*.

The icons for tasks or processes that include post-processing contain a '!' symbol. Statuses are reported as described in the following table:

State of *Treat 'Processes with Error' as success* checkbox	Data load status for task or process execution	Post-processing in SAP Integrated Business Planning completes as:	Status result	Web services status
Selected	Succeeded	Succeeded	 Succeeded	`TASK:SUCCESS`
Selected	Succeeded	Failed	 Succeeded	`TASK:SUCCESS_WITH_ERRORS_D`
Deselected	Succeeded	Failed	 Failed	`TASK:SUCCESS_WITH_ERRORS_E`
Deselected	Failed	Not applicable	 Failed	`TASK:ERROR`

Task overview: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Optimizing SAP Integrated Business Planning Outbound Performance [page 162]
Project Operations

4.11 Optimizing SAP Integrated Business Planning Outbound Performance

Outbound task/process performance when loading data from IBP into HANA On-premise can be optimized by avoiding certain filter expressions.

Usage of the `TSTFR` and `TSTTO` functions combined with datetime functions in filter expressions cannot be pushed down to the source, thereby causing performance issues. Use `PERIODID` in filter experessions to narrow down the query and optimize performance instead.

> ### Example
>
> If you want to filter on results between a 4 week time frame, `PERIODID` functions representing weeks can be used to filter on weeks 0 – 4 instead.

Parent topic: Tasks, Processes, and Projects [page 141]

Related Information

What is a Task? [page 142]
What is a Process? [page 145]
Available Actions in Processes and Tasks [page 151]
Replicate a Task or Process [page 152]
Edit a Task or Process [page 153]
Promoting a Task or Process [page 155]
Versioning Tasks and Processes [page 156]
Roll Back to a Previous Version [page 157]
Change Data Capture (Delta Loads) [page 158]
Post-Processing for SAP Integrated Business Planning [page 160]
https://launchpad.support.sap.com/#/notes/2493042

5 Data Flows

Data flows define the movement and transformation of data from one or more sources to a single target.

By calling a web service function in parallel, you can reduce the performance bottleneck caused by row-by-row function calls.

Related Information

5.1 What is a Data Flow?

A data flow defines the movement and transformation of data from one or more sources to a single target. Within a data flow, transforms are used to define the changes to the data that are required by the target. When the task or process is executed, the data flow steps are executed in left-to-right order.

Although a data flow can have more than one data source, it can have only one target. This target must be an object in the target datastore that is associated with the data flow's parent task.

In a task, global variables and scripts that assign values to variables are defined at the task level and are applied to all data flows in that task.

In a process, global variables are defined at the process level. Include scripts in the process before or after data flows as defined by your business logic.

Parent topic: Data Flows [page 163]

Related Information

Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]
What is a Task? [page 142]
What is a Process? [page 145]
Data Flow Management [page 165]

Add a Data Flow from Scratch [page 167]
Design a Data Flow [page 171]

5.2 Data Flow Management

You can manage targets and data flows in the *Data Flows* tab of the task editor.

The *Data Flows* tab contains a table of all the data flows defined for the task, grouped according to their target objects. When a task is run, its data flows are executed in the order in which their target objects appear in the table (the data flows belonging to the target object at the top of the table are run first, then those of the next target object in the table, and so on).

> **i** Note
>
> If you want to execute data flows in parallel or to execute data flows from several tasks, consider using a process.

As needed you can modify existing data flows using the data flow editor.

If you need to create additional data flows you can either duplicate an existing data flow and then modify it to meet your needs or you can create a data flow from scratch.

Duplicate data flows

Duplicating a data flow gives you a good starting point for your new data flow. You can duplicate a data flow in the following ways:

- *Replicate* a data flow from one task to another.
  The target task must use the same source and target datastore types as the task of the data flow that you choose to replicate.
- Within a task, duplicate a data flow to a different target object using the *Copy to new target* action.

Create data flows from scratch

You can create a data flow from scratch in the following ways:

- Add a data flow to an existing target object (*Add Data Flow*).
- Add a data flow to a new target object (*Add Target Object*).

Parent topic: Data Flows [page 163]

Related Information

Duplicate a Data Flow [page 166]
Add a Data Flow from Scratch [page 167]
What is a Data Flow? [page 164]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]
What is a Data Flow? [page 164]

5.2.1 Duplicate a Data Flow

You can duplicate an existing data flow and then modify the duplicated data flow to meet your needs.

Replicate between tasks

You can replicate an existing data flow to a different task.

The target task must use the same source and target datastore types as the original task for the replicated data flow.

> **i** Note
>
> You cannot replicate a data flow in the production environment.

1. From the *Projects* tab, select the task that contains the data flow you want to replicate and click *Edit*.
2. In the *Data Flows* tab of the task editor, select the data flow you want to replicate and click *Actions* > *Replicate*.
3. Select the project and task to which you want to add the replicated data flow and click *OK*.
4. Enter a name for the replicated data flow.
5. If the source or target datastore is a File Format Group, click the *Verify* icon beside the new name to ensure that the name you entered is unique, then modify it if necessary.

   Also for File Format Group datastores, resolve entries under *Related duplicated tables* as needed.

6. Click *OK* to exit the task editor.

Modify the new data flow if necessary.

Copy to a new target

Within a task, you can create a copy of a data flow and use it to load data to a different target object.

1. From the *Projects* tab, select the desired task and click *Edit*.
   The task editor opens.
2. In the *Data Flows* tab of the task editor, select the task you want to copy and click *Actions* > *Copy to new target*.
3. Enter a name for the new data flow.
4. Select an existing target object or import a new target object and then click *Copy Data Flow*.
   The data flow is copied to the target object.

As needed, modify the new data flow.

5.2.2 Add a Data Flow from Scratch

Data flows can be added to a task when the task is created or at a later time.

Create a new data flow when there is no suitable candidate to copy or replicate.

Creating a task from scratch is described in the following steps:

1. In the *Projects* tab, select the task you want to add the data flow to and click *Edit*.
2. In the *Data Flows* tab, do one of the following:
   - Add a data flow to an **existing** target object.
     Select the target object and click *Actions* > *Add data flow*.
   - Add a data flow to a **new** target object.
     Click *Add Target Object*, select the new target object, and click *Create Data Flow*.
3. In the *Add Data Flow* screen, enter properties for the new data flow.

   The available options depend on the data flow's target option type.

   For **HANA Cloud targets**, the first time a task runs, all data is loaded from the source. For subsequent runs, the load option determines how the original data is treated. Based on the application the data is being loaded to, some options may not be available.

   > **i Note**
   >
   > The options are not available for SAP Integrated Business Planning products.

   The load options are described in the following table:

Option	Description
No selection	Loads new records.
Auto correct load based primary key correlation	Updates existing record or inserts new record based on the primary keys defined in the target object. Updates occur for subsequent loads of the same records (same key). Inserts occur for new records (new keys). **i Note** If there is not a primary key match, records are appended to the object and duplicate records are inserted.
Delete data from table before loading	Clears the existing contents of the table before loading.

For **flat file targets**, the options are described in the following table:

Option	Description
Root Directory	Path name on the SAP Data Services Agent host system. **i Note** The SAP Data Services Agent must also be configured to have access to the directory that contains the source or target files. For more information, see the *Agent Guide*.
Encrypt with PGP	Creates a PGP-protected file.
Remote File Path	Path on the SFTP file server. Option is only available if SFTP has been configured for the target datastore.
User ID of the External Public Key	An email address, name, or other identifying information. It was specified when the external (third-party) public key was generated.
Include Digital Signature	Used to verify the authenticity of the data's origin and integrity.
Delete file before loading	Removes the existing file before loading a new file.

For **SuccessFactors adapter targets**, the options are described in the following table:

Option	Description
Batch Size	The number of data rows to process as a single batch. Default: **`200`**
Column delimiter	The character sequence used to separate data between columns. Default: **`/127`**
Row delimiter	The character sequence used to separate data between rows. Default: **`/007`**

Option	Description
Auto correct load based on primary key correlation	Updates existing record or inserts new record based on the primary keys defined in the target table. Updates occur for subsequent loads of the same records (same key). Inserts occur for new records (new keys). **i Note** If there is not a primary key match, records are appended to the table and duplicate records are inserted.
Use audit	Log load success or failure at the data level.

4. Click *OK* to add the new data flow to the task.

After adding the data flow, design it in the data flow editor.

Related Information

Loading into PGP-protected Target Files [page 214]
Design a Data Flow [page 171]

5.3 Data Flow Design

A data flow may contain multiple sources, but has a single target object.

The first transform takes its input from source tables or files. The input is transformed as needed and mapped to the Output pane. Subsequent transforms in the data flow take as input the output columns of the previous transform step. The final transform must be a target transform. SAP Cloud Integration for data services automatically creates the correct type of target transform based on the target type.

About the target schema

The Output pane of the final transform shows the target object schema. Changes to the schema cannot be made in the Output pane of the target transform. If changes are required, they must be made in the database, file format or web service. Changed database and web service objects must be reimported in the datastore. Changed file format objects do not need to be reimported.

i Note

In order to reimport a web service object, the web service must be up and running.

Transform order in a data flow

Within a data flow, data must be transformed in a specific order. First any ABAP transforms, (for SAP sources), next any additional transforms, and finally a target transform.

The target transform is the only required transform in a data flow. All other transforms are optional and serve to manipulate the data as needed to meet your requirements.

Considerations

Before you begin to create a data flow from scratch, consider the following points:

- For each target object, determine what sources are required and what transformations are needed for that data. With that information, you can map out what transform types you will use.
- Consider what global variables will be useful.
  Values assigned to global variables apply across all data flows within a task.
- If you have an existing data flow that you can adapt, you can create a duplicate and then modify the duplicated data flow as needed.

Best Practices

Best practice when creating a data flow from scratch is to begin by defining the first transform in the data flow. This is the transform that extracts the data from your source and may also manipulate your data. As needed, you can add intermediate transforms to manipulate the data. The target transform loads data to the target and must be the final transform in the data flow. As such, it would be the last transform you define.

Best practice is to rename columns or edit data types so they match those in the target schema as early in the data flow as possible. By doing this you can take advantage of Automap functionality in the Target Query transform.

Parent topic: Data Flows [page 163]

Related Information

Open the Data Flow Editor [page 171]
Design a Data Flow [page 171]
What is a Data Flow? [page 164]
Data Flow Management [page 165]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]

Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]
Duplicate a Data Flow [page 166]
Transform Types [page 172]
Add a Data Flow from Scratch [page 167]
Map Import and Export Columns [page 176]
Add or Refresh Columns Displayed in a Target Object [page 179]
Join Tables [page 180]
Filter Data [page 181]
Sort Data [page 187]
Group Result by Columns [page 188]

5.3.1 Open the Data Flow Editor

Open the data flow editor to design and debug data flows.

Follow the steps below to open a data flow for editing.

1. From the *Projects* tab, expand the project that contains the task and data flow you want to edit.
2. Select the task that contains the data flow you want to edit and click *Edit*.
3. From the Data Flows tab of the task, select a data flow and click *Actions* > *Edit*.
   The data flow editor opens.

5.3.2 Design a Data Flow

Use the data flow editor to design data flows that define how data is extracted from its source, transformed, and loaded to a target. The data flow editor can also be used to debug or refine existing data flows.

The following steps describe how to use the data flow editor to define a data flow from scratch.

1. If the data flow editor is not already open, open it.

   When a data flow is defined from scratch, by default the canvas contains a target transform connected to the target object that was selected when the data flow was created.
2. From the object palette, drag a source object to the canvas.
3. Select the appropriate input and click *OK*.
4. If your data flow requires input from more than one source, repeat steps 2 and 3.
5. As needed, drag transforms from the object palette to the canvas and enter appropriate names.

   The transforms available in the object palette depend on the type of source datastore.

The software automatically creates an ABAP group when you drag an ABAP Query, ABAP Aggregation or custom ABAP transform onto the canvas. This group must be first in the data flow. Sources are automatically placed inside this group. In order for data to progress through the data flow, you must connect the final transform inside the ABAP group to the *ABAP Endpoint* object.

When using SAP HANA analytic or calculation views as a source, you must add an Aggregation transform after the source in your data flow.

6. Connect transforms to each other to define the sequence of the data flow. Click the yellow square on the right edge of the upstream object and drag toward the yellow square on the left edge of the object that you want to connect to.

   The system executes the steps in left-to-right order. Connections are indicated by lines that connect the output of one object to the input of another.

7. Double-click a transform to configure the details of how data passes through it.

   You can edit the column mappings, apply filters, create joins, and perform other actions.

8. (Optional) View a sample of the design-time data at any point in the data flow where the Design-time Data Viewer () is available.

9. When you are done editing the data flow design, click *Done* to save it and close the editor.

10. In the task editor, select the data flow and click *Validate*.

    Based on the validation results, make any necessary changes to the data flow.

Related Information

Transform Types [page 172]
Add a Data Flow from Scratch [page 167]
Open the Data Flow Editor [page 171]
Map Import and Export Columns [page 176]
Group Result by Columns [page 188]
View Data During Data Flow Design and Debug [page 210]

5.4 Transform Types

A transform is a step in a data flow that acts on a data set. A data flow may contain one or more transforms.

Available transforms and their purposes are shown in the following table:

Transform type	Purpose
Query	Retrieves a data set from a source and optionally transforms the data according to the conditions that you specify.

Transform type	Purpose
Target Query	A special type of Query transform that must be the last transform before the target. In addition to Query transform capabilities, the Target Query transform also loads the data to the target.
Aggregation	Collects data across multiple records. An Aggregation transform groups by the specified columns and then aggregates the data on a per column basis.
XML Map	Retrieves one or more flat or hierarchical source data sets and produces a single target data set. You can use the XML Map transform to perform a variety of tasks. For example: • You can create a hierarchical target data structure such as XML from a hierarchical target data structure. • You can create a hierarchical target data structure based on data from flat tables. • You can create a flat target data set such as a database table from data in a hierarchical source data structure.
Target XML Map	A special type of XML Map transform that must be the last transform before the target when the target is an XML template. In addition to XML Map transform capabilities, the Target XML Map transform also defines the schema of the target XML file and loads the data to the target.
XML Batch	Groups of flat or hierarchical data sets as blocks of rows before sending them to the next transform. For example, you might use XML Batch to create groups of rows before sending them to a web service target.
Web Service Call	Loads structured data using a call to an external web service target.
Row Generation	Generates a column filled with integer values starting at zero by default and incrementing by one in each row. You can set the column starting number in the *Row number starts at* option and specify the number of rows in the *Row count* option. For flexibility, you can enter a global variable. Row Generation transform does not have any data inputs.
ABAP Query	Retrieves a data set from an SAP Applications source and optionally transforms the data inside the SAP application according to the conditions that you specify. The transformed data is returned to SAP Cloud Integration for data services.
ABAP Aggregation	Collects data across multiple records from an SAP Applications source. An ABAP Aggregation transform groups by the specified columns and then aggregates the data on a per column basis inside the SAP application. The transformed data is returned to SAP Cloud Integration for data services.
Custom ABAP	Allows you to use your own ABAP programs.

Considerations for choosing a transform type

When aggregating data from SAP applications sources, for the best performance use an ABAP Aggregation transform rather than an Aggregation transform. The ABAP Aggregation transform pushes down the operations to the SAP application server.

When performing joins where all sources are extractors, for best performance use a Query transform rather than an ABAP Query transform. However, if you need to join an extractor to a table, then you must use an ABAP Query transform.

Parent topic: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]
Data Flow Management [page 165]
Map Import and Export Columns [page 176]
Add or Refresh Columns Displayed in a Target Object [page 179]
Join Tables [page 180]
Filter Data [page 181]
Group Result by Columns [page 188]
Define a Custom ABAP Transform [page 194]
Loading into PGP-protected Target Files [page 214]

5.5 Transform Operations

A transform step applies a set of rules or operations to transform the data. You can specify or modify the operations that the software performs.

Data transformation can include the following operations:

- Map columns from input to output
- Join data from multiple sources
- Choose (filter) the data to extract from sources
- Perform functions on the data

- Perform data nesting and unnesting
- Construct XML Map iteration rules
- Define a web service response

> **i** Note
>
> Not all transforms can perform each operation.

Parent topic: Data Flows [page 163]

Related Information

Map Import and Export Columns [page 176]
Add or Refresh Columns Displayed in a Target Object [page 179]
Join Tables [page 180]
Filter Data [page 181]
Sort Data [page 187]
Group Result by Columns [page 188]
Specify Distinct Output Rows [page 190]
Construct XML Map Iteration Rules [page 191]
Construct an Input Request for a Web Service Call [page 193]
Custom ABAP Transforms [page 194]
Generate and Load an ABAP Program [page 199]
Enrich Data with the Lookup Function [page 200]
Group Hierarchical Data into Blocks [page 201]
Improve Transform Performance [page 202]
What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]

5.5.1 Map Import and Export Columns

As your data moves from its source to its target, it passes through a sequence of one or more transforms. You can map input to output columns or view existing mappings in the transform workspace and in the *Mapping* tab.

A column in a table or extractor is represented by a row in the Input or Output panes. Mapping syntax considerations include the following guidelines:

- Extractor names must be enclosed in double quotation marks ("), for example, `"0MATERIAL_ATTR_SOP".MATNR`.
- A hash mark (#) indicates a comment.
- A hash mark (#) cannot be included within a mapping expression. It is interpreted as the start of a comment and anything to the right of the hash mark is ignored. A validation error may occur because only part of the script statement (to the left of the hash mark) is validated.

For information about how to sort and filter the names, data types, and descriptions displayed in the lists of inputs and outputs when mapping, see Sorting and Filtering Columns in the Input and Output Panes [page 178].

To map input columns to output columns, navigate to a transform in a data flow and do one of the following actions:

Option	Description
Review the current mapping	If a column has already been mapped, the mapping icon appears in the first column of the Output pane. Click a column in the Output pane. The column in the Input pane from which it is mapped is highlighted and the mapping is displayed in the Mapping tab of the Transform Details. A red exclamation point icon indicates that the mapping is invalid or may contain an invalid expression. You can review the mapping in the Mapping tab of the Transform Details.
Create a simple mapping	Drag one or more columns from the Input pane to the Output pane. The mapping icon appears and the column is mapped directly with no changes. → Tip In a Target Query, *Automap by name* is available. *Automap by name* maps all columns from the Input pane to columns with the same name that exist in the Output pane (target). *Automap by name* requires that the Input pane contains only one source.
Create a complex mapping	Use function helpers or operators to create a mapping that consists of more than a single input column. • Build a function by clicking the function name in the categories in the Mapping tab. For example, you might want to apply the `decode` function based on the value of an input column: `Decode(table1.status_ID = `0`, `NO`, table1.status_ID = `1`, `YES`, `N/A`)` • Drag one or more columns from the Input pane to the Mapping tab and modify it by applying a function or using operators (+, -, *, /, !=, and so on).

Option	Description
	For example, you could use the concatenation operator (| |) to combine discrete first and last name input columns into a single output column: `table1.first_name || ` ` || table1.last_name` For more information, see Expression operators [page 177].
Add an Output column	In the Output pane, in the bottom row, click the Insert icon and complete the required fields in the dialog box to create a new column. **i Note** You cannot add a column in the Output pane of a Target Query transform. Those columns are defined by the Target table.

Related Information

Expression operators [page 177]
Sorting and Filtering Columns in the Input and Output Panes [page 178]
Data Flow Management [page 165]
Add or Refresh Columns Displayed in a Target Object [page 179]
Join Tables [page 180]
Filter Data [page 181]
Group Result by Columns [page 188]
Specify Distinct Output Rows [page 190]
Construct XML Map Iteration Rules [page 191]

5.5.1.1 Expression operators

You can use expression operators to construct mapping expressions that consist of more than a single input column.

SAP Cloud Integration for data services supports the following operators, listed in order of precedence:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division

Operator	Description
=	Assignment, comparison
<	Comparison, less than
<=	Comparison, less than or equal to
>	Comparison, greater than
>=	Comparison, greater than or equal to
!=	Comparison, not equal to
||	Concatenate
%	Return the remainder when one number is divided by another
AND	Logical AND
OR	Logical OR
NOT	Logical NOT
IS NULL	Comparison, is a NULL value
IS NOT NULL	Comparison, is not a NULL value

Related Information

Map Import and Export Columns [page 176]

5.5.1.2 Sorting and Filtering Columns in the Input and Output Panes

Items in the Input and Output panes display in the order that they are received from the data source. When preparing to map columns for transforms, sorting and filtering the list of names, data types, and descriptions may make your mapping effort easier.

Sorting

You can sort the list of table and extractor columns in the Input and Output panes by clicking on *Name*, *Data Type*, and *Description*. A bold arrow indicates either an ascending or descending alphanumeric sort.

Filtering

You can filter the list of table and extractor columns in the Input and Output panes by entering text in one or more of the text fields beneath *Name*, *Data Type*, and *Description* and then pressing *Enter*. The system accepts partial entries as well as numbers in these fields and returns any name, data type, or description containing the text you have entered in the respective column's text field. You can also utilize RegEx operators when filtering in the Input and Output panes; some commonly-used filter operations are shown in the following table:

Character	Function	Example of Filter	Example of Results
|	Matches rows of the expression on the right AND/OR the left side	id|file	"id", "file"
[]	Matches any one of the enclosed characters	[abc]id	"aid", "bid" and "cid"
-	The minus sign represents a range of characters	[a-d]1	"a1", "b1", "c1" and "d1"
.	The dot matches any single character	a.b	"aab", "abb", "acb", ... "azb", "a!b", etc.
*	Indicates zero or more occurrences of the preceding element	ac*b	"ab", "acb", "accb", "acccb", etc.
+	Indicates one or more occurrences of the preceding element	ac+b	"acb", "accb", "acccb", etc.
?	Indicates zero or one occurrences of the preceding element	ac?b	"ab" and "acb"
{n}	Indicates the preceding item is matched exactly n times	a{2}	"aa"
()	Encloses an entire expression	a(c+|f+)b	"acb", "accb", ... "afb", "affb", etc.
^	Matches all rows starting with the succeeding element	^abc	Displays results starting with "abc"
$	Matches all rows ending with the preceding element	abc$	Displays results ending with "abc"
\	Escapes a character	\?	Displays results containing "?"

To reset a filtered list, delete any text you entered in the filter text fields, then press *Enter*.

Considerations When Sorting and Filtering Columns for Transforms

- An asterisk (*) does not function as a wildcard on its own. You must use an asterisk in combination with the dot (.) special character. For example, filtering with **`abc.*`** returns all text strings that begin with “abc”.
- To include any nested items in your sort or filter results, you must first expand their parent nodes.
- Sorted and filtered lists are not saved when you leave the Transform page.

5.5.2 Add or Refresh Columns Displayed in a Target Object

You may need to load data for a column that exists in a target object in your target application, but isn't already populated by your current tasks and data flows.

- In your project you have identified the task and data flow that you need to modify.

- The target object already contains the additional column needed to meet your requirements.
- If your target is a file format, from the Datastores tab, add additional columns to the file format.

To refresh columns displayed in an existing target object in a data flow:

1. If the column does not display in the target object, reimport the target object:
   a. From the *Datastores* tab, select the datastore which contains your target object.
   b. Click the *Import Objects* icon.
   c. Select the object you want to reimport.
   d. Click *Import*.

   > **i** Note
   >
   > If your target is a file format, columns added to the file format are automatically reflected in the Output pane of the Target Query.
2. From the *Projects* tab, select the task you want edit and click *Edit*.
3. Select the relevant data flow and click *Edit*.
4. In the final transform, locate the new column.

   > → Tip
   >
   > The new column has not yet been mapped and thus will not have a mapped icon in the mapping column.
5. Beginning at the upstream step in your data flow where the source object needed for the new column is introduced, propagate the column through the interim transforms.

   As needed, edit the mappings or add additional transforms to the data flow.
6. In the Target Query transform, map the column from the Input to Output panes.

5.5.3 Join Tables

You can use the Join tab to join two or more source tables. You specify join pairs and join conditions based on primary/foreign keys and column names.

To join two or more tables:

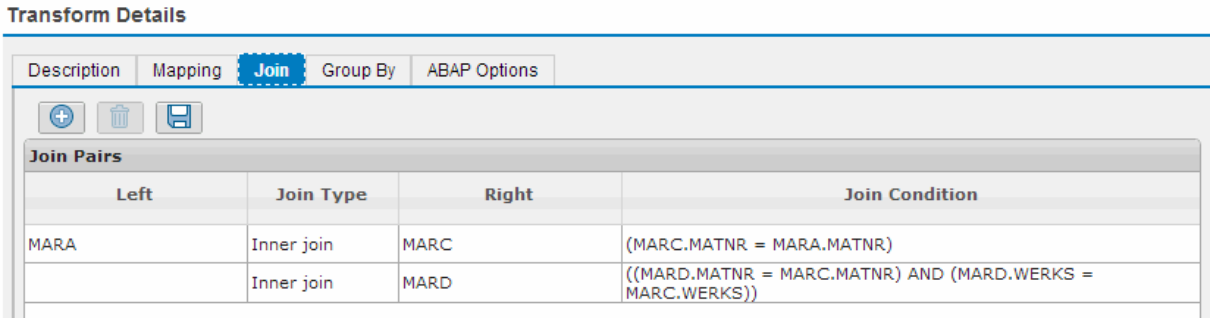
1. In the Edit Data Flow view, select the transform in which you want to perform the join.
2. If the tables you want to join are not already available in the *Input* pane, click *New* to add additional tables.
3. In the *Transform Details*, in the *Join* tab, click the plus icon to add a new join.
4. Select the tables you want to join and the join type.
5. Type a join condition.
6. Click *Save*.
7. If needed, create additional join conditions.

   Subsequent join pairs take the results of the previous join as the left source.

   > **i** Note
   >
   > In an ABAP Query, mixed inner and left outer joins are not supported.

Example: Example: Multiple join conditions

For example, given three tables, MARA, MARC, and MARD with appropriate primary key/foreign key relationships, you might join the tables as shown in the following screenshot:



Related Information

Data Flow Management [page 165]
Filter Data [page 181]

5.5.4 Filter Data

You can filter or restrict your data using the Filter tab.

To filter your data, perform these steps:

1. In the Edit Data Flow wizard, select the transform in which you want to add a filter.
2. Click the *Filter* tab.
3. (Optional) If you want to ignore identical duplicate rows so that your results contain only distinct rows, click *Select Distinct Rows*.

   This is similar to specifying a `SELECT DISTINCT` SQL statement.
4. From the Input pane, drag the column containing the data you want you filter and drop it in the Filter field.
5. As needed, type filter conditions or use the built-in functions.

   Examples of filter conditions are shown in the following table:

Type	Filter	Description
Constant	`VBAK.SPART = '07'`	In a sales order header table, filters for rows containing Division 7.
Complex	`VBAP.NETWR < ( VBAP.WAVWR * VBAP.ZMENG )`	Filters for rows where the net value of the sales order is less than the product of the cost of the item multiplied by the quantity ordered.

Type	Filter	Description
Global variable	`(CSKB.ERSDA >= $G_SDATE)`	In a cost elements table, filters for rows with a date equal to or more recent than the value of the global variable $G_SDATE.
Function	`BKPF.CPUDT >= sysdate() -1`	Filters for Financial Documents Header rows created yesterday or more recently.

6. If your source is an adapter datastore, you can also filter the rows retrieved from the datastore in the *Adapter Source* tab.

   The columns that you can use for adapter-based filtering depend on the type of adapter.

> **! Restriction**
>
> When you filter in an XML Map transform, source columns must come from the source schemas in the current iteration rule or those that appear in the iteration rules associated with the parents of the selected target schema. Additionally, the path from the column being used to the source schema must contain no repeatable schemas.
>
> Target columns must come from the selected target schema or parents of the selected target schema. Additionally, the path from the column being used to the target schema must contain no repeatable schemas.

> **i Note**
>
> If your expression contains varchar comparisons, SAP Cloud Integration for data services ignores trailing blanks in the data. For Oracle data, use the rtrim or rpad functions if the number of trailing blanks might differ on either side of the comparison.

Related Information

Filter Options [page 182]

Data Flow Management [page 165]

Map Import and Export Columns [page 176]

5.5.4.1 Filter Options

Use built-in filter options to filter data within SAP Cloud Integration for data services.

Conversion [page 183]
: Conversion functions convert data from one type to another.

Cryptographic [page 183]
: List and descriptions of aggregate filter options.

Date [page 184]
: List and descriptions of date filter options.

Lookup [page 185]
List and description of lookup filter option.

Math [page 185]
List and descriptions of math filter options.

Miscellaneous [page 185]
List and descriptions of miscellaneous filter options.

String [page 186]
List and descriptions of string filter options.

Validation [page 187]
List and descriptions of validation filter options.

Related Information

Sort Data [page 187]

5.5.4.1.1 Conversion

Conversion functions convert data from one type to another.

Function	Description
cast	Converts an expression of one data type to another.
julian_to_date	Converts a Julian value to a date.
to_char	Converts a date or numeric data type to a string.
to_date	Converts an input string to a date type based on the input format.
to_decimal	Converts a varchar to a decimal.
to_decimal_ext	Converts a varchar to a decimal and includes precision as a parameter.

5.5.4.1.2 Cryptographic

List and descriptions of aggregate filter options.

Function	Description
decrypt_aes	Decrypts the input string using the user-specified passphrase and key length using the AES algorithm.
decrypt_aes_ext	Decrypts the input string with the user-specified passphrase, salt, and key length using the AES algorithm.

Function	Description
encrypt_aes	Encrypts the input string using the user-specified passphrase and key length using the AES algorithm.
encrypt_aes_ext	Encrypts an input string using the specified passphrase, salt, and key length with the AES algorithm.

5.5.4.1.3 Date

List and descriptions of date filter options.

Function	Description
add_months	Adds a given number of months to a date.
concat_date_time	Returns a datetime from separate date and time inputs.
date_diff	Returns the difference between two dates or times.
date_part	Extracts a component of a given date.
day_in_month	Determines the day in the month on which the given date falls.
day_in_week	Determines the day in the week on which the given date falls.
day_in_year	Determines the day in the year on which the given date falls.
fiscal_day	Converts a given date into an integer value representing a day in a fiscal year.
isweekend	Indicates that a date corresponds to Saturday or Sunday.
julian	Converts a date to its integer Julian value, the number of days between the start of the Julian calendar and the date.
last_date	Returns the last date of the month for a given date.
local_to_utc	Converts the input datetime of any time zone to Coordinated Universal Time (UTC).
month	Determines the month in which the given date falls.
quarter	Determines the quarter in which the given date falls.
sysdate	Returns the current date as listed by the Job Server's operating system.
systime	Returns the current time as listed by the Job Server's operating system.
sysutcdate	Returns the current UTC date as listed by the operating system of the server where the Agent is installed.
utc_to_local	Converts an input that is in Coordinated Universal Time (UTC) to the set time zone value.
week_in_month	Determines the week in the month in which the given date falls.
week_in_year	Determines the week in the year in which the given date falls.
year	Determines the year in which the given date falls.

5.5.4.1.4 Lookup

List and description of lookup filter option.

Function	Description
lookup	Finds a value in one table or file based on values in a second table or file.

5.5.4.1.5 Math

List and descriptions of math filter options.

Function	Description
abs	Returns the absolute value of an input number.
ceil	Returns the smallest integer value greater than or equal to an input number.
floor	Returns the largest integer value less than or equal to an input number.
ln	Returns the natural logarithm of the given numeric expression.
log	Returns the base-10 logarithm of the given numeric expression.
mod	Returns the remainder when one number is divided by another.
power	Returns the value of the give expression to the specified power.
rand	Returns a random number between 0 and 1.
rand_ext	Returns a random number between 0 and 1.
round	Rounds a given number to the specified precision.
sqrt	Returns the square root of the given expression.
trunc	Truncates a given number to the specified precision.

5.5.4.1.6 Miscellaneous

List and descriptions of miscellaneous filter options.

Function	Description
decode	Returns an expression based on the first condition in the specified list that evaluates to TRUE.
gen_row_num	Returns an integer value beginning with 1 then incremented sequentially by 1 for each additional call. This function can be used to generate a column of row IDs.
gen_row_num_by_group	Returns group row number of the record.
gen_uuid	Returns a unique varchar string identifier.

Function	Description
greatest	Returns greatest of the list of one or more expressions.
ifthenelse	Allows conditional logic in mapping and selection operations.
is_group_changed	Returns 1 if the group is changed, 0 otherwise.
job_name	Returns the name of the job in which the call to this function exists.
least	Returns the least in a list of one ormore expressions.
nvl	Replaces NULL values.
previous_row_value	Returns the column value of previous row.
raise_exception	Calling this function causes an exception to be generated.
raise_exception_ext	Same as raise_exception, but takes a second parameter for an exit code.
wait_for_file	Returns the existing files that match the input file pattern.

5.5.4.1.7 String

List and descriptions of string filter options.

Function	Description
ascii	Returns the decimal value of the first character for the given string using ASCII character set. If the character passed is not a valid ASCII character, -1 is returned.
chr	Get character representation of provided ASCII value.
index	Returns the index of a given word in a string.
init_cap	Changes the characters in a string to title case.
length	Returns the number of characters in a given string.
literal	Returns an input constant expression without interpolation. Allows you to assign a pattern to a variable without interpolation.
lower	Changes the characters in a string to lowercase.
lpad	Pads a string with characters froma specified pattern.
lpad_ext	Pads a string with logical characters from a specified pattern.
ltrim	Removes specified characters from the start of a string.
ltrim_blanks	Removes blank characters from the start of a string.
ltrim_blanks_ext	Removes blank and control characters from the start of a string.
match_pattern	Matches whole input strings to simple patterns supported by Data Services. This function does not match substrings.
match_regex	Matches whole input strings to the pattern that you specify with regular expressions (regular expressions based on the POSIX standard) and flags. This function does not match substrings.
match_simple	

Function	Description
print	Prints the given string to the trace log.
replace_substr	Returns a string where every occurrence of a given search string in the input is substituted by the given replacement string.
replace_substr_ext	Takes an input string, replaces specified occurrences of a specified sub-string with a specified replacement and returns the result. You can also use this function to search for hexadecimal or reference characters.
rpad	Pads a string with characters from a given pattern.
rpad_ext	Pads a string with logical characters from a given pattern.
rtrim	Removes given characters from the end of a string.
rtrim_blanks	Removes blank characters from the end of a string.
rtrim_blanks_ext	Removes blank and control characters from the end of a string.
substr	Returns a specific portion of a string starting at a given point in the string.
translate	Translates selected characters of an input string into other specified characters.
upper	Changes the characters in a string to uppercase.
word	Returns one word out of a string.
word_ext	Returns the word identified by its position in a delimited string.

5.5.4.1.8 Validation

List and descriptions of validation filter options.

Function	Description
is_valid_date	Indicates if an expression can be converted into a valid date value.
is_valid_datetime	Indicates if an expression can be converted into a valid datetime value.
is_valid_decimal	Indicates if an expression can be converted into a valid decimal value.
is_valid_double	Indicates if an expression can be converted into a valid double value.
is_valid_int	Indicates if an expression can be converted into a valid integer value.
is_valid_real	Indicates if an expression can be converted into a valid real value.
is_valid_time	Indicates if an expression can be converted into a valid time value.

5.5.5 Sort Data

You can sort the order of your data by using the Order By tab.

To sort your data:

1. In the Edit Data Flow wizard, select the transform in which you want to sort your data.

Sorting is supported in the Query, ABAP Query, and XML Map transforms.

2. Click the *Order By* tab.
3. From the Input pane, drag the column containing the data you want you use to sort and drop it into the Order By table.
4. Specify whether you want to sort in ascending or descending order.
5. Add additional columns to the Order By tab and arrange them as necessary.

   For example, you might choose to sort your data first by country in ascending order, and then by region in descending order.

> **i** Note
>
> The data will be sorted in the order that the columns are listed in the Order By tab.

5.5.6 Group Result by Columns

Use the GROUP BY tab to specify a list of columns for which you want to combine output.

For each unique set of values in the group by list, SAP Cloud Integration for data services combines or aggregates the values in the remaining columns. For example, you might want to group sales order records by order date to find the total sales ordered on a particular date.

The Aggregation and ABAP Aggregation transforms require that you specify columns to use to group the result set. All columns must either be included in a Group By or must be aggregated. To aggregate, add new columns to output with appropriate type and other info, then type in the mapping and choose an aggregate function.

To create a Group By statement:

1. In the Edit Data Flow view, select the transform in which you want to perform the group by.
2. In the *Transform Details*, click the *Group By* tab.
3. From the Input pane, drag one or more columns to the Column field in the *Group By* tab.
4. As needed, order the columns using the up and down arrows.
5. Click *Save*.
6. In the Output pane, insert a new column and enter the appropriate name, data type and other information.
7. In the *Transform Details*, in the *Mapping* tab, use the Aggregate function to create the mapping.

> **i** Note
>
> Each column must be either used in the Group By or mapped with an aggregation function.

> **!** Restriction
>
> When you use GROUP BY in an XML Map transform, you can specify either source or target columns in the grouping list.
>
> When source columns are used, they must descend from the source schema in the current iteration rule. In addition, the path from the source schema to the column must contain no repeatable nodes.
>
> When target columns are used, they must descend from the selected target schema. In addition, the path from the selected target schema to the column must contain no repeatable nodes.

If you specify a grouping list, then all columns in the selected output schema must be either in the grouping list or mapped to an aggregate function such as avg, count, max, min, or sum.

Related Information

XML Map Grouping Methods [page 189]
Data Flow Management [page 165]
Transform Types [page 172]
Map Import and Export Columns [page 176]

5.5.6.1 XML Map Grouping Methods

The XML Map transform groups output items in different ways depending upon the columns specified and whether or not aggregation functions are used.

Grouping method	Description
Simple grouping	The XML Map transform groups output items together according to the unique values of the grouping list when the following conditions are met: • Source or target columns are specified in the grouping list • If source columns are specified, no aggregation functions are defined in the selected target schema In this grouping method, no items are removed from the output data set.
Group aggregation	The XML Map transform performs exactly like a standard SQL GROUP BY clause when the following conditions are met: • Source columns are specified in the grouping list • Aggregation functions are defined in the selected target schema • Columns in the aggregation functions descend from the sources in the current iteration rule • Paths from the iterating sources to the columns do not contain any repeatable nodes **i Note** All columns in the output schema must be either part of the grouping list or mapped to an aggregate function such as avg, count, max, min, or sum.

Grouping method	Description
Instance aggregation	The XML Map transform evaluated the aggregation functions for each of the items in the output data set when the following conditions are met: • Aggregation functions are defined under the selected target schema • Columns used in the aggregation functions descend from the sources in the current iteration rule • Paths from the sources to the columns being used contain repeatable nodes The XML Map transform also evaluates the aggregation functions for each of the items in the output data set when the following conditions are met: • Aggregation functions are defined under the selected target schema • Columns used in the aggregation functions descend from the selected target schema • Paths from the selected target schema to the columns being used contain repeatable nodes

! Restriction

You cannot use both group and instance aggregation at the same time.

5.5.7 Specify Distinct Output Rows

In an XML Map transform, if a column specified in the Distinct tab contains a distinct value, the row is a new output row.

To add a column to the Distinct columns list, select the column in the output schema area and drag it to the list in the Distinct tab. SAP Cloud Integration for data services adds the column to the bottom of the list.

To remove a column, select the column and click the delete icon.

To consider the entire output row as distinct, select Whole row is DISTINCT.

! Restriction

You cannot specify both source and target columns in the Distinct tab at the same time.

When source columns are used, they must descend from the source schemas in the current iteration rule. In addition, the path from the source schema to the column must contain no repeatable nodes.

When target columns are used, they must descend from the selected target schema. In addition, the path from the selected target schema to the column must contain no repeatable nodes.

5.5.8 Construct XML Map Iteration Rules

In an XML Map transform, iteration rules define how the output data set for the selected output schema is calculated.

An iteration rule is associated only with a repeatable target node, and defines how to construct the instances of the target schema from the source data. It is a mechanism to specify the input data sets and the way to combine them to create the target data set.

In the iteration rule tab, a hierarchical tree represents the logical combination of operations and input schemas that form a rule. Each operation in the rule is displayed as a node and may contain other operations or input schemas as children.

Use the iteration rule tab to create iteration rules for each repeatable schema in your output:

Adding a new element to a rule

From the Create icon, choose Create Rule Operator and specify the type of operation to perform.

Elements that can be added to an iteration rule include the following:

Element	Description
INNER JOIN	Performs a SQL INNER JOIN on the sources. Create the expression to use for the join condition in the On area of the Iteration Rule tab. When you create the expression, you can use the following types of columns: • Source columns from the sources under the current operation and the left side of the current iteration rule tree. • Source columns from the sources that appear in the iteration rules associated with the parent schemas of the selected target schema. • Target columns from the parent schemas of the selected target schema. **! Restriction** When using a source column, the path from the column being used to the source schema must contain no repeatable schemas. **! Restriction** When using a target column, it must be a scalar column and descend from the parent schema of the selected target schema. In addition, the path from the parent schema to the target column must contain no repeatable schemas.

Element	Description
LEFT OUTER JOIN	Performs a SQL LEFT OUTER JOIN on the sources. Create the expression to use for the join condition in the On area of the Iteration Rule tab. When you create the expression, you can use the following types of columns: • Source columns from the sources under the current operation and the left side of the current iteration rule tree. • Source columns from the sources that appear in the iteration rules associated with the parent schemas of the selected target schema. • Target columns from the parent schemas of the selected target schema. **! Restriction** When using a source column, the path from the column being used to the source schema must contain no repeatable schemas. **! Restriction** When using a target column, it must be a scalar column and descend from the parent schema of the selected target schema. In addition, the path from the parent schema to the target column must contain no repeatable schemas.
* - Cross operation	Performs a Cartesian product of two or more sources. When the sources have no parent-child relationship, the behavior is the same as a standard SQL CROSS JOIN. When the sources have a parent-child relationship, the Cartesian operation provides a mechanism to iterate through all instances of the repeatable elements identified by the source schemas in the operation in the document order.
|| - Parallel operation	Combines corresponding rows from two or more sources to generate the output set. For example, the first rows in a pair of input tables are combined to become the first row of the output set, the second rows are combined to become the second output row, and so on. If the sources have different numbers of rows, the output set will contain the same number of rows as the largest source. For extra rows in the output set that contain data from only one source, the additional columns that would contain data from the other sources are considered empty. **i Note** The Parallel operation is not a standard SQL operation.
Available input schemas	

i Note

There is no limit to the number of sources that may be used in an iteration rule.

Automatic rule generation

The iteration rule can be generated automatically. After you have created mappings for the columns under the selected target schema, click *Propose rule* in the Iteration Rule tab. The software generates the iteration rule tree. Always validate that the generated iteration rule matches your requirements. Modify the rule as needed, and add the ON condition expression when appropriate.

> → Remember
>
> Automatic rule generation is a best-guess function. For example, the software cannot know the ON condition, or whether to use INNER JOIN or LEFT OUTER JOIN. Use the automatic rule generation as a guide and always verify that the iteration rule that it creates fits your needs.

5.5.9 Construct an Input Request for a Web Service Call

You can create one row using the row generation transform to construct an input request for a web service call.

When calling a web service, an input request is always required. If the web service function expects an input with constant values only, you can use the row generation transform to construct the input message and map it with the schema created in the XML Map transform. A typical data flow is as follows:

Follow the steps below to construct an input request for a web service call:

1. In the data flow editor, drag the row generation transform onto the canvas and open the transform.

   The *Row count* is set at 1 by default. In this case, the value in the *Row count* option determines how many times the web service function will be called at run time.

   The *Row number starts at* option can be left as default, as the value in the row does not affect anything in this case.
2. Connect the row generation transform with the XML Map transform where you have built the nested structure for the web service call.
3. Open the XML Map transform and select the output schema.
4. In the *Transform Details*, in the *Iteration Rule* tab, click the plus icon and select *Create rule expression*.
5. Select the row generation transform you just defined and click *OK*.

5.5.10 Custom ABAP Transforms

Running custom ABAP transforms can extend SAP Cloud Integration for data services capabilities.

You can use custom ABAP transforms to incorporate ABAP functionality that is not available in the ABAP Query and ABAP Aggregation transforms. For example, when working with logical databases that are not supported in the product, you can use custom ABAP transforms to extract data. Custom ABAP transforms may also be useful to optimize generated code.

To create an ABAP transform, you create a separate ABAP FORM and map it to the ABAP transform.

> **!** Restriction
>
> You should have extensive knowledge about using ABAP before you create custom ABAP transforms in SAP Cloud Integration for data services.

1. Define a Custom ABAP Transform [page 194]
   The Custom ABAP transform uses ABAP programs you have created.
2. Create an ABAP FORM [page 195]
   A custom ABAP transform uses an ABAP FORM as the source for an ABAP program.
3. Define Parameters [page 198]
   Create input parameters in the custom ABAP transform that will be used in the ABAP FORM.
4. Define the Output Schema [page 199]
   Define the Output schema which is consumed by SAP Cloud Integration for data services when the ABAP program is run.

Related Information

5.5.10.1 Define a Custom ABAP Transform

The Custom ABAP transform uses ABAP programs you have created.

To define the details of a Custom ABAP transform:

1. If the data flow editor is not already open, open it.
2. Drag the *Custom ABAP* transform from the object palette onto your canvas.
3. Connect the *Custom ABAP* object to the *ABAP endpoint*.
4. Double-click the *Custom ABAP* object to view and edit the transform details.
5. In the transform details, in the *ABAP Options* tab, provide the following information:
   a. In the *ABAP Language File Name* field, type the path to the ABAP file on the server where the Data Services Agent is running. Your custom ABAP FORM will be stored here when you have completed and saved it in the editor.

> **Note**
>
> Ensure that the path you choose is included in the list of file directories configured for access by the Agent. This list can be found in the *Configure Directories* tab of the Agent Configuration UI.

b. Edit the *ABAP Job Name* and *ABAP Program Name* or accept the defaults.

The default for both fields is Z`<data flow name>`.

6. (Optional) Define ABAP parameters to be able to pass global variables to embedded data flows.

Global variables cannot be passed directly to the ABAP program. Instead, parameters are mapped to the global variables and can be used to pass dates or other information into the custom ABAP program.

Task overview: Custom ABAP Transforms [page 194]

Next task: Create an ABAP FORM [page 195]

Related Information

Data Flow Management [page 165]
Transform Types [page 172]
Open the Data Flow Editor [page 171]

5.5.10.2 Create an ABAP FORM

A custom ABAP transform uses an ABAP FORM as the source for an ABAP program.

Before you create a custom ABAP transform, you create an ABAP FORM that contains ABAP statements. The ABAP FORM must load data into an output schema defined for the custom ABAP transform.

> **Note**
>
> You can also define and pass parameters to the custom ABAP transform.

1. Select the *ABAP FORM Editor* button.
2. Create or edit an ABAP FORM. Do one of the following:

Action	Procedure
Create a custom ABAP FORM	Use the given template in the *ABAP FORM Editor*
Use an existing ABAP FORM	Copy and paste the contents from a text editor into the *ABAP FORM Editor*

3. Save your ABAP FORM. Do one of the following:

Action	Description
Deselct the checkbox at the bottom of the *ABAP FORM Editor*	Saves changes in the UI repository
Select the checkbox at the bottom of the *ABAP FORM Editor*	Saves and overwrites changes to the agent system to the location defined in the *ABAP Language File Name* field

The data flow calls the version of the ABAP program that is saved to the agent system.

4. Click *OK*.

   Your changes have been saved to the UI repository or the ABAP language file. You can continue to make changes to your output schema, parameters, or global variables or proceed to run the ABAP program.

Task overview: Custom ABAP Transforms [page 194]

Previous task: Define a Custom ABAP Transform [page 194]

Next task: Define Parameters [page 198]

Related Information

ABAP FORM Requirements [page 196]
ABAP FORM Rules [page 198]

5.5.10.2.1 ABAP FORM Requirements

Include special keywords and syntax in your ABAP FORM so that SAP Cloud Integration for data services recognizes the various parts of the FORM.

Use special text and syntax when you create the ABAP FORM

Create an ABAP FORM in the *ABAP FORM editor* and save it with the extension `.aba`. To enable SAP Cloud Integration for data services to recognize the ABAP FORM block in the data flow, use the keyword and syntax as shown in the following table. Type the keyword in upper case as shown.

Keyword	Syntax
FORMNAME	<<< >>>

Use ENDFORM to close the FORMNAME block.

Sample code outline:

```
FORM <<<FORMNAME>>>.
…..
ENDFORM.
```

SAP Cloud Integration for data services finds <<<FORMNAME>>> and replaces it with a unique FORM name that it uses to execute the ABAP.

Include an ITAB in the FORM to contain SAP Cloud Integration for data services output

Place the table information inside the ITAB in the ABAP FORM block. Use a special tag and syntax so that SAP Cloud Integration for data services recognizes it. Use the keyword and syntax as shown in the following table. Type the keyword in upper case as shown.

Keyword	Syntax
OTAB1	<<< >>>

SAP Cloud Integration for data services finds the <<<OTAB1>>> internal table and knows where to put output data from the SAP application. End the OTAB1 tag with the same keyword and syntax.

Sample code outline:

```
FORM <<<FORMNAME>>>.
...
    <<<OTAB1>>>
    ...
    <<<OTAB1>>>
ENDFORM.
```

Include parameters to map global variables

Global variables cannot be passed directly to the ABAP program. Instead, parameters are mapped to the global variables and can be used to pass dates or other information into the custom ABAP program.

SAP Cloud Integration for data services uses the defined *Name* and *Mapped Global Variable* in the ATL generation. The *ABAP Parameter Name* is used in the ABAP FORM.

Sample code outline:

```
FORM <<<FORMNAME>>>.
...
    $PARAM3
    ...
ENDFORM.
```

5.5.10.2.2 ABAP FORM Rules

Follow specific SAP Cloud Integration for data services and ABAP rules when you create an ABAP FORM to avoid processing errors.

- Follow all ABAP rules for writing ABAP statements.
- Start the FORM with the following tag, written exactly as shown: **<<<FORMNAME>>>**.
- End the FORM with the following tag, written exactly as shown: **ENDFORM**.
- Use the ITAB internal table name with the following tags, written exactly as shown: **<<<OTAB1>>>**.
- Create the FORM in a text editor. Save the FORM file to the Job Server directory. Use the ABAP extension `.aba`.
- Maximum line size in the FORM is 71 characters.

The following example is a basic code for the contents of an ABAP FORM. The table name is **MARA**.

```
FORM <<<FORMNAME>>>.
TABLES: MARA.
SELECT * FROM MARA.
   <<<OTAB1>>>-MATNR = MARA-MATNR.
   APPEND <<<OTAB1>>>.
ENDSELECT.
ENDFORM.
```

5.5.10.3 Define Parameters

Create input parameters in the custom ABAP transform that will be used in the ABAP FORM.

Input parameters are mapped to your pre-defined global variables and are used to pass the global variables to the embedded data flow. Use the steps below to create a local parameter that can be used in all of the ABAP transform details.

1. In the *Name* column, enter an easy-to-understand name that helps you identify the purpose of the ABAP parameter.

   This name appears only in the Parameters table.
2. The *ABAP Parameter Name* column contains a pre-defined name for an ABAP parameter. Use this name in the custom ABAP form.

   > **i Note**
   >
   > It is highly recommended that you do not delete a parameter as the ABAP parameter names will automatically readjust in numerical order. In the event that you do delete a parameter, be sure to manually adjust the parameter names in your ABAP FORM.
3. (Optional) Add a *Description* for the parameter.

   This text is for your own reference and appears only in the Parameters table.
4. Select a *Mapped Global Variable* to be associated with the parameter.
5. Repeat steps 1-4 to add as many parameters as needed in your ABAP FORM.

To use global variables in an embedded data flow, you must use the local parameter name in all references to the global variable the transform details.

Task overview: Custom ABAP Transforms [page 194]

Previous task: Create an ABAP FORM [page 195]

Next task: Define the Output Schema [page 199]

5.5.10.4 Define the Output Schema

Define the Output schema which is consumed by SAP Cloud Integration for data services when the ABAP program is run.

The Custom ABAP transform type displays only the Output pane. The ABAP FORM provides the source information (input).

To create an Output column:

1. Click the icon under the *Actions* column of the Output table.
2. Enter the *Name* of the output column.
3. Select a *Data Type*.
4. (Optional) Add a *Description*.

Task overview: Custom ABAP Transforms [page 194]

Previous task: Define Parameters [page 198]

5.5.11 Generate and Load an ABAP Program

Generate an ABAP program to review a report and choose to load and the generated ABAP program to an SAP Application.

- Your data flow editor must contain an embedded R/3 data flow in order to generate and run an ABAP program.
- The default configuration of the source datastore must have the ABAP execution option *Generate and execute* selected.
- To load an ABAP program to a SAP application, the RFC user of the datastore default configuration requires authorization to generate the report and assign it to a transport.

You can generate an ABAP report in the data flow editor that can be used to view, fine-tune, and edit your ABAP program. Once the ABAP program is ready, you can choose to load it to an SAP Application defined in the *Upload Attributes* section of the datastore configuration.

> **i** Note
>
> The *Generate and view ABAP report* execution uses the default datastore configuration. In the default datastore configuration, ensure that the *ABAP execution option* is set to *Generate & Execute*.

1. Select *Generate and view ABAP report* from the icons listed above the data flow editor.
2. Do one of the following:
   - To generate an ABAP report for review, select the agent and click *OK*.
   - To generate an ABAP report and load the ABAP program to an SAP application, select the agent, check *Deploy ABAP to server*, and click *OK*. When the task is run, the ABAP program is loaded to an SAP Application.

     > **i** Note
     >
     > The generation of an ABAP report can be influenced by source datastore options in the SAP Business Suite Application's subcategory, *Upload Attributes*. For more information, see SAP Business Suite Applications [page 87].

Related Information

Manually Uploading ABAP Programs to the SAP System

5.5.12 Enrich Data with the Lookup Function

You can use the lookup function to enrich your data with additional information.

The type of sources that you can use in the lookup function depends on the transform where the function is used.

- In the ABAP Query transform, you can use other tables from the source.
- In the Query transform, you can use file format or database datastores.

> **!** Restriction
>
> Cloud application datastores cannot be used as the lookup source. Additionally, if ABAP transforms are present in the data flow, non-ABAP transforms cannot use the SAP source as the lookup source.
>
> Also, using a file location object associated with a file format in the lookup function is not supported

For example, you might want to load data from an SAP system into a table in a cloud-based application, while converting an ID into a group name based on a mapping stored in a CSV file.

ID	GroupName
1001	A
1002	A
1003	B
1004	B
1005	C

Use the lookup function editor to construct a mapping that enriches your data.

Lookup from file formats

To enrich data with information from a file, specify the file format group, file format name, and file name containing the information to use as the `datastore_name`, `lookup_table`, and `file_name` parameters in the lookup function editor.

Lookup from database tables

To enrich data with information from a database table, specify the datastore and table name containing the information to use as the `datastore_name` and `table_name` parameters in the lookup function editor.

5.5.13 Group Hierarchical Data into Blocks

Use the XML Batch transform to group flat or hierarchical data sets into blocks before sending the result to the next transform. For example, to improve web service performance, you might want to send a data set to the web service target using groups of multiple rows per call instead of a single row per call.

> → Tip
>
> When working with flat data sets, consider using the GROUP BY capabilities of another transform. While XML Batch can process flat data sets, the output is always hierarchical.

When you use the XML Batch transform, you cannot manually create mappings between the input and output schemas. XML Batch supports a single input schema parent that is mapped as a child of the top level of the output schema. Use the options available in the *Details* tab to configure the transform.

Table 16: XML Batch options

Option	Description
Batch Size	Specifies the maximum size of rows for each batch. The value can be a positive integer or global variable. Default batch size: 10,000

Option	Description
Batch key columns	Optional. Specifies the input columns on which a given batch is constructed. When a column is selected, the column value is used to group rows into the batch. For each batch, rows are grouped up to the maximum batch size. Any additional rows are added to the next batch. To add a batch key column, drag only the first level key in the input schema to the batch key column field in the *Details* tab. When a batch key column is selected, the *Input already sorted by batch key columns* option is available. Selecting this option improves performance for data that has already been sorted by value in the selected column, and does not require additional sorting. **⚠ Caution** Select *Input already sorted by batch key columns* only when you are certain that the data is sorted. If there is unsorted data, the generated batches will be incorrect.

Related Information

Group Result by Columns [page 188]

5.5.14 Improve Transform Performance

Enhance performance by assigning a join rank to each source in your setup and by indicating whether to cache a source's data.

Join rank settings [page 203]
Enhance performance by assigning a join rank to each join in your setup.

Cache joins [page 209]
The join operation in a Query transform uses the cache settings from the source, unless you change the setting in the Query editor.

Related Information

5.5.14.1 Join rank settings

Enhance performance by assigning a join rank to each join in your setup.

When you rank each join, SAP Data Services considers the rank relative to other tables and files joined in the data flow. The optimizer, which is the optimization application inside the Data Services engine, joins sources with higher rank values before joining sources with lower rank values.

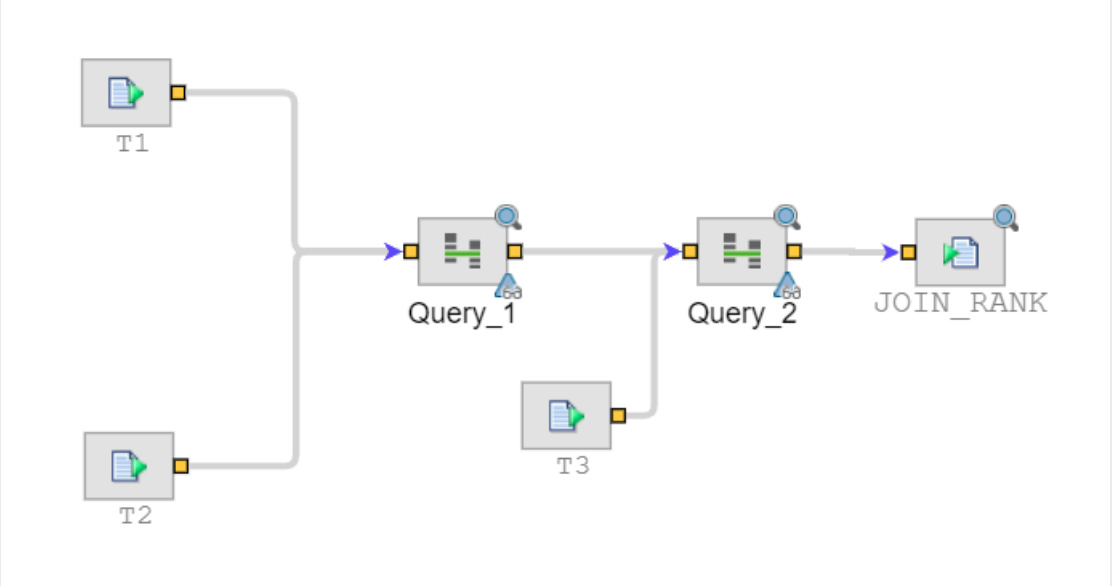
The order of execution depends on join rank and, for left outer joins, the order defined in the FROM clause. Setting the join rank for each join pair doesn't affect the result, but it can enhance performance by changing the order in which the optimizer performs the joins.

Set up joins in the Query transform. In a data flow that contains adjacent Query transforms, the ranking determination can be complex. The optimizer bases the way it joins your data in the following ways:

- The optimizer can combine the joins from consecutive Query transforms into a single Query transform, reassigning join ranks.
- The optimizer can consider the upstream join rank settings when it makes joins.

Example

In a data flow with multiple Query transforms with joins, we present four scenarios to demonstrate how the Data Services optimizer determines join order under different circumstances. The scenarios are based on the following data flow example:



- Query_1 contains an inner join between T1 and T2.
- Query_2 contains an inner join between the result of Query_1 and T3.

Related Information

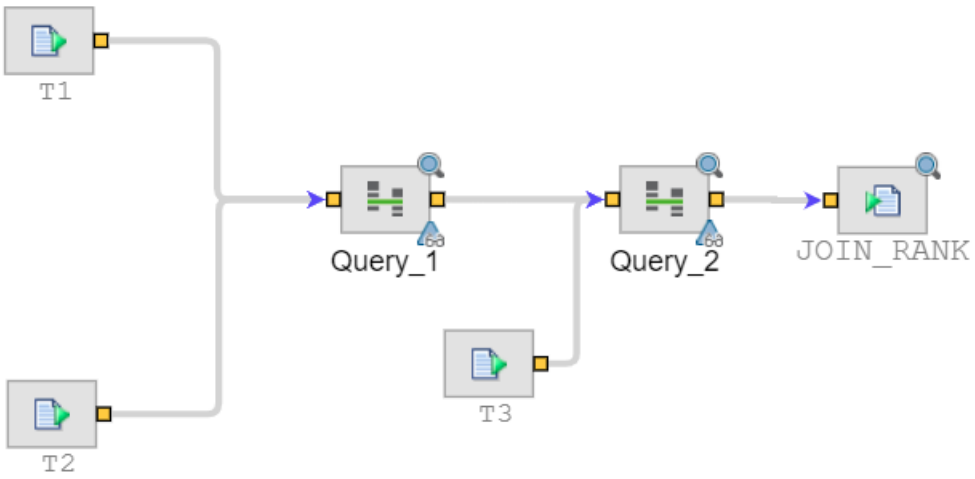
Scenario 1: All joins have join rank values [page 204]

Scenario 2: Query_2 join ranks not defined [page 205]
Scenario 3: T1 and T2 join ranks not defined [page 206]
Scenario 4: No joins have join rank values [page 207]
Set join rank values [page 207]

5.5.14.1.1 Scenario 1: All joins have join rank values

SAP Data Services determines the join ranks when all sources have join rank values.

Use the example in Join rank settings [page 203] for the following scenario.



The following table shows the join rank values for the joins in Query_1 and Query_2 as set in the data flow.

Table 17: Join ranks set in data flow

Query editor	Table	Join rank
Query_1	T1	30
	T2	40
Query_2	Query_1 result set	10
	T3	20

When the optimizer, which is the optimization application inside the Data Services engine, combines the joins in Query_2, it internally determines new join ranking based on the values in the original joins. The following table contains the join rank values determined by the optimizer for the combined joins in Query_2.

Table 18: Joins combined in Query_2

Query editor	Table	Join rank
Query_2	T1	30

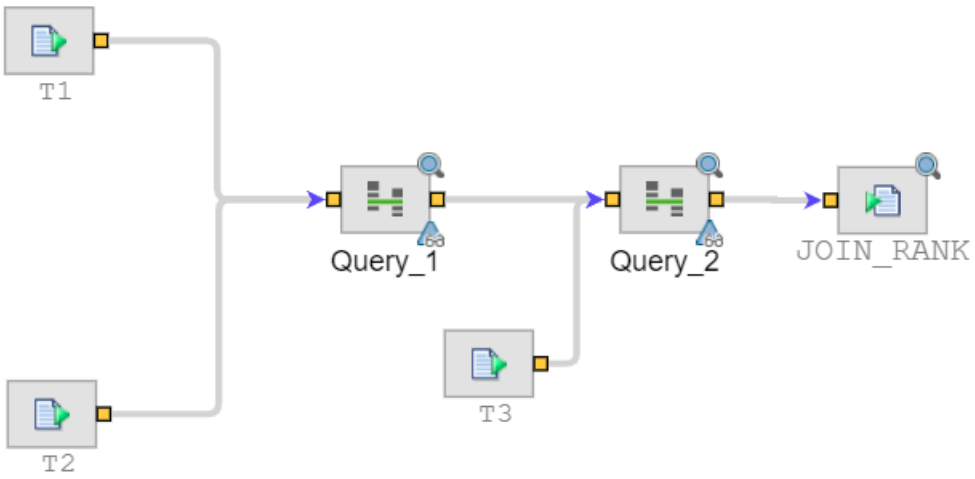
Query editor	Table	Join rank
	T2	40
	T3	41

Internally, the optimizer adjusts the join rank value for T3 from 20 to 41 because, in the data flow, Query_2 has a higher join rank value assigned to T3 than to "Query_1 result set."

5.5.14.1.2 Scenario 2: Query_2 join ranks not defined

SAP Data Services determines the join ranks when the sources in Query_2 aren't defined.

Use the example in Join rank settings [page 203] for the following scenario.



In this scenario, there are no settings for join ranks in Query_2. When you don't specify a join rank, Data Services uses the default of zero (0). Therefore, in Query_2, Data Services uses the join rank values of zero (0).

Table 19: Join ranks set in data flow

Query editor	Table	Join rank
Query_1	T1	30
	T2	40
Query_2	Query_1 result set	not set (default = 0)
	T3	not set (default = 0)

Internally, the optimizer, which is the optimization application inside the Data Services engine, assigns an internal join ranking in the combined joins in Query_2 as shown in the following table.

Table 20: Joins combined in Query_2

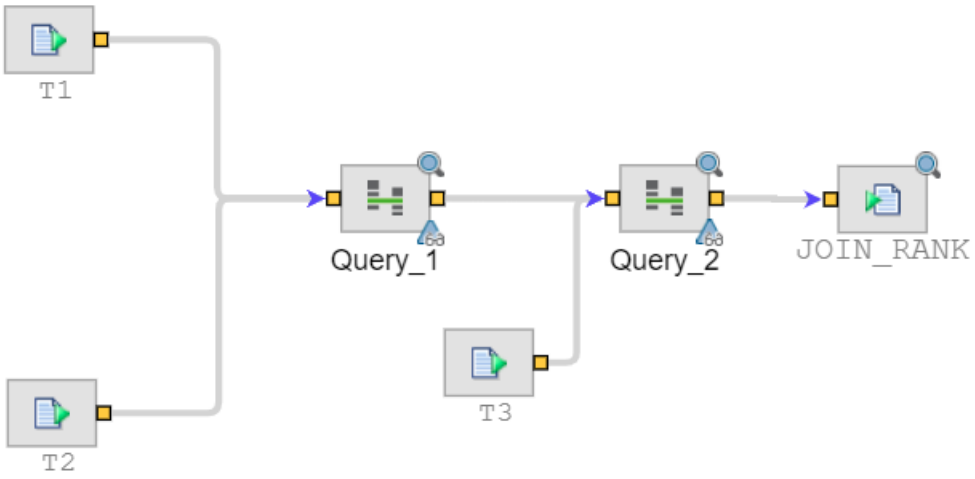
Query editor	Table	Join rank
Query_2	T1	30
	T2	40
	T3	40

You may be surprised to see a join rank value of 40 for T3. The optimizer considered that, even though "Query_1 result set" had a zero (0) join rank in the data flow, the result set consisted of sources that do have join ranks. The optimizer used the higher join rank from T1 and T2.

5.5.14.1.3 Scenario 3: T1 and T2 join ranks not defined

SAP Data Services determines the join ranks when there are no rank values set for the source tables T1 and T2.

Use the example in Join rank settings [page 203] with the following scenario.



In this scenario, there are no join ranks set for T1 and T2 source tables in Query_1. When there are no set join ranks, then the optimizer, which is the optimization application inside the Data Services engine, applies the default join rank of zero (0). The following table shows the Join rank values in the data flow, before the optimizer combines the joins into Query_2.

Table 21: Join ranks in data flow

Query editor	Table	Join rank
Query_1	T1	not set (default=0)
	T2	not set (default=0)
Query_2	Query_1 result set	10
	T3	20

Internally, the optimizer assigns a join rank of 10 for T1 and T2 because, in the data flow, the combined T1 and T2 tables, named “Query_1 result set,” has a join rank of 10.

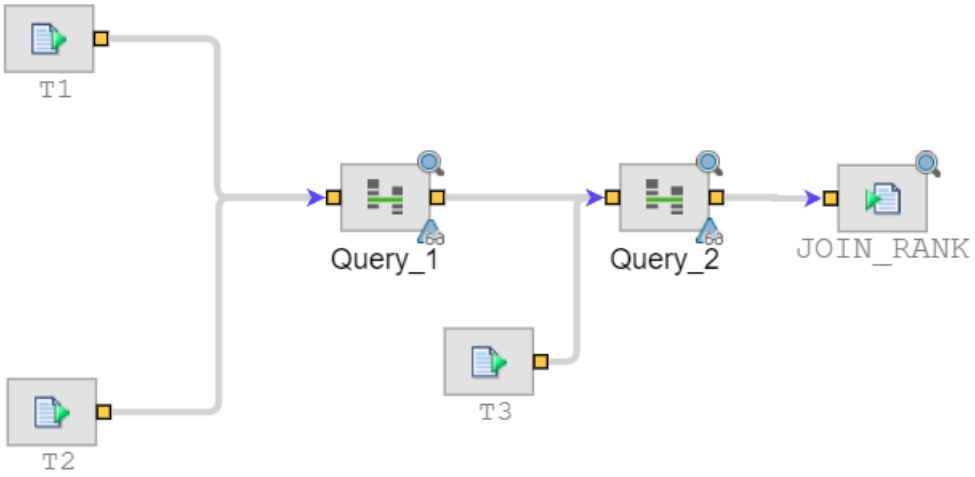
Table 22: Joins combined in Query_2

Query editor	Table	Join rank
Query_2	T1	10
	T2	10
	T3	20

5.5.14.1.4 Scenario 4: No joins have join rank values

SAP Data Services determines join ranks when there are no join rank values for any sources.

Use the example in Join rank settings [page 203] with the following scenario.



When you do not set join rank values in the data flow, the optimizer, which is the optimization application inside the engine, cannot optimize the joins. The optimizer uses the default setting of zero (0) for all tables in the joins.

5.5.14.1.5 Set join rank values

To increase the priority of tables or files in a join in relation to other sources, you can assign them a rank.

The system gives priority to tables and files with higher join rank values before considering sources with lower join ranks. A join rank defaults to zero unless changed.

For example, when you have the following tables with the indicated join rank...

Table Name	Join Rank
Table A	0
Table B	20
Table C	0
Table D	70

...the system processes the tables in the following order when performing the join:

Table and Assigned Join Rank	Priority
Table D with a join rank of 70	First
Table B with a join rank of 20	Second
Table A and Table C with the default join rank of 0	After higher-ranked sources based on performance optimization needs

To set a join rank for a source, follow these steps:

- If you're in the data flow editor, do the following:
  1. After dragging an input source into the editor, in the *Select Input* dialog box enter an integer higher than zero to increase the priority of this source in relation to others in the join, keeping in mind the ranks that may have been assigned to other sources in the join.
  2. (Optional) In the *Cache* field, you can choose *Yes* or *No* to indicate whether the system should cache the source data. For more details, see Cache joins [page 209].
- If you're in the Query transform, use one of the following methods:
  - On the *Options* tab:
    1. Select an input table or file.
    2. Navigate to the *Reader Options*, *File Options*, or *IBP Options* tab depending on the source with which you are working.
    3. In *Join Rank*, enter an integer higher than zero to increase the priority of this source in relation to others in the join, keeping in mind the ranks that may have been assigned to other sources in the join.
    4. (Optional) In the *Cache* field, you can choose *Yes* or *No* to indicate whether the system should cache the source data. For more details, see Cache joins [page 209].
    5. (Optional) To view the assigned join rank of each input table or file in the list, click on each one individually. Modify if needed.
    6. Close the window to save your changes.
  - On the *Join* tab:
    1. Double-click the *Join Rank* field of an input schema.
    2. Enter an integer higher than zero to increase the priority of this source in relation to others in the join, keeping in mind the ranks that may have been assigned to other sources in the join.
    3. (Optional) In the *Cache* field, you can can choose *Yes*, *No*, or *Automatic* to indicate whether the system should cache the source data. For more details, see Cache joins [page 209].
    4. Close the window to save your changes.

5.5.14.2 Cache joins

The join operation in a Query transform uses the cache settings from the source, unless you change the setting in the Query editor.

Cache settings in the source include the following:

- Cache is enabled or disabled.
- If enabled, the cache type: **Pageable** or **In Memory**.

In the Query editor, the cache setting is set to **Automatic** by default. The Automatic setting carries forward the cache settings from the source table.

When you configure joined sources in the Query transform, and you change the cache setting from **Automatic**, the cache setting in the Query transform overrides the setting in the source.

> **i** Note
>
> If any one input schema in the Query editor has a cache setting other than **Automatic**, the optimizer considers only the Query editor cache settings and ignores all source editor cache settings.

The following table shows the relationship between cache settings in the source and cache settings in the Query editor, and the effective cache setting for the join.

Cache Setting in Source	Cache Setting in Query Editor	Effective Cache Setting
Yes	Automatic	Yes
No	Automatic	No
Yes	Yes	Yes
No	Yes	Yes
Yes	No	No
No	No	No

> **i** Note
>
> For the best results when joining sources, we recommend that you define the join rank and cache settings in the Query editor.

The effect of cache setting on joins

In the Query editor, cache a source only when you use it as an inner source in a join.

If caching is enabled, and Data Services determines that data caching is possible, Data Services uses the source data in an inner join under the following conditions:

- The source is specified as the inner source of a left outer join.
- When using an inner join between the two tables, the source has a lower join rank.

Caching does not affect the order in which tables are joined.

If Data Services pushes down operations to the underlying database because of optimization conditions, it ignores the cache setting.

If a table becomes too large to fit in the cache, ensure that you set the cache type to **Pageable**.

Related Information

Save data to cache [page 210]
Join rank settings [page 203]

5.5.14.2.1 Save data to cache

Improve performance by having the system load a source's data to cache.

- If you're in the data flow editor and have dragged in an input source, choose *Yes* or *No* in the *Select Input* dialog box to indicate whether the system should cache the source data. For more details, see Cache joins [page 209].
- If you're in the Query transform, use one of the following methods:
  - On the *Options* tab:
    1. Select a source.
    2. Navigate to the *Reader Options*, *File Options*, or *IBP Options* tab depending on the source with which you are working.
    3. In the *Cache* field, choose *Yes* or *No* to indicate whether the system should cache the source data. For more details, see Cache joins [page 209].
    4. Close the window to save your changes.
  - On the *Join* tab:
    1. Double-click the *Join Rank* field of an input schema.
    2. In the *Cache* field, choose *Yes*, *No*, or *Automatic* to indicate whether the system should cache the source data. For more details, see Cache joins [page 209].
    3. Close the window to save your changes.

5.6 View Data During Data Flow Design and Debug

As you design or debug a data flow, at each transform step you can use the design-time data viewer to preview a sample of the input and output data that would be passed at that step in the data flow.

This allows you to compare the data before and after the transform acts on it to ensure that your design returns the results you expect.

The data flow does not need to be complete; however it must use a valid, accessible source that contains data.

1. If the data flow editor is not already open, open it.

2. View the design-time data in one of the following ways:

   - In the data flow editor, click the Design-time Data Viewer icon () in the lower right corner of a transform.
   - In the transform details, click *View Design-time Data*.

   > **! Restriction**
   >
   > You cannot view design-time data within the ABAP portion of a data flow. For additional restrictions, see SAP Notes.

3. In the dialog, accept the default settings for the design-time data viewer and global variables or change the configuration parameters to meet your needs.

   If you want to be able to download information such as logs and generated ATL file to use when debugging failed data views, select *Include debug information*.

4. Click *OK*.
   The viewer displays a subset of your data as it would be generated at that point in the data flow. If the data view fails and you have chosen to include debug information, you can click *Download Debug Information* to download a zip file.

5. Rerun the design-time data viewer as you continue to design or debug.

   As needed in the process, you can change the data viewer configuration settings from the action toolbar at the top of the data flow editor.

Task overview: Data Flows [page 163]

Related Information

Configure the Design-time Data Viewer [page 212]
What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]
Open the Data Flow Editor [page 171]

5.6.1 Configure the Design-time Data Viewer

The data viewer that is available from the data flow editor must be configured for each session (each time you log in). Changes to the default settings are not persistent. Global variable values may be defined on a task-by-task basis during a session.

1. If the data flow editor is not already open, open it.
2. From the action icons at the top of the data flow editor, click *Configure the Design-Time Data Viewer* ().
3. Select the agent you want to use.
4. (Optional) Choose to include debug information.

   If you include debug information and the data view fails, you can download a zip file containing logs and the generated ATL file.
5. In *Details*, accept the defaults or specify the following values:

Option	Description
System Configuration	A defined set of datastore configurations that are used together when the design-time data is retrieved.
Timeout (seconds)	The time at which the data viewer stops running if the data view is not complete. Default is 60 seconds.
Data Sample Size (rows)	Number of rows to read from the source. Default is 50. The maximum data sample size is 5,000 rows. SAP may modify this limit at any time without notice to prevent a decrease in performance. Any changed limit is reflected in an error message if a user exceeds the limit. **i Note** For customers using SAP Integrated Business Planning with a JBDC connection, the maximum is 500 rows.
Data Sample Frequency	Selects every nth row. For example, if the frequency is set to 3, then rows 1, 4, 7, 10 and so on are read from the source. Default is 1.

   Data sample size and sample frequency work together. For example, if you set the data sample frequency to 5 and the sample size to 10, then rows, 1, 6, 11, 16, 21, 26, 31, 36, 41, and 46 are retrieved from the source.
6. (Optional) Choose to specify values for global variables to be used in the current run only.

> **i Note**
>
> Values you specify for the current run are applicable only to the current task. In the same session, if you use the design-time data viewer for a data flow from a different task, you must specify the values for the current run for that task.

Related Information

Open the Data Flow Editor [page 171]
Create System Configurations [page 135]

5.7 Find Where a Data Flow is Used

A task or process cannot be deleted if its associated contents are in use. Find where a data flow is used by viewing its dependencies.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]
Find Where an Object is Used [page 136]

Find where a data flow is used in a task

View the dependencies of a data flow in the task editor.

1. In the *Project* tab, select a specific task.
2. Click *Edit* to get into the task editor.
3. Select a specific data flow and click *Actions* > *View where used* in the upper left corner to view the dependencies of the data flow.

Find where a data flow is used in a process

View the dependencies of a data flow in the process editor.

1. In the *Project* tab, select a specific process.
2. Click *Edit* to get into the process editor.

3. Select a specific data flow and click the where used icon () in the tool bar to view the dependencies of the data flow.

5.8 Loading into PGP-protected Target Files

In order to load data to a PGP-protected target file, the public key of the external third-party that will receive the file must be used to encrypt the source file.

Additionally, to encrypt a file with your digital signature to verify the authenticity of the data's origin and integrity, you must use your organization's public key.

As needed for your situation, from the Data Services Agent Configuration program, make sure that the following prerequisites are met:

Table 23: Prerequisites to encrypt a file to load to a target

	Prerequisite	More information
☐	You have received the public key of the external third-party that will receive the target.	Make sure to get the user ID of the key. The user ID can be an email address, name, or other identifying information.
☐	You have imported the external third-party public key.	Importing an External Public Key [page 217]

Additionally, to generate your digital signature, make sure you have met the following prerequisites:

Table 24: Prerequisites to generate a digital signature

	Prerequisite	More information
☐	A PGP key pair exists for your organization.	Generating a PGP Key Pair [page 218]
☐	The organization key pair is imported to the server hosting your agent.	If the key pair was not generated on the server hosting your agent, you must move it to the server. Moving your Organization Key Pair [page 219]
☐	You have exported your organization's public key.	Exporting your Public Key [page 220]
☐	You have sent your public key to the external third-party that owns the target.	

First use the Data Services Agent Configuration program to meet the prerequisites. Then, use the SAP Cloud Integration for data services user interface to create and run the task that creates the PGP-encrypted target file.

1. In the SAP Cloud Integration for data services user interface, create a task to load a target file.
2. Create a data flow. In the *Set Up* step, in the *Encrypt with PGP* field, select *yes* and type the user ID of the external third-party public key.
3. If you want to include a digital signature, in the *Include Digital Signature* field, select *yes*.

Next steps:

Validate and run the task as usual.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]

5.9 Reading from PGP-protected Source Files

In order to read and decrypt a PGP-protected source file, your organization's public key must be used to encrypt the source file.

Additionally, to decrypt a file which contains a digital signature to verify the authenticity of the data's origin and integrity, you must have the external (third-party) key from the owner of the source file.

As needed for your situation, from the Data Services Agent Configuration program, make sure that the following prerequisites are met:

Table 25: Prerequisites to decrypt a source file

	Prerequisite	More information
□	A PGP key pair exists for your organization.	Generating a PGP Key Pair [page 218]

	Prerequisite	More information
☐	The organization key pair is imported to the system hosting your agent.	If the key pair was generated on the system hosting your agent, you do not need to import it. If the key pair was generated on a different system in your organization, then you must move it to the system that hosts your agent. Moving your Organization Key Pair [page 219]
☐	The owner of the source file has your public key.	Export your public key and send it to the owner of the source file. Exporting your Public Key [page 220]
☐	The owner of the source file has encrypted the file using your public key.	

Additionally, if the source file contains a digital signature, make sure you have met the following prerequisites:

Table 26: Prerequisites to verify a digital signature

	Prerequisite	More information
☐	You have received the external (third-party) public key from the owner of the source file.	
☐	You have imported the external (third-party) public key to the system which hosts your agent.	Importing an External Public Key [page 217]

First use the Data Services Agent Configuration program to meet the prerequisites. Then, use the SAP Cloud Integration for data services user interface to create and run the task to read and decrypt the source file.

1. In the SAP Cloud Integration for data services user interface, create a task and data flow to read the encrypted source data.
2. In the data flow, select the transform that reads the source data.
3. In the *Transform Details* do the following:
   a. From the *File Options* tab, in the *Selected input information*, in the *PGP Protected* field, select *yes*.
   b. If the file contains a digital signature, in the *PGP Signature* field, select *yes*.

Validate and run the task as usual.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]

5.10 Importing an External Public Key

Import an external (third-party) public key to use when encrypting data you are loading to a file.

> **Note**
>
> The external (third-party) public key must be imported to the server hosting the SAP Data Services agent used in the task.

1. If the SAP Data Services Agent configuration program is not already running. start it.
   - On Windows platforms, run `configureAgent.bat`.
   - On Linux platforms, run `configureAgent.sh`.

   > **Note**
   >
   > You must run the configuration program from a user account that has administrative privileges. On Windows platforms that have User Account Control (UAC) enabled, you can also choose the *Run as administrator* option.

   By default, the configuration program is located in the directory where you installed the SAP Data Services Agent.
2. Click *Configure PGP*.
3. Click *Import an external (third-party) public key*.
4. Type or browse to the location of the external (third-party) public key.
5. Click *Apply*.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]

5.11 Generating a PGP Key Pair

Within an SAP Cloud Integration for data services organization, generate a single PGP key pair.

The key pair contains a public key and a private key. The organization public key can be sent to third-parties who can use it to encrypt data. SAP Cloud Integration for data services can decrypt the data using the organization private key.

1. If the SAP Data Services Agent configuration program is not already running. start it.
   - On Windows platforms, run `configureAgent.bat`.
   - On Linux platforms, run `configureAgent.sh`.

   > **i Note**
   >
   > You must run the configuration program from a user account that has administrative privileges. On Windows platforms that have User Account Control (UAC) enabled, you can also choose the *Run as administrator* option.

   By default, the configuration program is located in the directory where you installed the SAP Data Services Agent.
2. Click *Configure PGP*.
3. Click *Generate a key pair for your organization*.
   a. Select the key size, hash algorithm, and symmetric algorithm appropriate for your requirements.
   b. Enter a user ID.

      The user ID is the name bound to the public key. It can be an email address, name, or other identifying information.
4. Click *Apply*.

A PGP key pair is generated and saved to the host system where your SAP Data Services Agent is installed.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]

5.12 Moving your Organization Key Pair

If your organization has multiple agents, all agents must share the same key pair. The file containing the organization's PGP key pair must be stored locally on each system that hosts an SAP Data Services Agent.

A PGP key pair has been generated for the organization.

After the organization's key pair has been generated, it must be exported to a known location and then imported to each system which hosts an SAP Data Services Agent.

1. If the SAP Data Services Agent configuration program is not already running. start it.
   - On Windows platforms, run `configureAgent.bat`.
   - On Linux platforms, run `configureAgent.sh`.

   > **i Note**
   >
   > You must run the configuration program from a user account that has administrative privileges. On Windows platforms that have User Account Control (UAC) enabled, you can also choose the *Run as administrator* option.

   By default, the configuration program is located in the directory where you installed the SAP Data Services Agent.
2. Click *Configure PGP*.
3. Click *Export your organization's key pair*.

4. Type or browse to the desired location and type a passphrase.

   Take note of this information as it will be required later when you import the key pair.
5. Click *Apply*.
6. From a system which hosts a different SAP Data Services Agent, start the SAP Data Services Agent configuration program as described in Step 1.
7. Click *Import your organization's key pair*.
8. Enter the location and passphrase you created in Step 4 when you exported the key pair from the system where it was generated.
9. Click *Apply*.
10. Repeat steps 6 - 9 for each system which hosts an SAP Data Services Agent.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Exporting your Public Key [page 220]
Use Parallel Processing for Web Services [page 221]

5.13 Exporting your Public Key

Export your organization's public key so it can be used when encrypting the source data.

1. If the SAP Data Services Agent configuration program is not already running. start it.
   - On Windows platforms, run `configureAgent.bat`.
   - On Linux platforms, run `configureAgent.sh`.

> **Note**
>
> You must run the configuration program from a user account that has administrative privileges. On Windows platforms that have User Account Control (UAC) enabled, you can also choose the *Run as administrator* option.

By default, the configuration program is located in the directory where you installed the SAP Data Services Agent.

2. Click *Configure PGP*.
3. Click *Export your organization's public key*.
4. Type or browse to a location where your public key can be accessed as required.
5. Click *Apply*.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Use Parallel Processing for Web Services [page 221]

5.14 Use Parallel Processing for Web Services

By default, SAP Cloud Integration for data services calls web service functions on a row-by-row basis. However, for tasks that deal with a large number of records, this sequential operation can cause a performance bottleneck. While some web services can be called using a batch of records from the XML Batch transform, this is not always possible if the web service function does not support batch input.

By calling a web service function in parallel, you can reduce the performance bottleneck caused by row-by-row function calls.

To call a web service function with parallel processing, you must configure the degree of parallelism for the data flow, and enable parallel execution on the function itself.

1. Configure the data flow's degree of parallelism.
   a. Open the data flow editor.
   b. In the toolbar, click *View Options*.
   c. Specify the degree of parallelism.

      The degree of parallelism determines how many times the data flow can call the web service function simultaneously. For example, if you set the degree of parallelism to 4, the data flow can open 4 connections to the web service function at one time.

      The default degree of parallelism is 2.
2. Enable parallel execution for the web service function.
   a. In the *Datastores* tab, select the web service datastore that contains the function call that you want to modify.
   b. Click the *Tables* tab in the datastore editor.
   c. Select the web service function, and click *Properties*.
   d. Select *Enable parallel execution*.

Task overview: Data Flows [page 163]

Related Information

What is a Data Flow? [page 164]
Data Flow Management [page 165]
Data Flow Design [page 169]
Transform Types [page 172]
Transform Operations [page 174]
View Data During Data Flow Design and Debug [page 210]
Find Where a Data Flow is Used [page 213]
Loading into PGP-protected Target Files [page 214]
Reading from PGP-protected Source Files [page 215]
Importing an External Public Key [page 217]
Generating a PGP Key Pair [page 218]
Moving your Organization Key Pair [page 219]
Exporting your Public Key [page 220]
Open the Data Flow Editor [page 171]

6 Scripts and Functions

Scripts and functions allow you to manipulate and enrich the data within a data flow.

Scripts [page 223]
Scripts are single-use objects used to call functions and assign values to variables in a task or a process.

Set Global Variables [page 235]
Global variables are symbolic placeholders. When a task or process runs, these placeholders are populated with values. This allows users flexibility of run-time values used in extractions.

Functions [page 237]
Functions in SAP Cloud Integration for data services take input values and produce a return value if necessary. Input values can be parameters passed into a data flow, values from a column of data, or variables defined inside a script.

Related Information

6.1 Scripts

Scripts are single-use objects used to call functions and assign values to variables in a task or a process.

A script can contain the following statements:

- Function calls
- If statements
- While statements
- Assignment statements
- Operators

The basic rules for the syntax of the script are as follows:

- End each line with a semicolon (;).
- Start variable names with a dollar sign ($).
- Enclose string values in single quotation marks (').
- Start comments with a pound sign (#).
- Function calls always specify parameters even when the function uses no parameters.
- Use variables only when you declare them in the task or process that calls the script.

> **Example**
>
> The following script statement determines today's date and assigns the value to the variable `$TODAY`:
>
> ```
> $TODAY = sysdate();
> ```

Parent topic: Scripts and Functions [page 223]

Related Information

Data Services scripting language [page 224]
Set Global Variables [page 235]
Functions [page 237]

6.1.1 Data Services scripting language

Use the Data Services scripting language to write scripts, apply built-in functions, and to write expressions. Note that the Data Service Scripting Language supported by SAP Cloud Integration for data services is a subset of that used by SAP Data Services. Refer to the list of supported functions shown in the Related Information section.

Write expressions such as complex column mapping expressions and WHERE clause conditions.

Related Information

Using the scripting language [page 224]
Language syntax [page 225]
Data Services scripting language [page 235]
List of functions [page 240]

6.1.1.1 Using the scripting language

In SAP Cloud Integration for data services, you can use the scripting language in two locations.

- In the *Process Editor*, you can create a script.
- In the *Execution Properties* at the task level, you can make edits in preload and postload scripts as needed to meet your business specifications.

6.1.1.2 Language syntax

When you use the scripting language, adhere to specific syntax so the objects you are building function correctly.

Use the syntax from the scripting language in expressions as well as in scripts. With the scripting language, assign values to variables, call functions, and use standard string and mathematical operators. Ensure that you know the proper syntax for statements, columns, table references, strings, variables, and so on.

Related Information

Syntax for statements in scripts [page 225]
Syntax for column and table references in expressions [page 225]
Strings [page 226]
Variables [page 227]
Variable interpolation [page 228]
SAP scripting language operators [page 228]
NULL values [page 230]
SAP scripting language keywords [page 232]

6.1.1.2.1 Syntax for statements in scripts

Statements in scripts are detailed steps written in the flow of logic.

Ensure that you use the following syntax for statements:

- End each statement in a script with a semicolon (;).
- Begin each comment line with a pound sign (#).

6.1.1.2.2 Syntax for column and table references in expressions

The SAP Cloud Integration for data services scripting language recognizes column and table names without special syntax.

Expressions are a combination of constants, operators, functions, and variables that evaluate to a value of a given data type. Use expressions inside script statements or add them to data flow objects. Because expressions can be used inside data flow objects, they can contain column names.

No special syntax is required for column or table names. For example, you can indicate the `start_date` column as the input to a function as follows:

```
to_char(start_date, 'dd.mm.yyyy')
```

Before you include a column name, ensure that it is a part of the input schema of the query.

6.1.1.2.3 Strings

String syntax includes using quotation marks, escape characters, and trailing blanks.

- Quotation marks: Choose the type of quotation mark to use based on whether you use identifiers or constants.
- Escape characters: Some characters need escape characters to differentiate the character use in the string.
- Trailing blanks: Remove trailing blanks with built-in functions.

Related Information

Quotation marks [page 226]
Escape characters [page 227]
Trailing blanks [page 227]

6.1.1.2.3.1 Quotation marks

The type of quotation marks to use in strings depends on whether you are using identifiers or constants.

The following table describes the types of quotation marks to use for each string type.

Identifier	Name of an object such as table, column, data flow, or function. Identifiers need quotation marks when they contain special characters that are non alphanumeric. **Example** Use a double quotation for the following string because it contains blanks: `"compute large numbers"`
Constant	A fixed value used in computation. There are two types of constants: • String constants require single quotation marks. For example: `'Hello World'` or `'1995.01.23'`. • Numeric constants do not require quotation marks. For example, `2.14`.

6.1.1.2.3.2 Escape characters

Constants that contain single quotes, backslashes, or other special characters use escape characters so that the function knows how to process them.

When your script uses a syntax character that is not intended as syntax, precede the character with an escape character.

SAP Cloud Integration for data services uses the backslash (\) as the escape character.

Character	Example
Multiple single quotation marks.	`'World\'s Books'`
Multiple backslashes.	`'C:\\temp'`

6.1.1.2.3.3 Trailing blanks

SAP Cloud Integration for data services does not strip trailing blanks from strings that are used in scripts.

To remove the trailing blanks from strings, use the built-in functions rtrim or rtrim_blank.

Related Information

rtrim [page 353]
rtrim_blanks [page 354]

6.1.1.2.4 Variables

Precede variable names with a dollar sign ($).

- You define global variables used in a script or expression in a task or a process. Edit or add Global variables when editing data transformation under *Transform Details* or in *Execution Properties*.
- Use the following statement to ensure that the function passes the return value outside the function:

  ```
  RETURN(<expression>)
  ```

  `<expression>` contains the definition of the value to be returned.

6.1.1.2.5 Variable interpolation

Embed expressions within constant strings using the correct syntax so that the software correctly evaluates the variables.

When you embed expressions within constant strings, the software evaluates the variables and substitutes the value into the string. The software does not need the concatenation operator (||) to make the substitution.

The following expression uses the concatenation operator:

```
print('The value of the start date
    is:'||sysdate()+5);
```

Simplify the expression as follows excluding the concatenation operator:

```
print('The value of the start date
    is:[sysdate()+5]');
```

Use curly braces ({}) and square brackets ([]) to enclose the embedded expressions:

- The square brackets ([]) indicate to substitute the value of the expression.
- The curly braces ({}) indicate to add single quotation marks to the value of the expression.

Strings that include curly braces or square brackets cause processing errors. Avoid the errors by preceding the braces or brackets with a backslash (\).

```
* Any software coding and/or code snippets are examples. They are not for
productive use. The example code is only intended to better explain and visualize
the syntax and phrasing rules. SAP does not warrant the correctness and
completeness of the example code. SAP shall not be liable for errors or damages
caused by the use of example code unless damages have been caused by SAP's gross
negligence or willful misconduct.
```

6.1.1.2.6 SAP scripting language operators

Operators act like functions but the are symbols that specify the action the function takes.

The following table contains descriptions of the operators that you use in scripts and expressions. The table lists the operators in order of precedence.

> **i** Note
>
> When the software pushes operations to a DBMS, the DBMS determines the precedence based on DBMS rules.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication

Operator	Description
/	Division
=	Assignment, comparison
<	Comparison, less than
<=	Comparison, less than or equal to
>	Comparison, greater than
>=	Comparison, greater than or equal to
!=	Comparison, not equal to
||	Concatenate
%	Return the remainder when one number is divided by another
AND	Logical AND
OR	Logical OR
NOT	Logical NOT
IS NULL	Comparison, is a NULL value
IS NOT NULL	Comparison, is not a NULL value
LIKE	Comparison, matches a specific character string with a specified pattern. Use the following wildcards with LIKE: • %: A string of zero or more characters. • _ (underscore): A single character. • []: A single character with a specific range or set. • [^]: A single character not within the specified range or set. **Note** LIKE does not support the function character '[' inside a rang. For example, '\[\[\]%' .
NOT LIKE	Comparison, excludes rows that match the LIKE criterion.

Use a comparison operator in the following ways:

- In a script or script function as a condition; for example:

```
if ($x IN (1,2,3)), while ($x IN (1,2,3)) and ifthenelse()
```

- In a data flow such as in a WHERE clause ifthenelse() function, case transform, etc.
- As a condition of the IF block, WHILE block or TRY CATCH block

The following examples illustrate valid comparison expression syntax:

```
expression = expression
expression != expression
expression < expression
expression > expression
expression <= expression
expression >= expression
expression IS NULL
```

```
expression IS NOT NULL
expression IN (expression list)
expression IN domain
expression LIKE constant
expression NOT LIKE constant
```

`NOT` (any of the valid comparisons); for example `NOT ($x IN (1,2,3))`

```
comparison OR comparison
comparison AND comparison
```

The following syntax is not valid:

```
$x NOT IN (1,2,3)
EXIST or NOT EXIST
```

6.1.1.2.7 NULL values

Indicate NULL values using the keyword NULL.

For example, you can check whether a column (COLX) is null or not:

```
COLX IS NULL
COLX IS NOT NULL
```

The software does not check for NULL values in data columns. Use the function `nvl` to remove NULL values.

Related Information

NULL values and empty strings [page 230]

nvl [page 339]

6.1.1.2.7.1 NULL values and empty strings

The software has specific rules for syntax with `NULL` values and empty strings.

There are the two rules that regulate empty strings:

- Assigning an empty string to a variable of type varchar.
  - The software treats the value of the variable as a zero length string.
  - If you assign an empty string to a variable that is not a varchar, the software issues an error.
  - Assign a NULL value to a variable of any type using the NULL constant.
- When an empty string is a constant:
  - The software treats an empty string that is a constant (' ') as a varchar value of zero length.
  - Use a NULL constant for the NULL value.

> **Note**
>
> Oracle does not distinguish an empty string from a NULL value. When you insert an empty string or a NULL value into a varchar column, Oracle treats both the empty string and NULL value as NULL values. Therefore, the software treats the value as a NULL value.

NULLs and empty strings in conditionals

There are three rules for NULLs and empty strings in conditionals:

- Equals and Not Equal to comparison operators always evaluate to FALSE against a NULL value.
- Use the IS NULL and IS NOT NULL operators to test the presence of NULL values.
- When comparing two variables, always test for NULL.

Equals (=) and Not Equal (<>) evaluate to FALSE against NULL

The FALSE result includes comparing a variable that has a value of NULL against a NULL constant.

The following table shows the comparison results for the variable assignments `$var1 = NULL` and `$var2 = NULL`:

Condition	Translates to	Returns
`If (NULL = NULL)`	NULL is equal to NULL	FALSE
`If (NULL != NULL)`	NULL is not equal to NULL	FALSE
`If (NULL = '')`	NULL is equal to empty string	FALSE
`If (NULL != '')`	NULL is not equal to empty string	FALSE
`If ('bbb' = NULL)`	bbb is equal to NULL	FALSE
`If ('bbb' != NULL)`	bbb is not equal to NULL	FALSE
`If ('bbb' = '' )`	bbb is equal to empty string	FALSE
`If ('bbb' != '')`	bbb is not equal to empty string	TRUE
`If ($var1 = NULL )`	NULL is equal to NULL	FALSE
`If ($var != NULL)`	NULL is not equal to NULL	FALSE
`If ($var1 = '' )`	NULL is equal to empty string	FALSE
`If ($var != '')`	NULL is not equal to empty string	FALSE
`If ($var1 = $var2 )`	NULL is equal to NULL	FALSE
`If ($var != $var2)`	NULL is not equal to NULL	FALSE

The following table shows the comparison results for the variable assignments `$var1 = ''` and `$var2 = ''`:

Condition	Translates to	Returns
`If ($var1 = NULL)`	Empty string is equal to NULL	FALSE

Condition	Translates to	Returns
`If ($var != NULL)`	`Empty string` is not equal to `NULL`	FALSE
`If ($var1 = '')`	`Empty string` is equal to empty string	TRUE
`If ($var != '')`	`Empty string` is not equal to empty string	FALSE
`If ($var1 = $var2)`	`Empty string` is equal to `Empty string`	TRUE
`If ($var != $var2)`	`Empty string` is not equal to `Empty string`	FALSE

Use the IS NULL and IS NOT NULL operators

Use the IS NULL and IS NOT NULL operators to test the presence of null values. For example, assuming a variable is assigned: `$var1 = NULL;`

Condition	Translates to	Returns
`If ('bbb' IS NULL`	`bbb` is `NULL`	FALSE
`If ('bbb' IS NOT NULL)`	`bbb` is not `NULL`	TRUE
`If ('' IS NULL`	Empty string is `NULL`	FALSE
`If ('' IS NOT NULL)`	Empty string is not `NULL`	TRUE
`If ($var1 IS NULL)`	`NULL` is `NULL`	TRUE
`If ($var IS NOT NULL)`	`NULL` is not `NULL`	FALSE

When comparing two variables, always test for NULL

In this scenario, you are not testing a variable with a value of `NULL` against a `NULL` constant as in the first rule. Either test each variable and branch accordingly or test in the conditional as shown in the second row of the following table.

Condition	Recommendation
`if($var1 = $var2)`	Do not compare without explicitly testing for `NULL`s. Using this logic is not recommended because any relational comparison to a `NULL` value returns `FALSE`.
`if ( (($var1 IS NULL) AND ($var2 IS NULL)) OR ($var1 = $var2))`	Executes the `TRUE` branch if both `$var1` and `$var2` are `NULL`, or if neither are `NULL` but are equal to each other.

6.1.1.2.8 SAP scripting language keywords

Keywords are select words in the scripting language that you use in expressions based on syntax rules and desired behavior.

Related Information

BEGIN [page 233]
CATCH [page 233]
ELSE [page 234]
END [page 234]
IF [page 234]
RETURN [page 234]
TRY [page 235]
WHILE [page 235]

6.1.1.2.8.1 BEGIN

The keyword BEGIN indicates the beginning of the code that becomes the function, script, or other construct.

The software automatically adds BEGIN and END statements to function, transform, and script definitions.

6.1.1.2.8.2 CATCH

The keyword CATCH indicates a catch for a try—catch block.

If an error occurs while executing any of the statements between the TRY and the CATCH statements, the software executes the statements defined by the CATCH. Use the CATCH keyword as shown in the following script, or use `CATCH(ALL)`.

```
BEGIN
TRY
    BEGIN
        <script_step>;
        <script_step>;
    END
CATCH (<exception_number>)
    BEGIN
        <catch_step>;
        <catch_step>;
    END
CATCH (<exception_number>)
    BEGIN
        <catch_step>;
        <catch_step>;
    END
END
```

6.1.1.2.8.3 ELSE

The keyword ELSE defines the second branch for an IF statement.

If there is no ELSE following an IF statement, the software takes no action if the condition is not met.

6.1.1.2.8.4 END

The keyword END indicates the end of the code that becomes the function, script, or other construct.

The software automatically adds BEGIN and END statements to function, transform, and script definitions.

6.1.1.2.8.5 IF

The keyword IF indicates a conditional step in the code.

Construct an IF statement with or without an ELSE step. Use the IF keyword as follows:

```
IF (<condition>) <script_step>; ELSE <script_step>;
```

or

```
IF (<condition>) <script_step>;
```

where `<condition>` is an expression that evaluates to True or False. `<script_step>` indicates the set of instructions to execute based on the value of `<condition>` . If `<script_step>` contains more than one statement, enclose these statements in BEGIN and END statements.

6.1.1.2.8.6 RETURN

The keyword RETURN indicates the value to be returned by a function.

Use the RETURN keyword as follows:

```
RETURN (<expression>);
```

where `<expression>` defines the value to be returned.

6.1.1.2.8.7 TRY

The keyword TRY indicates the start of a try—catch block.

Related Information

CATCH [page 233]

6.1.1.2.8.8 WHILE

The keyword WHILE defines a set of statements to execute until a condition evaluates to FALSE.

Use the WHILE keyword as follows:

```
WHILE (<condition>) <script_step>;
```

where `<condition>` is an expression that evaluates to True or False. `<script_step>` indicates the set of instructions to execute based on the value of `<condition>`. If `<script_step>` contains more than one statement, enclose each statement in BEGIN and END statements.

6.1.1.3 Data Services scripting language

Use the SAP Cloud Integration for data services scripting language to write scripts, apply built-in functions, and to write expressions.

Write expressions such as complex column mapping expressions and WHERE clause conditions.

6.2 Set Global Variables

Global variables are symbolic placeholders. When a task or process runs, these placeholders are populated with values. This allows users flexibility of run-time values used in extractions.

The table below shows some of the global variables that are available to you in SAP Cloud Integration for data services. For a full list as well as more information about their use, see the topics within the Global Variables section of the SAP Integrated Business Planning for Supply Chain documentation.

All global variables are applied at the task/process level. You can edit global variables both on the task/process level in the *Execution Properties* tab and on the data flow level inside a transform on the *Global Variables* panel. Global variables apply across all data flows within a task or process.

> **Note**
>
> Certain global variables are used by the application to process the data after it is loaded. For example, SAP Integrated Business Planning requires `$G_PLAN_AREA`, `$G_SCENARIO`, `$G_TIME_PROFILE`, and `$G_BATCH_COMMAND`. If the global variables are not included in the task or process, an error is returned.

Global Variables	Data Type	Description
$G_IBP_FILTER_ID	varchar(100)	Planning filter ID. To use planning filters defined in the Planning Filters SAP IBP app, add this global variable to your task and leave the Value field empty.
$G_IBP_SKIP_UNCHANGED_DATA	varchar(1)	When enabled, the system performs post-processing only on data that has changed. Skipping unchanged data provides faster run-time for jobs with a heavy data load that contain a considerable amount of unchanged data. Valid values are **X** to enable the variable, or blank. **Note** $G_IBP_SKIP_UNCHANGED_DATA is supported only for WebSocket RFC connections.
$G_IBP_USER_ID	varchar(50)	Business user ID. Authorizations assigned to the given user ID are applied to the selection in SAP IBP.
$G_BATCH_COMMAND	char(50)	By default, uses "Insert_Update".('INSERT_UPDATE','DELETE' or 'REPLACE')
$G_TIME_PROFILE_LEVEL	int	By default, uses "-1". When left empty, defaults to the base / storage level of the planning area (-1).
$G_PLAN_AREA	varchar	Planning area. Required only for version-specific Master Data upload.
$G_SCENARIO	varchar	Planning area version. (Scenario by its old name) required only for version-specific Master Data upload.

> **Note**
>
> For WebSocket RFC data flows, only two global variables are supported.

Depending on your requirements and environment, allow the default values or set values in one of the following locations:

Option	Description
Run Now **dialog box**	From the *Projects* tab, select a task or process. From the *Actions* menu, select *Run Now*.
Execution Properties **tab of a task or process**	Use for loading data. From the *Projects* tab, select a task or process. From the *Actions* menu, select *Edit*. Click the *Execution Properties* tab.
Preload or postload script	Use for testing and loading data.

Task overview: Scripts and Functions [page 223]

Related Information

Scripts [page 223]
Functions [page 237]

6.3 Functions

Functions in SAP Cloud Integration for data services take input values and produce a return value if necessary. Input values can be parameters passed into a data flow, values from a column of data, or variables defined inside a script.

Functions compared with transforms [page 238]
Some functions can produce the same or similar values as transforms. However, functions and transforms operate in a different scope.

Operation of a function [page 238]
The type of function determines where you can use the function. The function operation determines where you can call the function.

Arithmetic in date functions [page 239]
The software performs some implicit data type conversions on date, time, datetime, and interval values.

Including functions in expressions [page 240]
Use a function in an expression only when the function makes sense in the expression that you create.

List of functions [page 240]
Use this list of functions as a starting point to see descriptions that include input parameters, required syntax, return values, and data types.

Parent topic: Scripts and Functions [page 223]

Related Information

Scripts [page 223]
Set Global Variables [page 235]

6.3.1 Functions compared with transforms

Some functions can produce the same or similar values as transforms. However, functions and transforms operate in a different scope.

- Functions operate on single values, such as values in specific columns in a data set.
- Transforms operate on data sets, creating, updating, and deleting rows of data.

Parent topic: Functions [page 237]

Related Information

Operation of a function [page 238]
Arithmetic in date functions [page 239]
Including functions in expressions [page 240]
List of functions [page 240]

6.3.2 Operation of a function

The type of function determines where you can use the function. The function operation determines where you can call the function.

For example, a lookup database function operates as an iterative function. The lookup function caches information about the table and columns on which it operates between function calls.

By contrast, conversion functions, such as to_char, operate as stateless functions. Conversion functions operate independently in each iteration.

Aggregate functions, such as max, require a set of values with which to operate. You cannot call the lookup function (iterative) or the max function (aggregate) from a script or conditional where the context does not support how these functions operate.

The function type determines where you can use a function. The following table describes each type of function and where you can call it from.

Type	Description
Aggregate	Generates a single value from a set of values. Aggregate functions, such as max, min, and count, use the data set specified by the expression in the *Group By* tab of a query. Call an aggregate function only from within a Query transform. You cannot call an aggregate function from custom functions or scripts.
Iterative	Maintains state information from one invocation to another. An iterative function, such as the lookup function, contains state information that lasts only until you execute the query in which you use the function. Call an iterative function only from within a Query transform. You cannot call an iterative function from other functions or scripts.
Stateless	Does not maintain state information from one invocation to the next. Use stateless functions, such as to_char or month, anywhere you can use expressions.

Parent topic: Functions [page 237]

Related Information

Functions compared with transforms [page 238]
Arithmetic in date functions [page 239]
Including functions in expressions [page 240]
List of functions [page 240]

6.3.3 Arithmetic in date functions

The software performs some implicit data type conversions on date, time, datetime, and interval values.

Parent topic: Functions [page 237]

Related Information

Functions compared with transforms [page 238]
Operation of a function [page 238]
Including functions in expressions [page 240]
List of functions [page 240]

6.3.4 Including functions in expressions

Use a function in an expression only when the function makes sense in the expression that you create.

You can use functions in the following objects:

- Transforms (Query , Case, SQL, Map_Operation)
- Script objects
- Conditionals

Before you use a function, ensure that the function operation makes sense in the expression you are creating.

For example:

- You cannot use the max function in a script or conditional where there is no collection of values on which to operate.
- Parameters can be output by a task or a process but not by a data flow.

You can use the function editor to define the parameters for an existing function. The function editor is embedded in the data flow editor.

Parent topic: Functions [page 237]

Related Information

Functions compared with transforms [page 238]
Operation of a function [page 238]
Arithmetic in date functions [page 239]
List of functions [page 240]

6.3.5 List of functions

Use this list of functions as a starting point to see descriptions that include input parameters, required syntax, return values, and data types.

SAP Cloud Integration for data services supports the functions listed below. Custom functions are not available.

Use the decode function to return an expression based on the first condition in the specified list of conditions and expressions that evaluates to TRUE.

decrypt_aes [page 269]
Use the decrypt_aes function to decrypt the input string with the user-specified pass phrase and key length using the AES algorithm.

decrypt_aes_ext [page 271]
Use the decrypt_aes_ext function to decrypt the input string with the user-specified passphrase, salt, and key length using the AES algorithm.

encrypt_aes [page 272]
Use the encrypt_aes function to encrypt the input string using the specified passphrase and key length with the AES algorithm.

encrypt_aes_ext [page 273]
Use the encrypt_aes_ext function to encrypt an input string using the specified passphrase, salt, and key length with the AES algorithm.

exec [page 274]
Use the exec function to send a command to the operating system.

file_copy [page 279]
Use the file_copy function to copy an existing file to a different location using the same file name or a different file name.

file_delete [page 281]
Use the file_delete function to delete an existing file, or delete a group of files indicated by a wildcard (*).

file_exists [page 282]
Use the file_exists function to see if a file or directory exists.

file_move [page 283]
Use the file_move function to move an existing file or group of files to a different location using the same file name or a different file name.

fiscal_day [page 285]
Use the fiscal_day function to convert a date into an integer value that represents a day in a fiscal year.

floor [page 286]
Use the floor function to return the largest integer value equal to or less than a number.

gen_row_num [page 287]
Use the gen_row_num function to return an integer value beginning with 1, then incremented sequentially by 1 for each additional call.

gen_row_num_by_group [page 288]
Use the gen_row_num_by_group function to generate a column of row identification numbers for each ID group in the specified column.

gen_uuid [page 290]
Use the gen_uuid function to generate a unique identifier.

get_data [page 290]
Retrieves stored data that contains the task name and the most current load date.

greatest [page 291]
Use the greatest function to return the greatest of the list of one or more expressions.

Use the month function to determine the month in which the given date falls.

nvl [page 339]
Use the nvl function to replace NULL values with a given value.

power [page 340]
Use the power function to return the value of the given expression to the specified power.

previous_row_value [page 341]
Use the previous_row_value function to return the column value of the previous row.

print [page 342]
Use the print function to print a given string to the trace log.

quarter [page 343]
Use the quarter function to determine the quarter in which the given date falls.

raise_exception [page 344]
Use the raise_exception function to generate an exception message for the Job Server error log..

raise_exception_ext [page 345]
Use the raise_exception_ext function generates an exception with an exit code.

rand [page 346]
Use the rand function to return a random number between 0 and 1.

rand_ext [page 346]
Use the rand_ext to return a random number between 0 inclusive and 1 exclusive.

replace_substr [page 347]
Use the replace_substr function to replace each occurrence of a specified substring with a different substring.

replace_substr_ext [page 348]
Use the replace_substr_ext function to replace each occurrence of a specified substring with a replacement string. The specified substring can contain hexadecimals that refer to a UNICODE character, or non printable character references such as form feed or new line.

round [page 350]
Use the round function to round a given number to a specified precision.

rpad [page 351]
Use the rpad function to pad a string of characters from a given pattern.

rpad_ext [page 352]
Use the rpad_ext function to pad a string with logical characters from a given pattern.

rtrim [page 353]
Use the rtrim function to remove specified characters from the end of a string.

rtrim_blanks [page 354]
Use the rtrim_blanks function to remove blank characters from the end of a string.

rtrim_blanks_ext [page 355]
Use the rtrim_blanks_ext function to remove blank and control characters from the end of a string.

save_data [page 356]
Use the save_data function to create and store a persistent variable with a name, which could be the task name or any other string, and any piece of data. This data could be the end date timestamp of the most current load.

week_in_year [page 378]
Use the week_in_year function to return the week in the year in which the given date falls.

word [page 380]
Use the word function to return one word out of a given string.

word_ext [page 381]
Use the word_ext function to return a word that you identify by a position in a delimited string.

year [page 382]
Use the year function to determine the year in which the given date falls.

Parent topic: Functions [page 237]

Related Information

Functions compared with transforms [page 238]
Operation of a function [page 238]
Arithmetic in date functions [page 239]
Including functions in expressions [page 240]

6.3.5.1 abs

Use the abs function to return the absolute value of a number. The absolute value (sometimes known as the modulus) of a number is the value of a number without regard to its sign – it can also be thought of as the distance of a number from zero.

Syntax

```
abs(<num>)
```

Return value

decimal, double, int, or real

The absolute value of the given number, `<num>`. The type of the return value is the same as the type of the original number.

Where

`<num>`	The source number.

Example

Function	Results
`abs(12.12345)`	`12.12345`
`abs(-12.12345)`	`12.12345`

6.3.5.2 add_months

Use add_months to add a given number of months to a date.

Syntax

```
add_months(<original_date>,<months_to_add>)
```

Return value

date

Where

`<original_date>`	Specify the starting year.month.date.
`<months_to_add>`	Number of months to add to the original date.

Details

The `<months_to_add>` can be any integer. If `<original_date>` is the last day of the month or if the resulting month has fewer days than the day component of `<original_date>`, then the result is the last day of the resulting month. Otherwise, the result has the same day component as `<original_date>`.

Example

Function	Results
`add_months('1990.12.17', 1)`	`'1991.01.17'`
`add_months('2001.10.31', 4)`	`'2002.2.28'`

6.3.5.3 ascii

Use the ascii function to return a decimal value of an ASCII code of the first character in the input string.

Syntax

Syntax

```
ascii(<input_string>)
```

Return Value

Int

Where

`<input_string>`	The source string.

Details

Returns the decimal value of the ASCII code of the first character in the input string. Returns -1 if the first character is not a valid ASCII character.

Example

Function	Results
`ascii('AaC')`	65

6.3.5.4 avg

Use the avg function to calculate the average of a given set of values.

Syntax

```
avg(<value_list>)
```

Return value

decimal, double, int, or real

The calculated average of `<value_list>` . The function calculates the average to the same precision as the input value.

Where

`<value_list>`	The source values for which to calculate an average, such as values in a table column.

6.3.5.5 cast

Use the cast function to explicitly convert an expression of one data type to another.

Syntax

```
Cast('<expression>','<data_type>')
```

Return Value

Returns the same value as in data_type.

Where

`<expression>`	Input expression to be cast to target data type.
`<data_type>`	Target data type that is a built-in data type and specified as a constant string. For example, 'decimal`(28,7)`'.

Details

The cast function explicitly converts the value of the first parameter into the built-in data type that you specify in the second parameter. The following table shows all explicit data type conversions that are valid for this function.

Cast type compatibility matrix

From / To	Date	Date time	Decimal	Double	Int	Interval	Real	Time	Time stamp	Varchar
Date	X	X							X	X
Date time	X	X						X	X	X
Decimal			X	X	X	X	X			X
Double			X	X	X	X	X			X
Int			X	X	X	X	X			X
Interval			X	X	X	X	X			X
Real			X	X	X	X	X			X
Time		X						X	X	X
Time stamp	X	X						X	X	X
Varchar	X	X	X	X	X	X	X	X	X	X

The following table contains syntax for the given data_type.

Target data type syntax

Data type	Syntax
varchar	'varchar(length)'
decimal	'decimal(precision,scale)'
integer	'int'
real	'real'
double	'double'
timestamp	'timestamp'
datetime	'datetime'
date	'date'
time	'time'
interval	'interval'

The following table shows the date&time format for the cast() function:

Timestamp	yyyy.mm.dd hh24:mi:ss.ff
Datetime	yyyy.mm.dd hh24:mi:ss
Date	yyyy.mm.dd
Time	hh24:mi:ss

Example

Input	Output
`cast('20.3','decimal(3,1)')`	20.3

6.3.5.6 chr

Use the chr function to convert a decimal ASCII code to a character.

Syntax

```
chr (<integer_expression>)
```

Return Value

ASCII character

Where

`<integer_expression>`	Integer from 0 through 255. Returns NULL if the integer expression is not in this range.

Details

This function returns the character associated with the specified ASCII code decimal number. If you specify a value of less than 0 or greater than 255 for the integer_expression parameter, the software returns NULL. Use chr to insert control characters into character strings. For example, `chr(9)` can be used to insert <tab>.

Example

Function	Results
`chr(65)`	'A'

6.3.5.7 ceil

Use the ceil function to return the smallest integer value greater than or equal to a number.

Syntax

```
ceil(<num>)
```

Return value

decimal, double, int, or real

The indicated integer, cast as the same type as the original number, `<num>`.

Where

<num>	The source number.

Example

Function	Results
ceil(12.12345)	13.00000
ceil(12)	12
ceil(-12.223)	-12.000

6.3.5.8 concat_date_time

Use the concat_date_time function to return a datetime from separate date and time inputs.

Syntax

```
concat_date_time(<date>,<time>)
```

Where

<date>	Date input value.
<time>	Time input value.

Return value

datetime

The datetime value obtained by combining the inputs.

Example

```
concat_date_time(MS40."date",MS40."time")
```

6.3.5.9 count

Use the count function to count the number of values in a group.

count

Syntax

```
count(<column>)
```

Return value

int

The number of rows in the column that have a non-NULL value.

Where

`<column>`	The column in the input table in which to count values.

Example

6.3.5.10 count_distinct

Use the count_distinct function to return the number of distinct non NULL values in a group.

Syntax

```
count_distinct(<expression>)
```

Return Value

Integer

Where

`<expression>`	Any valid expression of any type except NRDM or long data type.

Input

Name	Region	Country
Cust 1	East	US
Cust 2	East	US
Cust 3	West	US
Cust 4	East	France

Output

```
count_distinct(REGION) = 2
```

To calculate the number of distinct regions per country, add the country column to the group by clause, as follows:

`count_distinct(REGION)`	Country
2	US
1	France

6.3.5.11 current_configuration

Use the current_configuration function to return the name of the datastore configuration that the software uses at runtime.

If the datastore does not support multiple configurations, for example, the datastore is a memory datastore, the function returns the name of the datastore instead.

Syntax

```
current_configuration(ds_name)
```

Return Value

varchar

Where

`<ds_name>`	The name you enter when you create the datastore.

Example

Create a task or process and add a script with, for example, the following line.

```
print('Datastore Configuration used at runtime: [current_configuration()]')
```

Returns, for example, the following to the trace log:

```
Datastore configuration used at runtime: Test_DS
```

6.3.5.12 current_system_configuration

Use the current_system_configuration function to return the name of the system configuration the software uses at runtime.

If there is no system configuration defined, the function returns a NULL value.

Syntax

```
current_system_configuration()
```

Return Value

varchar

Example

Create a task or process and add a script with, for example, the following line:

```
print('System Configuration used at runtime: [current_system_configuration()]')
```

This line returns, for example, the following to the trace log:

```
System configuration used at runtime: Production
```

6.3.5.13 date_diff

Use the date_diff function to return the difference between two dates or times.

Syntax

```
date_diff(<date1>,<date2>,'<fmt_str>')
```

Return Value

int

Where

<date1, date2>	The dates between which the function determines the difference.	
<fmt_str>	The string that describes the format of the dates. Choose from the following values:	
	D	Day
	H	Hours
	M	Minutes
	S	Seconds
	MM	Months
	YY	Years

Details

If date1 is smaller than date2, the date_diff function returns a positive value. To cause the function to return only a positive value, surround the function with the abs() function.

> **Note**
>
> When you use the sysdate function with date_diff, be aware that the value the sysdate function returns is datetime. Internally Data Services reads both the date and the time when it runs a sysdate function. The data that is used by the job depends on the data type of a particular column. For example, if the data type of a column in a query is date, Data Services uses only the date for calculations. It ignores the time data. If you change the data type to datetime, Data Services uses both a date and a time. If the data type is datetime and you don't want to use the time data, use the to_char function to truncate the timestamp from sysdate.

> **Example**

Function	Results
`date_diff(start_date,sysdate(),'D')`	The number of days between the date in the column start_date and the current date.
`date_diff(start_time,systime(),'M')`	The number of minutes between the time in the column start_time and the current time.

6.3.5.14 date_part

Use the date_part function to extract a component of a given date.

> **Syntax**

```
date_part(<in_date>,'<fmt_str>')
```

Return Value

int

Where

`<in_date>`	The input date.

<fmt_str>	The string describing the format of the extracted part of the date. Choose from the following values:	
	YY	Year
	MM	Month
	DD	Day
	HH	Hours
	MI	Minutes
	SS	Seconds

Details

This function takes in a datetime and extracts the component requested as an integer.

> **Note**
>
> The function displays year as four digits, not two.

> **Example**

Function	Results
`date_part('1990.12.31', 'YY')`	`1990`
`date_part('1991.01.17 23:44:30', 'SS')`	`30`

6.3.5.15 day_in_month

Use the day_in_month function to determine the day in the month on which the input date falls.

> **Syntax**

```
day_in_month(<date1>)
```

Return value

int

The number from 1 to 31 that represents the day in the month that `<date1>` occurs.

Where

`<date1>`	The source date.

This function extracts the day component from the date value.

Example

Function	Results
`day_in_month(to_date('Jan 22, 1997','mon dd, yyyy'))`	22
`day_in_month(to_date('02/29/1996','mm/dd/yyyy'))`	29
`day_in_month(to_date('1996.12.31','yyyy.mm.dd'))`	31

6.3.5.16 day_in_week

Use the day_in_week function to determine the day in the week on which the input date falls.

Syntax

```
day_in_week(<date1>)
```

Return value

int

The number from 1 (Monday) to 7 (Sunday) that represents the day in the week that `<date1>` occurs.

Where

`<date1>`	The source date.

This function allows you to categorize dates according to the day of the week the date falls on. For example, all dates for which this function returns a "3" occur on Wednesday.

Example

Function	Results
`day_in_week(to_date('Jan 22, 1997','mon dd, yyyy'))`	3 (Wednesday)
`day_in_week(to_date('02/29/1996','mm/dd/yyyy'))`	4 (Thursday)
`day_in_week(to_date('1996.12.31','yyyy.mm.dd'))`	2 (Tuesday)

6.3.5.17 day_in_year

Use the day_in_year function to determine the day in the year on which the input date falls.

Syntax

```
day_in_year(<date1>)
```

Return value

int

The number from 1 to 366 that represents the day in the year that `<date1>` occurs.

Where

`<date1>`	The source date.

Example

Function	Results
`day_in_year(to_date('Jan 22, 1997','mon dd, yyyy'))`	22
`day_in_year(to_date('02/29/1996','mm/dd/yyyy'))`	60
`day_in_year(to_date('1996.12.31','yyyy.mm.dd'))`	366 (1996 was a leap year.)

6.3.5.18 db_database_name

Use the db_database_name function to return the database name of the datastore configuration in use at runtime.

Syntax

```
db_database_name(<ds_name>)
```

Return Value

varchar

Where

`<ds_name>`	The datastore name you enter when you create the datastore.

Details

This function is useful if your datastore has multiple configurations and is accessing an MS SQL Server or SAP ASE database. For a datastore configuration that is using either of these database types, you enter a database name, when you create a datastore. This function returns that database name.

For example, master is a database name that exists in every Microsoft SQL Server and SAP ASE database. However, if you use different database names, you can use this function in, for example, a SQL statement instead of using a constant. Using the function in a SQL statement allows the SQL statement to use the correct database name for each run no matter what datastore configuration is in use.

This function returns an empty string for datastore configurations without MS SQL Server or SAP ASE as the Database Type.

Example

If you have a SQL transform that performs a function that is written differently for different versions of database types, you can tell the system which text to use for each database version. In this example, the sql() function is used within a script.

```
IF (db_type('sales_ds') = 'DB2')
    $sql_text = '…';
ELSE
BEGIN
    IF (db_type('sales_ds') = 'MicroSoft_SQL_Server')
        $db_name = db_database_name('sales_ds');
        $sql_text = '…';
END
Sql('sales_ds', '{$sql_text}');
```

6.3.5.19 db_owner

Use the db_owner function to return the real owner name for the datastore configuration that is in use at runtime.

Syntax

```
db_owner(<ds_name>, <alias_name>)
```

Return Value

varchar

Where

`ds_name`	The datastore name that you entered when you created the datastore.

`alias_name`	The name of the alias that you created in the datastore, then mapped to the real owner name when you created a datastore configuration.

Details

This function is useful if your datastore has multiple configurations because with multiple configurations, you can use alias owner names instead of database owner names. By using aliases instead of real owner names, you limit the amount of time it takes to port tasks to different environments.

For example, you can use this function in a SQL statement instead of using a constant. This allows the SQL statement to use the correct database owner for each run no matter what datastore configuration is in use.

Example

```
$real_owner = db_owner('sales_ds', 'sales_person');
```

6.3.5.20 db_type

Use the db_type function to return the database type of the datastore configuration in use at runtime.

Syntax

```
db_type(<ds_name>)
```

Return Value

varchar

Possible db_type() return values for datastore types are as follows:

Datastore Types	Possible db_type() Return Value
Adapter	Adapter
Database	DB2, Microsoft_SQL_Server, Oracle, SAP, SAP_BW, SAP Sybase (for SAP ASE), Sybase_IQ
SAP Applications	SAP
SAP BW Source	SAP

Datastore Types	Possible db_type() Return Value
SAP BW Target	SAP_BW

Where

<ds_name>	The datastore name you enter when you create the datastore.

Details

This function is useful if your datastore has multiple configurations. For example, you can use this function in a SQL statement instead of using a constant. Using the function in a SQL statement allows the SQL statement to use the correct database type for each run no matter what datastore configuration is in use.

Example

If you have a SQL transform that performs a function that you have to write differently for database types, you can tell the system what to do if the database type is Oracle.

In this example, the sql() function is used within a script.

```
IF (db_type('sales_ds') = 'Oracle')
BEGIN
    IF (db_version('sales_ds') = 'Oracle 9i')
        $sql_text = '…';
    ELSE
           $sql_text = '…';
END
Sql('sales_ds', '{$sql_text}');
```

6.3.5.21 db_version

Use the db_version function to return the database version of the datastore configuration in use at runtime.

Syntax

```
db_version(<ds_name>)
```

Return Value

varchar

Possible db_version() return values are:

Database type	Version
Oracle	Currently supported versions
Microsoft SQL Server	Currently supported versions
DB2 UDB	Currently supported versions
SAP ASE	Currently supported versions
SAP Sybase IQ	Currently supported versions
""	An empty string is returned for any other database type

Where

<ds_name>	The datastore name you enter when you create the datastore.

Details

This function is useful if your datastore has multiple configurations. For example, you can use this function in a SQL statement instead of using a constant. Using the function in a SQL statement allows the SQL statement to use the correct database version for each run no matter what datastore configuration is in use.

Example

If you have a SQL transform that performs a function that is written differently for different versions of Oracle, you can tell the system which text to use for each database version. In this example, the sql() function is used within a script.

```
IF (db_type('sales_ds') = 'Oracle')
BEGIN
    IF (db_version('sales_ds') = 'Oracle 9i')
        $sql_text = '…';
    ELSE
           $sql_text = '…';
END
Sql('sales_ds', '{$sql_text}');
```

6.3.5.22 decode

Use the `decode` function to return an expression based on the first condition in the specified list of conditions and expressions that evaluates to TRUE.

Syntax

```
decode(<condition_and_expression_list>,<default_expression>)
```

Return value

`<expression>` or `<default_expression>`

Returns the value associated with the first `<condition>` that evaluates to TRUE. The data type of the return value is the data type of the first `<expression>` in the `<condition_and_expression_list>`.

If the data type of any subsequent `<expression>` or the `<default_expression>` is not convertible to the data type of the first `<expression>`, SAP Cloud Integration for data services produces an error at validation. If the data types are convertible but do not match, a warning appears at validation.

Where

`<condition_and_expression_list>`	A comma-separated list of one or more pairs that specify a variable number of conditions. Each pair contains one `<condition>` and one `<expression>` separated by a comma. Specify at least one `<condition>` and `<expression>` pair. The `<condition>` evaluates to TRUE or FALSE. If the `<condition>` evaluates to TRUE, the `<expression>` is the value that the function returns.
`<default_expression>`	An expression that the function returns if none of the conditions in `<condition_and_expression_list>` evaluate to TRUE. A `<default_expression>` is required.

Details

The `decode` function provides an easier way to write nested `ifthenelse` functions. In nested `ifthenelse` functions, you write nested conditions and ensure that the parentheses are in the correct places, as the following example shows:

Example

```
ifthenelse ((EMPNO = 1), '111',
```

```
  ifthenelse((EMPNO = 2), '222',
      ifthenelse((EMPNO = 3), '333',
        ifthenelse((EMPNO = 4), '444',
          'NO_ID'))))
```

In the `decode` function, you list the conditions, as the following example shows. Therefore, `decode` is less error prone than nested `ifthenelse` functions.

Example

```
decode ((EMPNO = 1), '111',
(EMPNO = 2), '222',
(EMPNO = 3), '333',
(EMPNO = 4), '444',
'NO_ID')
```

To improve performance, SAP Cloud Integration for data services pushes this function to the database server when possible. Thus, the database server, rather than SAP Cloud Integration for data services, evaluates the `decode` function.

Use this function to apply multiple conditions when you map columns or select columns in a query. For more flexible control over conditions in a script, use the IF keyword in the scripting language.

If a condition compares a varchar value with trailing blanks, the `decode` function ignores the trailing blanks.

To compare a NULL value (NULL constant or variable that contains a NULL constant), use the IS NULL or IS NOT NULL operator. If you use the Equal (=) or Not equal to (<>) operator, the comparison against a NULL value always evaluates to FALSE.

Example

Function	Results
`decode((COUNTRY = 'FRANCE'), 'French', (COUNTRY = 'GERMANY'), 'German', (COUNTRY = 'ITALY'), 'Italian', (COUNTRY = 'USA'), 'America', (COUNTRY IS NULL), 'Unknown', 'Others')`	If the value in the COUNTRY column is `FRANCE`, the value returned is `French`. If COUNTRY is `NULL`, the value returned is `Unknown`. If COUNTRY does not contain any of the values listed, the `decode` function returns the value `Others`.

6.3.5.23 decrypt_aes

Use the decrypt_aes function to decrypt the input string with the user-specified pass phrase and key length using the AES algorithm.

Note

The decrypt_aes function is intended to decrypt data that was encrypted by encrypt_aes function.

Syntax

```
decrypt_aes(<encrypted_input_string>,<passphrase>,<key_length_in_bits>)
```

Return value

Returns plain string as varchar.

In case of a failure, the function throws an exception of type execution error, which results in termination of the job. You can catch the exception by using try/catch handlers.

If the encrypted input string is empty, then the return value is an empty string.

If the encrypted input string is NULL, then the return value is NULL.

Where

`<encryptedinput_string>`	A varchar input string to be decrypted.
`<passphrase>`	A varchar character string.
`<key_length_in_bits>`	An int value of 128, 192, or 256.

Example

For security purposes, secure the passphrase in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

```
 #read the passphrase
from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
encrypt_aes(SOURCE.SSN,
$G_passphrase, 128);
```

Similar to other string functions, this function can be called from a custom function, in the column mapping of a Query transform, or in a script in the work flow.

Related Information

encrypt_aes [page 272]

6.3.5.24 decrypt_aes_ext

Use the decrypt_aes_ext function to decrypt the input string with the user-specified passphrase, salt, and key length using the AES algorithm.

Ensure that the passphrase and salt are the same as the passphrase and salt used to encrypt the data.

The function generates an AES key of the specified key length using the specified passphrase and the key generation algorithm PKCS5_PBKDF2_SHA256. This key is used for decrypting the encrypted input string.

Syntax

```
decrypt_aes_ext(<Varchar Encrypted_input_string>,<Varchar
Passphrase>,<Varchar Salt>,<Int Key_length_in_bits>)
```

Return value

Returns plain string as varchar.

In case of a failure, the function throws an exception of type execution error, which results in the termination of the job. You can catch the exception by using try/catch handlers.

If the encrypted input string is empty, then the return value is an empty string.

If the encrypted input string is NULL, then the return value is NULL.

If you fail to provide the same passphrase and key length used for encryption to this function, then the call does not fail but instead returns an incorrect output.

Where

`<Encrypted_input_string>`	A varchar input string to be decrypted.
`<Passphrase>`	A varchar character string with at least one character.
`<Salt>`	A varchar that must be exactly eight ASCII characters.
`<Key_length_in_bits>`	An int value of 128, 192, or 256.

Example

For security purposes, secure the passphrase and salts in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

```
#read the passphrase from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
$G_salt = sql('PASSWORD_DATASTORE', 'select SALT from PASSWORD');
```

```
decrypt_aes_ext(ENCRYPTED.SSN, $G_passphrase, $G_salt, 128);
```

Similar to other string functions, call this function from a custom function, in the column mapping of a Query transform, or in a script in the work flow.

6.3.5.25 encrypt_aes

Use the encrypt_aes function to encrypt the input string using the specified passphrase and key length with the AES algorithm.

> **Note**
>
> Do not decrypt data that you encrypted within Data Services using the encrypt_aes function outside of Data Services. Instead, use the decrypt_aes function to decrypt this data.

> **Syntax**

```
encrypt_aes(<input_string>,<passphrase>,<key_length_in_bits>)
```

Return value

Returns encrypted string as varchar. The size of the encrypted string is about twice as large as the size of plain text. Therefore, ensure that you have enough space to hold the encrypted string.

In case of a failure, the function throws an execution error and terminates the job. You can catch the exception by using try/catch handlers.

If the input string is empty, then the function returns an encrypted string. The encrypted string is different for multiple calls of the encrypt_aes() function with an empty input string.

If the input string is NULL, then the return value is NULL.

Where

`<input_string>`	A varchar input string to be encrypted.
`<passphrase>`	A varchar character string.
`<key_length_in_bits>`	An int value of 128, 192, or 256.

Details

For security purposes, secure the passphrase in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

Example

```
 #read the passphrase
from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
encrypt_aes(SOURCE.SSN,
$G_passphrase, 128);
```

Like other string functions, you can call the encrypt_aes function from a custom function, in the column mapping of a Query transform, or in a script in the work flow.

6.3.5.26 encrypt_aes_ext

Use the encrypt_aes_ext function to encrypt an input string using the specified passphrase, salt, and key length with the AES algorithm.

Syntax

```
encrypt_aes_ext(<Varchar Input_string>,<Varchar Passphrase>,<Varchar
salt>,<Int Key_length_in_bits>)
```

Return value

Returns encrypted string as base64 encoded string. The size of the encrypted string is 1.3 times larger than the size of plain text. Therefore you must have enough space to hold the encrypted string.

In case of a failure, the function throws an exception of type execution error, which results in the termination of the job. You can catch the exception by using try/catch handlers.

If the input string is empty, then the return value is empty.

If the input string is NULL, then the return value is NULL.

Where

`<Input_string>`	A varchar input string to be encrypted.
`<Passphrase>`	A varchar character string.

`<Salt>`	A varchar that must be exactly eight ASCII characters.
`<Key_length_in_bits>`	An int value of 128, 192, or 256.

Details

The function generates an AES key of specified key length using the specified passphrase, salt, and the key generation algorithm PKCS5_PBKDF2_SHA256. The function uses this key for encrypting the input string.

For security purposes, secure the passphrase and salts in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

Example

```
#read the passphrase from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
$G_salt = sql('PASSWORD_DATASTORE', 'select SALT from PASSWORD');
encrypt_aes_ext(SOURCE.SSN, $G_passphrase, $G_salt, 128);
```

Like other string functions, you can call the encrypt_aes_ext function from a custom function, from the column mapping of a Query transform, or from a script in the work flow.

6.3.5.27 exec

Use the exec function to send a command to the operating system.

Note

This function presents an elevated risk for command injection. Make sure you carefully check all parameters to avoid possible vulnerabilities.

If an injection could occur, a warning will be displayed the first time each such function is computed. If you prefer that that the job is terminated with an error when an injection could occur, add a `DSConfig flag ENABLE_SECURITY_ERROR = TRUE.`

Sends a command to the operating system on the SAP Cloud Integration for data services agent for execution. With this function, you can add a program to a SAP Cloud Integration for data services task or process.

Syntax

```
exec(<command_file, parameter_list, flag>)
```

Return value

Varchar(1020)

Returns up to 1020 characters that depend on the value of `<flag>` .

Where

`<command_file>`	A string that indicates the location and file name to execute. This string is relative to the Agent location. It can be an absolute or relative path. Ensure that the files and directories in the path are available from the Agent computer. The `<command_file>` can be a Windows batch file, a UNIX shell script, or a binary executable. To run other interpreted scripts, ensure that the `<command_file>` is the name of the command interpreter, such as 'perl', and the script is the first parameter in the `<parameter_list>`.
`<parameter_list>`	A string that lists the values to pass as arguments to the command file. Separate parameters with spaces. When passing no parameters to an executable, enter an empty string (' ').
`<flag>`	An integer that specifies what information appears in the return value string and how to respond when `<command_file>` cannot be executed or exits with a nonzero operating system return code.

Table 27: Exec function flags

Flag	If successful, returns:	On error:	Notes:
0	Standard output	Raises an exception: `System function failure.`	
1	NULL string	Raises an exception: `System function failure.`	Use this flag to track error states in either of the following cases: • The command never produces output • The calling job does not need output
2	Standard output	NULL string	Use this flag if you do not intend to track the status of the command other than the presence or absence of output.
3	NULL string	NULL string	
4	Standard output	Error message string	
5	NULL string	Error message string	
8	The concatenation of the return code and the combined stdout and stderr (standard error).	Returns the concatenation of the return code and the combined stdout and stderr (standard error).	

Flag	If successful, returns:	On error:	Notes:
256	NULL string	NULL string	Use this flag to run your program independently of SAP Cloud Integration for data services. Unlike flags 0-8, if you use flag 256, SAP Cloud Integration for data services does not wait until the command (executable program) completes before continuing with task or process processing. In this case, the command runs independently of SAP Cloud Integration for data services and stdout, stderr, and return code cannot be returned. Raises an exception (`System function failure`) if the program cannot be launched (e.g., program file not found).

Details

- Ensure that the program that this function executes does not wait for any user input (such as a prompt for a password). For flags 0-8, SAP Cloud Integration for data services waits for the program to complete. Therefore, if the program hangs for input, SAP Cloud Integration for data services also hangs. For flag 256, SAP Cloud Integration for data services continues if the program hangs for input.
- For flags 4 and 5, the return value format for an error message string is:

  ```
  'error-number: error-message-text'
  ```

  The first field is exactly 7 characters wide and the second character begins at index 10. If the program cannot be executed, the error number is 50307. If the program exits with a non zero return code, the error number is 50306. The text is from SAP `errormessage.txt`. For example:

  ```
  '  50306: Function <exec> failed to execute program 'foo.exe'. Program
  terminated with exit code 3.'
  ```

- For flag 8, the return value format is:

  ```
  'return-code: stdout-and-stderr'
  ```

The first field is exactly 7 characters wide and the second field begins at index 10. The program produces the return code. Zero indicates success. Consult your program documentation to determine the meaning of other codes.

Example

For example:

- `' 0: 8 file(s) copied.'`
- `' 1: The system cannot find the file specified.'`
- `' 1: a.tmp -> /usr/tmp/a.tmp cp: *.lcl: The system cannot find the file specified.'`
- `' -2: manmix(): fatal application error.'`

The 7-character format enables you to easily extract the first field, which is the return code from the executed command, as a string of digits. Data Services automatically converts the string of digits to an integer wherever necessary. The second field extracts as a regular string.

Example

For example:

- In a script:

  ```
  $foo = exec('foo.bat', ' ', 8);
  ```

  ```
  $foo_rc = substr($foo, 1, 7);
  ```

  ```
  $foo_txt = substr($foo, 10, 1020);
  ```

- In a data flow, map

  ```
  exec('foo.bat', '', 8)
  ```

  to an output column “foo” in a query. Then in a subsequent query, refer to the components of that column in a mapping or WHERE clause. For example:

  ```
  substr(query.foo, 1, 7);
  ```

  ```
  substr(query.foo, 10, 1020);
  ```

Related Information

Details about exec: Use of remote shells [page 278]

6.3.5.27.1 Details about exec: Use of remote shells

For the exec function, use a remote shell to run a command elsewhere on the network.

To use a remote shell, consider the following information:

- The `<command_file>` named in an exec call can be 'rsh' on either Windows or UNIX systems to call the remote shell facility. Use the 'rsh' as a means of running a command on a machine elsewhere on the network.

  > **Example**
  >
  > For example:
  >
  > - `exec('rsh', '<RemoteMachineName> <CommandToRunRemotely> <CmdArg1> <CmdArg2>', 0);`
  > - `exec('rsh', '<RemoteBox> -l<RemoteUser> <RemoteCommand> <CmdArg>', 3);`

  Call the remote shell facility sparingly, because the remote connection setup, remote authentication, and increased message traffic reduce performance.
- For `<flag>` values 4, 5, and 8, the return code that SAP Cloud Integration for data services receives is the rsh (or remsh) command. For example, 0 if it successfully gets a remote connection and authorization and nonzero if it does not get a remote connection and authorization. There is no relation between this return code and the return value of the remote command inherent in the remote shell mechanism on all the operating systems.
  To work around this behavior, wrap the remote command in a `.bat` file (Windows) or shell script (UNIX). Wrapping the remote command gets the command return code `%errorlevel%` if Windows or `$?` if UNIX, and prints it to stdout or stderr.

  > **Example**
  >
  > For example:
  >
  > - `exec('rsh', '<RemoteMachineName> <remcmdWrapper>.bat <CmdArg1> <CmdArg2>', 8);`
  > - `exec('rsh', '<RemoteBox> -l<RemoteUser> /usr/acta/<remcmdWrapper> <CmdArg>', 4);`
- The system administrator of the remote machine sets up access for the product user. The `.rhosts` and—or the `hosts.equiv` file has an entry allowing this access.
  - If the remote machine is Windows, ensure that the Remote Shell Service is running on it.
  - If the remote machine is UNIX, ensure that the Remote Shell daemon rshd is running on it.

  Consult your operating system documentation for more information.

> **Example**
>
> The following examples apply to Windows or UNIX. If you use the first two examples for UNIX, substitute 'sh', 'csh', 'ksh', 'bash' or 'tcsh' for 'cmd'. Also, the first two examples call 'cmd' rather than the program directly. Use 'cmd' or its equivalent if either:
>
> - The “command” is a built-in of the shell. For example, 'DIR' is not a program in Windows.
> - Piping, a single '|' in an argument, occurs

- In either Windows or UNIX, the pipe symbol sends the output of one command to another command. Only use a pipe inside quotes. In SAP Cloud Integration for data services, the double pipe symbol (||) concatenates strings. Only use a double pipe outside quotes.

Also, remember that the forward and backward slash symbols ('\' '/') are interchangeable in Windows. However use only the forward slash ('/') as a directory separator on UNIX.

```
exec('cmd', 'dir ' || $filename, 8);
exec('cmd', 'x:/bin/program1.exe | x:/bin/postprocess.bat', 4);
exec(SRC.PROGNAME, ARGTBL.ARGS || ' lastArg', 2);
exec('c:\Data Services\bin\clone_and_rename.bat', TBL.FNAME, 1);
exec('C:\Perl5\bin\perl.exe', 'C:\sandbox\stats.pl 20 50 3000', 0);
```

6.3.5.28 file_copy

Use the file_copy function to copy an existing file to a different location using the same file name or a different file name.

Note

This function presents an elevated risk for command injection. Make sure you carefully check all parameters to avoid possible vulnerabilities.

If an injection could occur, a warning will be displayed the first time each such function is computed. If you prefer that that the job is terminated with an error when an injection could occur, add a `DSConfig flag ENABLE_SECURITY_ERROR = TRUE.`

Syntax

```
file_copy(<source>,<target>,overwrite_if_exist)
```

Return Value

int

Returns 1 if the file is copied to the target location. Returns 0 if the file is not copied.

Where

`<source>`	The absolute path and name of the file to copy. Use a wildcard (*) in the file name to copy a group of files that match the wildcard criteria. Ensure that you have permission to access the source file location.

`<target>`	The absolute path for the location of the copied file. • To keep the same name as the source file, do not include a file name. • To rename the moved file, include a different file name. If you copy a group of files using a wildcard (*), enter the absolute path for the location of the copied files. Ensure that you have permission to access the target file and location.
`overwrite_if_exist`	Enter a 0 or 1. 0 = Do not overwrite any existing file. The software does not overwrite the file if it exists in the target location. **Note** In this case, the software return value is 0, and the software issues a warning that no files were copied to the target location. 1 = Overwrite any existing file. The software automatically overwrites the file if it exists in the target location. **Note** In this case, the software return value is 1, the software copies the source file to the target location, and it overwrites any existing file with the same name in the target location.

Details

The file_copy function overwrites any existing target file when you set the overwrite flag to 1. The source file still exists in the original location after `file_copy`.

Use `file_copy` on regular file types only. For example, you cannot use `file_copy` for directory file types or symbolic links.

Do not use the following characters in the source and target file name: `\ / : * ? " < > |` except when you use the asterisk (*) in a file name to indicate a wildcard.

Example

Function	Results
`file_copy('C:\temp\my_list.txt','D:\my_lists\list_a.txt',1)`	Copies a file and pastes it into a different location with a different file name. The function copies the `my_list.txt` file from the source location `C:\temp` and pastes it to the target location `D:\my_lists\` using a new name `list_a.txt`. The function automatically overwrites any existing file of the same name in the target location because the overwrite flag is set to 1.
`file_copy('C:\temp\my_*.txt','D:\my_lists',1)`	Copies a group of files from one location and pastes them into a different location. The function copies all files that match the wildcard file name `my_*.txt` from the source location `C:\temp` to the target location `D:\my_lists`. The function automatically overwrites any existing files of the same name in the target location because the overwrite flag is set to 1.

6.3.5.29 file_delete

Use the file_delete function to delete an existing file, or delete a group of files indicated by a wildcard (*).

Note

This function presents an elevated risk for command injection. Make sure you carefully check all parameters to avoid possible vulnerabilities.

If an injection could occur, a warning will be displayed the first time each such function is computed. If you prefer that that the job is terminated with an error when an injection could occur, add a `DSConfig flag ENABLE_SECURITY_ERROR = TRUE.`

Syntax

```
file_delete(<DelFileName>)
```

Return Value

int

Returns 1 if the stated file is deleted. Returns 0 if the stated file is not deleted.

Where

`<DelFileName>`	The absolute path and file name of an existing file to delete. Use a wildcard (*) in the file name to delete a group of files that match the wildcard criteria. Ensure that you have permission to the file and directory.

Details

Use `file_delete` on regular file types only. For example, you cannot use `file_delete` for directory file types or symbolic links.

You may not use the following characters in the deleted file name: `\ / : * ? " < > |` except when you use the asterisk (*) in a file name to indicate a wildcard.

Example

Function	Results
`file_delete('C:\users\myfile.txt')`	Deletes a file. The function deletes the file named `myfile.txt` from `C:\users`.
`file_delete('C:\users\my*.txt')`	Deletes a group of files matching a wildcard. The function deletes all files that match the wildcard file name `my*.txt` from the `C:\users` directory.

6.3.5.30 file_exists

Use the file_exists function to see if a file or directory exists.

Syntax

```
file_exists(<file_path>)
```

Return Value

int

Returns 1 if a file or directory is present on the disk, even if it is 0 bytes long. Returns a 0 if the file or directory is not present on the disk.

Where

`<file_path>`	The file name and path, relative to where the Agent is running. It can be an absolute or relative path.

Details

> **Example**
>
> Examples:
>
> Call sleep for 1 second when the file `temp.msg` exists in the directory called "c:".
>
> ```
> while (file_exists('c:/temp.msg') = 1)
> begin
>  sleep(1000);
> end
> ```
>
> Set a variable to a file name and use the function to check whether the file exists:
>
> ```
> $unix_file = '/tmp/t.cpp';
> if (file_exists($unix_file)) $type = 'unix';
> ```
>
> Set a variable based on the value of the function:
>
> ```
> $i = file_exists('c:/autoexec.bat')
> ```

6.3.5.31 file_move

Use the file_move function to move an existing file or group of files to a different location using the same file name or a different file name.

> **Note**
>
> This function presents an elevated risk for command injection. Make sure you carefully check all parameters to avoid possible vulnerabilities.
>
> If an injection could occur, a warning will be displayed the first time each such function is computed. If you prefer that that the job is terminated with an error when an injection could occur, add a `DSConfig flag ENABLE_SECURITY_ERROR = TRUE.`

> **Syntax**
>
> ```
> file_move(<source>,<target>, overwrite_if_exist)
> ```

Return Value

int

Returns 1 if the file is moved to the target location. Returns 0 if the file is not moved.

Where

`<source>`	The absolute path and name of the file to move. Use a wildcard (*) in the file name to move a group of files that match the wildcard criteria. Ensure that you have permission to access the source file and location.
`<target>`	The absolute path for the location of the moved file (or files). Ensure that you have permission to access the target file and location.
`overwrite_if_exist`	Enter a 0 or 1. 0 = Do not overwrite any existing file. The software does not overwrite the file if it exists in the target location. **i Note** In this case, the function return value is 0, and the software issues a warning that no files were moved to the target location. 1 = Overwrite any existing file. The software automatically overwrites the file if it exists in the target location. **i Note** In this case, the function return value is 1, the software moves the source file to the target location, and any existing file with the same name in the target location is overwritten.

Details

Overwrites any existing target file when the overwrite flag is set to 1. The source file does not exist in the original location after `file_move`.

Use `file_move` on regular file types only. For example, you cannot use `file_move` for directory file types or symbolic links.

- The source file no longer exists in the original location after `file_move`.
- You may not use the following characters in the source and target file name: \ / : * ? " < > |

  However, you may use the asterisk character (*) in a file name to indicate a wildcard.

- You can also use the `file_move` function to rename a file.

Example

Function	Results
`file_move('C:\temp\my_list.txt','C:\users',1)`	Moves a file from one folder to a different folder in the same directory. The software moves the file named `my_list.txt` from the `C:\temp` folder to the `C:\users` folder. After `file_move`, the source file `my_list.txt` no longer exists in the source folder. The software automatically overwrites any existing file named `my_list.txt` in the target location because the overwrite flag is set to 1.
`file_move('C:\temp\my_list.txt','C:\temp\my_list2.txt',0)`	Renames a file. The software renames the file `my_list.txt` to `my_list2.txt`. After the function successfully executes, the source file `my_list.txt` no longer exists. In this example, the `overwrite_if_exist` flag is set to 0, which means: • If the file doesn't already exist in the target location, the function return value is 1, and the file is moved to the target location. • If the file already exists in the target location, the function return value is 0, and the software issues a warning that the file was not moved to the target location because the file already exists.
`file_move('C:\temp\my*.txt', 'C:\users', 1)`	Move a group of files using a wildcard. The software moves all files that match the wildcard file name `my*.txt` from the source `C:\temp` to the target `C:\users`. After `file_move`, the files that match the wild card file name no longer exist in the source folder. The software automatically overwrites any files that already exist in the target folder because the overwrite flag is set to 1.

6.3.5.32 fiscal_day

Use the fiscal_day function to convert a date into an integer value that represents a day in a fiscal year.

Syntax

```
fiscal_day('<start_year_date>',<in_date>)
```

Return Value

int

Where

`<start_year_date>`	The first month and day of a fiscal year. Use the format: 'mm.dd'.
`<in_date>`	The date you want to convert. Use any valid datetime.

Details

Example

Function	Results
`fiscal_day('03.01', '1999.04.20')`	`51`

6.3.5.33 floor

Use the floor function to return the largest integer value equal to or less than a number.

Syntax

```
floor(<num>)
```

Return value

decimal, double, int, or real

The indicated integer, cast as the same type as the original number, `<num>` .

Where

`<num>`	The source number.

Details

Example

Function	Results
`floor(12.12345)`	`12.00000`
`floor(12)`	`12`
`floor(-12.223)`	`-13.000`

6.3.5.34 gen_row_num

Use the gen_row_num function to return an integer value beginning with 1, then incremented sequentially by 1 for each additional call.

Syntax

```
gen_row_num()
```

Return Value

int

Details

Each occurrence, or call, of the function in a data flow is a unique instance, resulting in a unique sequence. Two instances return values independent of each other. The first time the software calls an instance of this function, the function returns a value of 1. Subsequent calls of the same instance return the previous value incremented by 1, such as 2, 3, 4.

Each time the software calls the data flow, the software reinitializes all instances, and starts incrementing from 1.

Example

Function	Results
`gen_row_num(Col1)`	

Col1
0
0
0
0
0
0
0
0
0
0

Col1
1
2
3
4
5
6
7
8
9
10

6.3.5.35 gen_row_num_by_group

Use the gen_row_num_by_group function to generate a column of row identification numbers for each ID group in the specified column.

Syntax

```
gen_row_num_by_group(<expression_list>)
```

Return Value

Integer

Where

`<expression_list>`	A list of one or more comma-separated expressions.

Details

This function groups the rows in a table based on the values in the specified expression_list in the natural order. It returns a row ID beginning with 1, then increments it sequentially by 1 for each row in the group. When the group changes, the function restarts numbering at 1.

Example

For example, you have a table that lists record contracts by record number and contract ID. Values in Contract ID column are not unique.

Input

Record	Contract_ID	Revised_by	Revision_date
record 1	1	John	1/1/2005
record 2	1	Mary	1/15/2005
record 3	1	Tim	2/1/2005
record 4	2	Joe	2/24/05
record 5	2	Sue	2/30/05

When you apply `gen_row_num_by_group` function to the Contract_ID column, the software adds a new column to the output table that contains row numbers by group.

```
A version = gen_row_num_by_group (Contract_ID)
```

There are two groups in the Contract_ID column: Three rows for Contract_ID 1 and two rows for Contract_ID 2. The following table shows the output with the additional Version_Num column.

Output

Record	Contract_ID	Version_Num	Revised_by	Revision_date
record 1	1	1	John	1/1/2005
record 2	1	2	Mary	1/15/2005

Output

Record	Contract_ID	Version_Num	Revised_by	Revision_date
record 3	1	3	Tim	2/1/2005
record 4	2	1	Joe	2/24/05
record 5	2	2	Sue	2/30/05

If the <expression_list> value corresponds to a column in a table, like in the preceding example, the column must not be a nested relational data model (NRDM) or have the data type long. Also, do not use this function with any group by clauses or aggregate functions.

* Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

6.3.5.36 gen_uuid

Use the gen_uuid function to generate a unique identifier.

Syntax

```
gen_uuid()
```

Return value

Varchar

Returns a unique 32-character varchar string. For example, 550e8400e29b41d4a716446655440000.

6.3.5.37 get_data

Retrieves stored data that contains the task name and the most current load date.

Syntax

```
get_data ('<task_name>')
```

Where

`<task_name>`	The name created for the task.

Details

The `<task_name>` must be varchar. The maximum data size is 255 characters.

Example

Functions	Results
`get_data('hello_world')`	SAP Cloud Integration for data services retrieves the most current load date of hello_world.

Related Information

Change Data Capture (Delta Loads) [page 158]

6.3.5.38 greatest

Use the greatest function to return the greatest of the list of one or more expressions.

Syntax

```
greatest(<expression_list>)
```

Return Value

SAP Cloud Integration for data services uses the first expression to determine the return type. After comparison, the result is converted into the return data type.

Where

`<expression_list>`	A list of one or more comma-separated expressions.

Details

GREATEST returns the greatest of the list of one or more expressions. After comparison, the result is converted into a return data type. SAP Cloud Integration for data services implicitly converts expression in the list to a normalized data type before comparison.

The software uses the following rules to determine the normalized data type.

- If the return data type is varchar, then the software implicitly normalizes all expressions to varchar before comparison.
- If the return data type is one of the date data types, then the software implicitly normalizes all the expressions in the list to that data type before comparison.

  > **Example**
  >
  > For example, if the return data type is date, and another data type is 'datetime', then the software normalizes the 'datetime' data type to 'date' before comparison.

- If the return data type is numeric, then the software implicitly normalizes all the expressions to the highest precedence numeric expression in the list.

  > **Example**
  >
  > For example, `greatest (expr1,expr2,expr3,expr4)` where expr1 is as integer, expr2 is a decimal (4,2), expr3 is a float, expr4 is a decimal (38,7). The normalized data type is decimal.

  The software converts all the expressions in the list to decimal data type before comparison. If the normalized data type is decimal, then the precision is the highest precision among all decimal data type expressions. The software preserves the scale for decimal data type expressions during implicit conversion. When the software converts an integer data type expression to a decimal data type, its scale is 0. When float, double and varchar data types are converted into decimal data types, their scale is 6.

> **Note**
>
> `greatest()` returns NULL when at least one argument is NULL.

Example

Input

ID	GRADE_Q1	GRADE_Q2	GRADE_Q3	GRADE_Q4
1	'A'	'B'	'B'	'C'
2	'F'	'F'	'E'	'C'
3	'B'	'B'	NULL	NULL

Output

MAX_GRADE=greatest(GRADE_Q1,GRADE_Q2,GRADE_Q3,GRADE_Q4)

ID	MAX_GRADE
1	'C'
2	'F'
3	NULL

6.3.5.39 ifthenelse

Use the ifthenelse function to enable conditional logic in expressions.

Syntax

```
ifthenelse(<condition>, <true_branch>, <false_branch>)
```

Return value

`<true_branch>` or `<false_branch>`

Returns one of the values provided, based on the result of `<condition>`. The data type of the return value is the data type of the expression in `<true_branch>`. If the data type of `<false_branch>` is not convertible to the data type of `<true_branch>`, SAP Cloud Integration for data services produces an error at validation. If the data types are convertible but don't match, a warning appears at validation.

Where

`<condition>`	An expression that evaluates to TRUE or FALSE.

`<true_branch>`	An expression that the function returns if `<condition>` evaluates to TRUE.
`<false_branch>`	An expression that the function returns if `<condition>` evaluates to FALSE.

Details

If `<condition>` compares a varchar value with trailing blanks, the ifthenelse function ignores the trailing blanks.

To compare a NULL value (NULL constant or variable that contains a NULL constant), use the IS NULL or IS NOT NULL operator. If you use the Equal (=) or Not equal to (<>) operator to compare against a NULL value, `<condition>` always evaluates to FALSE.

To improve performance, SAP Cloud Integration for data services pushes this function to the database. Thus, the database evaluates the IFTHENELSE logic rather than the engine.

Use this function to apply conditional logic when mapping columns or selecting columns in a query. For more flexible control over conditions in a script, use the IF keyword in the scripting language.

Example

Function	Results
`ifthenelse (ZIP < 94000, 'SOUTH', 'NORTH')`	If the value in the column `ZIP` is less than 94000, the value returned is `SOUTH`. If `ZIP` is greater than 94000, then the value returned is `NORTH`.

6.3.5.40 index

Use the index function to return the index of a give character sequence in a string.

Syntax

```
index(<input_string>, <index_string>, <start>)
```

Return value

int

Specifies the first location of the indicated character sequence.

Where

`<input_string>`	The source string.
`<index_string>`	The character sequence sought in `<input_string>`.
`<start>`	The position where the function starts searching in `<input_string>` for the character sequence contained in `<index_string>`. `<Start>` should be a positive number between 1 and the length of `<input_string>`.

Details

The function searches for the `<index_string>` beginning at the `<start>` position in the input_string.

- If `<start>` is 0, the function resets it to 1.
- If `<start>` is greater than the number of characters in `<input_string>`, the function returns NULL.
- If `<index_string>` is not found in `<input_string>`, the function returns NULL.

Ensure that the characters in `<index_string>` exactly match the sequence of characters in `<input_string>`.

Example

The search is case-sensitive.

Function	Results
`index('Accounting Department', 'DEPARTMENT', 1)`	`NULL` The index_string does not match the case of the input_string so the function returns NULL.
`index('Accounting Department', 'Department', 1)`	`12` The string "Department" starts at position 12 when you begin counting at position 1, which is the "A" of Accounting.

6.3.5.41 init_cap

Use the init_cap function to convert the first letter of each word in a string to upper case and the rest of the value to lowercase. The function ignores all characters that are not alphabetic.

Syntax

```
init_cap(<value>,'<locale>')
```

Return value

varchar

The title case string. Words are delimited by white space or characters that are not alphanumeric.

Where

`<value>`	The string to be modified.
`<locale>`	Optional parameter that converts the string to the specified locale. **Note** The function supports ISO 639 language code and ISO 3166 country code formats.

Details

Example

Function	Results
`init_cap('Data Services')`	'Data Services'
`init_cap(StreetAddress)`	Writes the value, for example '1234 west washington school road', in column StreetAddress as '1234 West Washington School Road'.
`Print(Init_cap('have a nice day –hypen +plus _underscore \slash $dollar *star @at tab mIXedWORd UPPER lower !punctations 1234digits'));`	Have A Nice Day -Hypen +Plus _Underscore \Slash $Dollar *Star @At Tab Mixedword Upper Lower !Punctuations 1234digits

Function	Results
`init_cap(LastName,'tr')`	Converts the value in column`LastName` to have the first letter capitalized. If there are more than one last name in this column, the first letter of each word is capitalized. Converts the values in LastName to the Turkish locale, using the ISO 639 language code.

Limitations

- The function can be pushed down to Oracle databases only.
- You cannot use this function in an ABAP data flow.

6.3.5.42 is_group_changed

Use the is_group_changed function to return an integer, which indicates if the current occurrence of a group of values has changed from the previous occurrence.

Syntax

```
is_group_changed(<expression>)
```

Return Value

Integer

1 = Group has changed

0 = Group has not changed

Where

`<expression>`	One or more valid input expressions separated by commas.

Details

This function groups records based on the equal value of the input expressions in parameter1 in the natural order of the input record stream. It returns 1 when the group is changed, 0 otherwise.

Example

In the following example, the results show that four of the input groups have changed.

Function	Results
`is_group_changed(state,city)`	`1,0,1,0,0,1,1`

Group ID	State	City	Group change
1	California	Los Angeles	1
2	California	Los Angeles	0
3	California	San Francisco	1
4	California	San Francisco	0
5	California	San Francisco	0
6	Nevada	Reno	1
7	Colorado	Reno	1

6.3.5.43 is_valid_date

Use the is_valid_date function to indicate whether an expression can be converted into a valid calendar date value.

Syntax

```
is_valid_date(<input_expression>,'<date_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated. If the expression does not resolve to a value of data type `varchar`, the software issues a warning that the value has been converted to a `varchar`.	
`<date_format>`	The string identifying the date format of the input string. Construct the date format using the following codes and other literal strings or punctuation:	
	DD	2-digit day of the month
	MM	2-digit month
	MONTH	Full name of month
	MON	3-character name of month
	YY	2-digit year
	YYYY	4-digit year

Details

Example

For example the following expression returns 0 because there is no such date as January 34th.

```
is_valid_date ('01/34/2002', 'mm/dd/yyyy')
```

Example

Function	Results
`is_valid_date (Orders.SubmitDate,'mm/dd/yyyy')`	Tests whether the string `Orders.SubmitDate` can be converted to a calendar date with the `mm/dd/yyyy` date format.

6.3.5.44 is_valid_datetime

Use the is_valid_datetime to indicate whether an expression can be converted into valid calendar date and time values.

Syntax

```
is_valid_datetime(<input_expression>,'<datetime_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated.	
`<datetime_format>`	The string identifying the datetime format of the input expression. Construct the datetime format using the following codes and other literal strings or punctuation:	
	DD	2-digit day of the month
	MM	2-digit month
	MONTH	Full name of month
	MON	3-character name of month
	YY	2-digit year
	YYYY	4-digit year
	HH24	2-digit hour of the day (00-23)
	MI	2-digit minute (00-59)
	SS	2-digit second (00-59)

Details

Example

For example the following expression returns 0 because there is no such hour as 26:

```
is_valid_datetime ('01/14/2002 26:56:09', 'mm/dd/yyyy hh24:mi:ss')
```

Example

Function	Results
`is_valid_datetime (Orders.Received,'mm/dd/yyyy hh24:mi:ss')`	Tests whether the string `Orders.Received` can be converted to the `mm/dd/yyyy hh24:mi:ss` datetime format.

6.3.5.45 is_valid_decimal

Use the is_valid_decimal to indicate whether an expression can be converted into a valid decimal value.

Syntax

```
is_valid_decimal(<input_expression>,'<decimal_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated.
`<decimal_format>`	A string indicating the decimal format of the input expression. Use pound characters (#) to indicate digits and a decimal indicator. If necessary, include commas as thousands indicators. For example, to specify a decimal format for numbers smaller than 1 million with 2 decimal digits, use the following string: `'#,###,###.##'`. To indicate a negative decimal number, add a minus "-" sign at the beginning or end of this value. For example, to test if the stock price difference can be converted to decimal format, use the following function: `is_valid_decimal (Stocks.Price_difference, '-###.##')`

Details

Example

Function	Results
`is_valid_decimal (Orders.Price,'###,###.##')`	Tests whether the string `Orders.Price` can be converted to decimal format.

6.3.5.46 is_valid_double

Use the is_valid_double function to indicate whether an expression can be converted into a valid double value.

Syntax

```
is_valid_double(<input_expression>,'<double_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated.
`<double_format>`	A string indicating the double format of the input expression. Use pound characters (#) to indicate digits and a decimal indicator. If necessary, include commas as thousands indicators. For example, to specify a double format for numbers smaller than 1 million with 2 decimal digits, use the following string: `#,###,###.##'`

Details

Example

Function	Results
`is_valid_double (Product.Weight,'###.###')`	Tests whether the string `Product.Weight` can be converted to double format.

6.3.5.47 is_valid_int

Use the is_valid_int function to indicate whether an expression can be converted into a valid integer value.

Syntax

```
is_valid_int(<input_expression>,'<int_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated.
`<int_format>`	The format specifying the thousands separator of the input expression. For example, to specify an integer format, use the following string: #.###.###'. Valid separators include the period (.) and the comma (,). However, you can use only one valid separator type in a format. Separator defaults to the comma (,) when none is specified.

Details

Example

Function	Results
`is_valid_int (QuarterResults.Volume,'###.###')`	Tests whether the string `QuarterResults.Volume` can be converted to the `###.###` integer format.

6.3.5.48 is_valid_real

Use the is_valid_real function to indicate whether an expression can be converted into a valid real value.

Syntax

```
is_valid_real(<input_expression>,'<real_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated.
`<real_format>`	A string indicating the real format of the input expression. Use pound characters (#) to indicate digits and a decimal indicator. For example, to specify a real format for numbers smaller than 1 million with 2 decimal digits, use the following string: `'#,###,###.##'`.

Details

Example

Function	Results
`is_valid_real (QuarterResults.Mean,'#,###.#####')`	Tests whether the string `QuarterResults.Mean` can be converted to real format.

6.3.5.49 is_valid_time

Use the is_valid_time function to indicate whether an expression can be converted into a valid time value.

Syntax

```
is_valid_time(<input_expression>,'<time_format>')
```

Return value

int

- 1 = The expression is not NULL and is valid.
- 0 = The expression is not NULL and is invalid.
- NULL = The expression is NULL.

Where

`<input_expression>`	The expression to be validated.	
`<time_format>`	The string identifying the time format of the input expression. Construct the time format using the following codes and other literal strings or punctuation:	
	HH24	2-digit hour of the day (00-23)
	MI	2-digit minute (00-59)
	SS	2-digit second (00-59)

Details

Example

Function	Results
`is_valid_time (Orders.ReceivedTime,'hh24:mi:ss')`	Tests whether the string `Orders.ReceivedTime` can be converted to the `hh24:mi:ss` datetime format.

6.3.5.50 isweekend

Use the isweekend function to indicate whether a date corresponds to Saturday or Sunday.

Syntax

```
isweekend(<date1>)
```

Return value

int

The result of the date test:

- Returns 1 when the date is a Saturday or Sunday.
- Returns 0 when the date is not a Saturday or Sunday.

Where

`<date1>`	The value of type `date` or `datetime` to test.

Details

Example

Function	Results
`isweekend(hire_date)`	Tests whether the date in `hire_date` is a Saturday or Sunday.
`isweekend(SYSDATE)`	Tests whether the current date is a Saturday or Sunday.

6.3.5.51 job_name

Use the job_name function to return the name of the object, such as a job, in which the call to this function exists.

Returns the name of the task in which the call to this function exists.

Syntax

```
job_name()
```

Return Value

varchar

Details

Example

```
print('Starting execution of Job: [job_name()] as user: [system_user_name()]');
```

6.3.5.52 julian

Use the julian function to convert a date to the integer julian value. The Julian value is the number of days between the start of the Julian calendar and the given date.

Syntax

```
julian(<date1>)
```

Return value

int

The Julian representation of the date.

Where

`<date1>`	The source value of type `date` or `datetime`.

Details

Example

The following example uses the to_date function to convert the string to a date using the stated format. Then, the julian function converts the date to the Julian representation of the date.

Function	Results
`julian(to_date('Apr 19, 1997', 'mon dd, yyyy'))`	729436

6.3.5.53 julian_to_date

Use the julian_to_date function to convert a given Julian value to a date.

Syntax

```
julian_to_date(<input_julian>)
```

Return value

date

The date that corresponds to the input Julian value.

Where

`<input_julian>`	An integer representing the Julian value to be converted.

Details

Example

Function	Results
`julian_to_date(Julian_Date)`	Converts the number indicated by `Julian_Date` to its date value.

6.3.5.54 last_date

Use the last_date function to return the last date of the month for a given date.

Syntax

```
last_date(<in_date>)
```

Return Value

date

Where

`<in_date>`	The date for which the last date of the month is to be calculated.

Details

Example

Function	Returns
`last_date('1990.10.01')`	'1990.10.31'

6.3.5.55 least

Use the least function to return the least of the list of one or more expressions.

Syntax

```
least(<expression_list>)
```

Return Value

SAP Cloud Integration for data services uses the first expression to determine the return type. After comparison, the result is converted into the return data type.

Where

`<expression_list>`	A list of one or more comma-separated expressions.

Details

SAP Cloud Integration for data services implicitly converts expressions in the list to a normalized data type before comparison.

The software uses the following rules to determine the normalized data type:

1. If the return data type is varchar, then implicitly normalizes all expressions to varchar before comparison.
2. If the return data type is one of the date data types, then implicitly normalizes all expressions in the list to that data type before comparison.

   > **Example**
   >
   > For example, if the return data type is date, and another data type is 'datetime', then the 'datetime' data type is normalized to 'date' before comparison.

3. If the return data type is numeric, then implicitly normalizes all the expressions to the highest precedence numeric expression in the list.

   > **Example**
   >
   > For example, `least(expr1,expr2,expr3,expr4)` where expr1 is an integer, expr2 is a decimal (4,2), expr3 is a float, expr4 is a decimal (38,7). The normalized data type is decimal.

   The software converts all the expressions in the list to decimal data types before comparison. If the normalized data type is decimal, then the precision is the lowest precision among all decimal data type expressions. The decimal data type expressions preserve their scale during implicit conversion. When the software converts an integer data type expression to a decimal data type, the scale is 0. When the software converts float, double, and varchar data types into decimal data types, the scale is 6.

> **Note**
>
> `least()` returns NULL when at least one argument is NULL.

> **Example**

Input				
ID	**GRADE_Q1**	**GRADE_Q1**	**GRADE_Q3**	**GRADE_Q4**
1	'A'	'B'	'B'	'C'
2	'F'	'F'	'E'	'C'
3	'B'	'B'	NULL	NULL

Output		
MIN_GRADE=least (GRADE_Q1, GRADE_Q2, GRADE_Q3, GRADE_Q4)		
ID	**MAX_GRADE**	**MIN_GRADE**
1	'C'	'A'
2	'F'	'C'
3	NULL	NULL

6.3.5.56 length

Use the length function to return the number of characters in a given string.

Syntax

```
length(<value>)
```

Return value

integer

The number of characters in `<value>`.

Where

`<value>`	A string indicating the column name, variable, or other element whose length is calculated.

Details

Example

In the *Mapping* box of a query, use the length function to return the number of characters in each row of a column. With the OUTPUT field selected in the target schema of a query, enter the following statement in the *Mapping* box:

```
length(dal_emp.ename)
```

The software produces the following results:

Source column (dal_emp.ename)	Target column (output)
jones	5
nguyen	6
tanaka	6

6.3.5.57 literal

Use the literal function to return an input constant expression without interpolation.

Syntax

```
literal(<input>)
```

Return value

Same value as the value given for the input parameter but without interpolation.

Where

<input>	A constant expression of any data type.

Details

SAP Cloud Integration for data services does not use variable interpolation on constants. However, if you pass in a variable as a constant expression, SAP Cloud Integration for data services automatically uses variable interpolation, replacing special characters.

Replacing special characters is an issue with the match_pattern and match_regex functions because they require these special characters. If your `pattern_string` or `regular_expresion_pattern` parameter in these functions is a constant, you may want to disable interpolation. If so, use the literal function.

Example

For example, you want to match `$my_pattern` with the pattern `'PART[123]'`.

If you code it as `$my_pattern = 'PART[123]'; match_pattern(product,$my_pattern);`, the interpolation changes the pattern being matched to 'PART123'. However, if you code it as `$my_pattern = literal ('PART[123]'); match_pattern(product, $my_pattern);`, the return value is 1 because the pattern remains 'PART[123]'.

Alternatively, if you do not want to use a variable, you can code it as `match_pattern (product,'PART[123]');`. Then the software does not interpolate on the constant 'PART[123]'.

There is no runtime cost for the literal function. SAP Cloud Integration for data services substitutes the constant expression at compile time.

Example

To match only PART1 or PART2 or PART3 using the match_pattern function, assign a pattern to a variable without interpolation. Use the literal function in the following type of expression:

```
$pattern = literal('PART[123]');
```

If you do not use the literal function, the value assigned to `$my_pattern` in the following sample is 'PART123'. That is because Data Services automatically removes square brackets during interpolation.

```
$my_pattern='PART[123]';
print($my_pattern);
if (match_pattern('PART1',$my_pattern) <> 0)
 print('Matched');
else
print('Not Matched');
```

To disable interpolation, use the literal function. The following example returns the expected result.

```
$my_pattern=LITERAL('PART[123]');
print($my_pattern);
if (match_pattern('PART1',$my_pattern) <> 0)
print('Matched');
else
print('Not Matched');
```

6.3.5.58 ln

Use the ln function to return the natural logarithm of the given numeric expression.

Syntax

```
ln(<numeric_expression>)
```

Return Value

Float

If input is negative, return value is NULL.

Where

`<numeric_expression>`	Any numeric expression.

Details

Returns the natural logarithm of the given numeric expression.

Example

Function	Results
`ln(5.436563656918)`	1.693147

6.3.5.59 local_to_utc

Use the local_to_utc function to convert the input datetime of any time zone to Coordinated Universal Time (UTC).

Syntax

```
local_to_utc(<input datetime>, <timezone of the input with UTC offset>)
```

Return Value

datetime

Details

Converts the input datetime of any time zone to Coordinated Universal Time (UTC). The second parameter UTC offset is a constant value. If the UTC offset is not provided, then it is taken as the time zone of the agent host to calculate the UTC offset.

Details

Example

Function	Results
`local_to_utc('2014.02.01 00:00:00', 'UTC+08:30')`	'2014.01.31 15:30:00'

6.3.5.60 log

Use the log function to return the base-10 logarithm of the given numeric expression.

Syntax

```
log(<num>)
```

Return Value

Float

If the input is negative, the return value is NULL.

Where

`<num>`	The number for which you want a base- 10 logarithm returned.

Details

Example

Function	Results
`log(100.000)`	2.000000

6.3.5.61 lookup

Use the lookup function to retrieve a value in a table or file based on the values in a different source table or file.

Syntax

```
lookup (<lookup_table>, <result_column>, <default_value>, <cache_spec>,
<compare_column>, <expression>)
```

Return value

Any type

The value in the `<lookup_table>` that meets the lookup requirements. The return type is the same as `<result_column>`.

Where

`<lookup_table>`	The table or file that contains the result or value you are looking up (`<result_column>`). The Use a fully qualified table name that includes the datastore, owner, and table name. For example: oracle_ds.TIGER.sales. `<compare_column>` is also located in this table. You might need to put the owner in quotes, particularly if you use lower case letters.
`<result_column>`	The column containing the values you want to retrieve. This column is in the table or file listed for `<lookup_table>`.
`<default_value>`	The value returned when there is no matching row in the table or file listed for `<lookup_table>`Use a fully qualified table name that includes the datastore, owner, and table name. For example:.

<cache_spec>	The caching method that the lookup operation uses. Enclose with single quotes. There are three possible settings: • NO_CACHE: Reads values from the <lookup_table> for every row without caching values. • PRE_LOAD_CACHE: Loads the <result_column> and <compare_column> into memory after applying filters but before executing the function. Select this option if the number of rows in the table is small or you expect to access a high percentage of the table values. • DEMAND_LOAD_CACHE: Loads <result_column> and <compare_column> values into memory as the function identifies them. Select this option if the number of rows in the table is large and you expect to access a low percentage of the table values frequently. Select this option when you use the table in multiple lookups and the compare conditions are highly selective, resulting in a small subset of data.
<compare_column>	The column in the <lookup_table> that the function uses to find a matching row. When the function reads a varchar column in the <lookup_table>, it does not trim trailing blanks.
<expression>	The value that the function searches for in the <compare_column>. The value can be a simple column reference, such as a column found in both a source and the <lookup_table>. The value can also be a complex expression given in terms of constants and input column references. When <expression> refers to a unique source column, you do not need to include a table name qualifier. If <expression> is from another table or is not unique among the source columns, you need a table name qualifier. If <expression> is an empty string, the function searches for a zero-length varchar value in the <compare_column>. The function ignores trailing blanks in comparisons of <expression > and values in <compare_column>.

> **i** Note
>
> You can specify more than one <compare_column> and <expression> pair. To specify more than one pair, add additional pairs at the end of the function statement. Ensure that the values match for all specified pairs in order for the lookup function to find a matching row.

Details

The lookup function uses a value that you provide in <expression> to find a corresponding value in a file or different table. Specifically, the function searches for the row in the <lookup_table> where the value in the <compare_column> matches the value in <expression>. The function returns the <result_column> value from this matching row.

For example, if your source schema uses a customer ID to identify each row, but you want the customer name in your target schema, you can use the lookup function to return the customer name given the customer ID.

In SQL terms, the lookup function evaluates `<expression>` for each row, then executes the following command:

```
SELECT <result_column>
    FROM <lookup_table>
    WHERE <compare_column> = <expression>
```

The value returned by this SELECT statement is the result of the lookup function for the row.

You can specify multiple `<compare_column>` and `<expression>` pairs to uniquely identify the `<result_column>` value. However, the software provides only fields for one pair; add extra `<compare_column>` and `<expression>` pairs to the output.

When there are no matching rows in the `<lookup_table>`, the lookup function returns the `<default_value>`. When multiple matching rows exist in the `<lookup_table>`, the row that the lookup function returns is based on whether the lookup table is a standard RDBMS table, an SAP application table, or a flat file:

- For standard RDBMS tables, the lookup function finds the matching row with the maximum value in the `<result_column>` and returns that value.
- For SAP application tables or flat files, the lookup function randomly selects a matching row and returns the value in the `<result_column>` for that row.

To enhance performance, configure the lookup function to hold the values from the `<lookup_table>` in memory. To do so, use the `<cache_spec>` setting. The optimal setting depends on the number of rows the function must read, the number of rows in the table, and the available memory.

6.3.5.62 lower

Use the lower function to change the characters in a string to lower case.

Syntax

```
lower(<value>,'<locale>')
```

Return value

varchar

The lowercase string. The return type is the same as `<value>`. The function leaves any characters that are not letters unchanged.

Where

`<value>`	The string to be modified.
`<locale>`	Optional. Specify a locale to convert the string to the locale. **i Note** The function supports the ISO 639 language code and the ISO 3166 country code formats.

Details

Example

Function	Results
`lower('Accounting101')`	'accounting101'
`upper((LastName,1,1))| lower(substr(LastName,2,LENGTH(LastName )))`	The value in column `LastName` with the first letter uppercase and the rest of the value lowercase. Note that this example does not account for two-word last names.
`lower(LastName,'tr')`	The value in column `LastName` is converted to all lowercase. It is also converted to the Turkish locale, using the ISO 639 language code.

6.3.5.63 lpad

Use the lpad function to pad the left side of a string with specific characters.

Syntax

```
lpad(<input_string>,<size>,'<pad_string>')
```

Return value

varchar

The modified string. The return type is the same as `<value>`. The function leaves any characters that are not letters unchanged.

Where

`<input_string>`	The string source, such as a column name.
`<size>`	An integer value indicating the number of characters in the return string.
`<pad_string>`	A character or set of characters that this function concatenates to `<input_string>`.

Details

This function repeats the pattern at the beginning of the input string until the final string is the appropriate length. If the input_string is already longer than the expected length, then this function returns a truncated string without adding special characters.

Example

Function	Results
`lpad('Tanaka', 15, ' ')`	`'  Tanaka'` Note The character in `<pad_string>` is a space.
`lpad(last_name, 25, ' ')`	The value in the column last_name, padded with spaces from the left to 25 characters. If the value in last_name exceeds 25 characters, truncates from the right.

6.3.5.64 lpad_ext

Use the lpad_ext function to pad the left side of a string with logical characters from a given pattern.

Syntax

```
lpad_ext(<input_string>,<size>,'<pad_string>')
```

Return value

varchar

The modified string. The return type is the same as `<value>`. The function leaves any characters that are not letters unchanged.

Where

<input_string>	The string source.
<size>	An integer value indicating the number of characters in the return string.
<pad_string>	A logical character or set of logical characters that this function concatenates to the <input_string>.

Details

The logical characters prohibit this function from getting pushed down to the database.

The function repeats the value in <pad_string> from the beginning of the input string until the final string is the length set in <size>. If the value in <input_string> is already longer than the expected length, then this function truncates the string from the right.

Example

Function	Results
lpad_ext('Tanaka', 15, ' ')	' Tanaka'
lpad_ext(last_name, 25, ' ')	The value in the column last_name, padded with spaces to 25 characters on the left. If the string alone exceeds 25 characters, truncates the string to 25 characters from the right.

Example

The lpad_ext and lpad functions exhibit the same behavior when the software evaluates the functions. However, the database behavior is different when the software pushes the function down to the database and the value in <input_string> and—or <pad_string> contain multibyte characters.

Function	Input	Output
lpad	lpad("abc¶¶", 10,' ')	' abc¶¶'
lpad_ext	lpad_ext("abc¶¶", 10,' ')	' abc¶¶'
lpad	lpad("abc¶¶", 10,'')	'abc¶¶'
lpad_ext	lpad_ext("abc¶¶", 10,'')	'abc¶¶'
lpad	lpad("abcd", 10, '¶')	'¶¶¶abcd'
lpad_ext	lpad_ext("abcd", 10, '¶')	'¶¶¶¶¶¶abcd'

6.3.5.65 ltrim

Use the ltrim function to remove specified characters from the start of the string.

Syntax

```
ltrim(<input_string>, <trim_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`	The string to be modified.
`<trim_string>`	The characters to remove from `<input_string>`.

Details

The ltrim function is case-sensitive.

The function scans `<input_string>` left-to-right removing all characters that appear in `<trim_string>` until it reaches a character not in `<trim_string>`.

Example

Function	Results
`ltrim('Marilyn', ' ')`	'Marilyn'
`ltrim('ABCABCD', 'ABC')`	'D'
`ltrim('ABCABCD', 'EFG')`	'ABCABCD'
`ltrim('ABCDEABCDE', 'ABC'`	'DEABCDE'

Example

To remove all leading blanks in a string, use ltrim as follows:

```
ltrim(EMPLOYEE.NAME, ' ')
```

where `EMPLOYEE.NAME` specifies the `NAME` column in the `EMPLOYEE` table. You may also use the ltrim_blanks or ltrim_blanks_ext functions for this.

6.3.5.66 ltrim_blanks

Use the ltrim_blanks function to remove blank characters from the start of a string.

Syntax

```
ltrim_blanks(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`	The string to be modified.

Details

Example

Function	Results
`ltrim_blanks(' Marilyn')`	'Marilyn'
`ltrim_blanks(last_name)`	The value contained in the column last_name, with all leading blanks removed.

6.3.5.67 ltrim_blanks_ext

Use the ltrim_blanks_ext function to remove blank and control characters from the start of a string.

Syntax

```
ltrim_blanks_ext(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`	The string to be modified.

Details

Example

Function	Results
`ltrim_blanks_ext(' Marilyn')`	'Marilyn'
`ltrim_blanks_ext(last_name)`	The value contained in the column last_name, with all leading blanks and control characters removed.

6.3.5.68 match_pattern

Use the match_pattern function to match a whole input string to simple patterns supported by the software.

Syntax

```
match_pattern(<input_string>,<pattern_string>)
```

Return Value

int

Returns:

- 1: Pattern matched
- 0: Pattern did not match

Where

`input_string`	String to be matched. Supports UNICODE characters.
`pattern_string`	Pattern to find in the whole input string. Create `<pattern_string>` using characters listed in the following table. The function does not support substring matches.

Details

Table 28: Characters for pattern_string

X	Represents uppercase characters. Unicode 4.2 General Category Values specification. Key = Lu, uppercase letter (For example, Latin, Greek, Cyrillic, Armenian, Deseret, and archaic Georgian.)
x	Represents non uppercase characters. Unicode 4.2 General Category Values specifications keys: • Ll = Lowercase letter (For example, Latin, Greek, Cyrillic, Armenian, Deseret, and archaic Georgian.) • Lt = Titlecase letters (For example, Latin capital letter D with small letter Z.) • Lm = Modifier letter (For example acute accent, grave accent.) • Lo = Other letter (Includes Chinese, Japanese, and so on.)
9	Represents numbers.
\	Escape character.
*	Any characters occurring zero or more times.
?	Any single character occurring once and only once.
[ ]	Any one character inside the braces occurring once.
[!]	Any character except the characters after the exclamation point. For example, [!12] can allow any number that does not start with a 1 or 2.

All other characters represent themselves. To specify a special character as itself, use an escape character. For example, [!9] means any character except a digit. To specify any digit except 9, use [!\9].

The following table displays pattern strings that represent example values:

Table 29: Example pattern strings

Example value	Pattern string
Henrick	Xxxxxxx
DAVID	XXXXX
Tom Le	Xxx Xx
Real-time	Xxxx-xxxx
JJD)$@&*hhN8922hJ7#	XXX)$@&*xxX9999xX9#
1,553	9,999
0.32	9.99
-43.88	-99.99
Returns names with last name Jones	*Jones
Returns Henrick1 or HenrickZ	Henrick?
Returns David1 or David2 or David3	David[123]

Example

Use the `match_pattern` function in the Validation transform or in a WHERE clause of a Query transform. The input string can be from sources such as columns, variables, or constant strings.

Use case	Pattern	Function call in a script	Results
To match a zip code except one that begins with 1 or 2.	'[!12]9999'	`if (match_pattern('15014', '[!12]9999') <> 0) print('matched'); else print('not matched');`	Function prints “not matched”.
To match a zip code except one that begins with 1 or 2.	'[!12]9999'	`if (match_pattern('55014', '[!12]9999') <> 0) print('matched'); else print('not matched');`	Function prints “matched”.

Use case	Pattern	Function call in a script	Results
To process only customer phone numbers that fit the same pattern.	'999-999-9999'	`WHERE MATCH_PATTERN(CUSTOMER.PHONE_NUM,'999-999-9999') <> 0`	Phone numbers that do not match the pattern throw error 0.
To check a string against a complex pattern and print result to trace log.	'XXX)$@&*xxX9999xX9#'	`if(match_pattern('JJD) $@&*hhN8922hJ7#', 'XXX) $@&*xxX9999xX9#') <> 0) print ('matched'); else print('not matched');`	The result for this call is "matched".

6.3.5.69 match_regex

Use the match_regex function to match whole input strings to the pattern that you specify with regular expressions and flags.

Syntax

```
match_regex (<input_string>, <regular_expression_pattern>, <flags>)
```

Return Value

int

Returns:

- 1 = Pattern matched
- 0 = Pattern does not match

Where

`<input_string>`	String to be matched. The function supports UNICODE characters.

`<regular_expression_pattern>`	Pattern you want to find in a whole input string. The function does not match substrings. Provide the pattern in regular expression format with a varchar data type.
`<flags>`	Allows you to specify additional behavior that you want to occur while SAP Cloud Integration for data services searches the input_string for pattern matches. Specify a flag, or specify more than one flag separated by commas. Ensure that you enter flags using the correct case because flag options are case sensitive. If you do not want a flag, enter NULL.

Details

Use POSIX standards when you enter regular expressions. POSIX refers to the POSIX.1 standard IEEE Std 1003.1, which defines system interfaces and headers with relevance for string handling and internationalization. The XPG3, XPG4, Single Unix Specification (SUS), and other standards include POSIX.1 as a subset. The patterns that we list in the following tables adhere to the current standard. For more information and updates, see "Regular Expressions" in the International Components for Unicode (ICU) User Guide at https://unicode-org.github.io/icu/userguide/ .

Use the regular expression patterns in the following table for the `<regular_expression_pattern>` argument.

Character	Description
\a	Match a BELL, \u0007.
\A	Match at the beginning of the input. Differs from ^ in that \A does not match after a new line within the input.
\b, outside of a [Set]	Match if the current position is a word boundary. Boundaries occur at the transitions between \w (word character or characters) and \W (nonword character or characters), with combining marks ignored. For better word boundaries, see ICU Boundary Analysis.
\b, within a [Set]	Match a BACKSPACE, \u0008.
\B	Match if the current position is not a word boundary.
\cX	Match a control-X character.
\d	Match any character with the Unicode General Category of Nd (Number, Decimal Digit).

Character	Description
\D	Match any character that is not a decimal digit.
\e	Match an ESCAPE, \u001B.
\E	Terminates a \Q ... \E quoted sequence.
\f	Match a FORM FEED, \u000C.
\G	Match if the current position is at the end of the previous match.
\n	Match a LINE FEED, \u000A.
\N{UNICODE CHARACTER NAME}	Match the named character.
\p{UNICODE PROPERTY NAME}	Match any character with the specified Unicode Property.
\P{UNICODE PROPERTY NAME}	Match any character not having the specified Unicode Property.
\Q	Quotes all following characters until \E.
\r	Match a CARRIAGE RETURN, \u000D.
\s	Match a white space character. White space is defined as [\t\n\f\r\p{Z}].
\S	Match a non-white space character.
\t	Match a HORIZONTAL TABULATION, \u0009.
\uhhhh	Match the character with the hex value hhhh.
\Uhhhhhhhh	Match the character with the hex value hhhhhhhh. Provide exactly eight hex digits, even though the largest Unicode code point is \U0010ffff.
\w	Match a word character. Word characters are [\p{Ll}\p{Lu} \p{Lt}\p{Lo}\p{Nd}].
\W	Match a nonword character.
\x{hhhh}	Match the character with hex value hhhh. From one to six hex digits may be supplied.
\xhh	Match the character with two digit hex value hh.
\X	Match a Grapheme Cluster.

Character	Description
\Z	Match if the current position is at the end of input, but before the final line terminator, if one exists.
\z	Match if the current position is at the end of input.
\n	Back reference. Match whatever the nth capturing group matched. n must be a number greater than 1 and less than the total number of capture groups in the pattern. Note: Octal escapes, such as \012, are not supported in ICU regular expressions.
[pattern]	Match any one character from the set. See Unicode Set for a full description of what may appear in the pattern.
.	Match any character.
^	Match at the beginning of a line.
$	Match at the end of a line.
\	Quotes the following character. Characters that require quotes to be treated as literals are * ? + [( ) { } ^ $ | \ . /

Use the regular expression operators in the following table for the `<regular_expression_pattern>` argument.

Operator	Description
|	Alternation. A|B matches either A or B.
*	Match 0 or more times. Match as many times as possible.
+	Match 1 or more times. Match as many times as possible.
?	Match zero or one time. Prefer one.
{n}	Match exactly n times.
{n,}	Match at least n times. Match as many times as possible.
{n,m}	Match between n and m times. Match as many times as possible, but not more than m.
*?	Match 0 or more times. Match as few times as possible.
+?	Match 1 or more times. Match as few times as possible.
??	Match zero or one time. Prefer zero.
{n}?	Match exactly n times.

Operator	Description
{n,}?	Match at least n times, but no more than required for an overall pattern match.
{n,m}?	Match between n and m times. Match as few times as possible, but not less than n.
*+	Match 0 or more times. Match as many times as possible when first encountered, do not retry with fewer even if overall match fails. Possessive match.
++	Match 1 or more times. Possessive match.
?+	Match zero or one time. Possessive match.
{n}+	Match exactly n times.
{n,}+	Match at least n times. Possessive match.
{n,m}+	Match between n and m times. Possessive match.
(...)	Capturing parentheses. Range of input that matched the parenthesized subexpression is available after the match.
(?: ...)	Non-capturing parentheses. Groups the included pattern, but does not provide capturing of matching text. Somewhat more efficient than capturing parentheses.
(?> ...)	Atomic-match parentheses. First match of the parenthesized subexpression is the only one tried; if it does not lead to an overall pattern match, back up the search for a match to a position before the "(?>"
(?# ...)	Free-format comment (?# comment).
(?= ...)	Look-ahead assertion. True if the parenthesized pattern matches at the current input position, but does not advance the input position.
(?! ...)	Negative look-ahead assertion. True if the parenthesized pattern does not match at the current input position. Does not advance the input position.
(?<= ...)	Look-behind assertion. True if the parenthesized pattern matches text preceding the current input position, with the last character of the match being the input character just before the current position. Does not alter the input position. The length of possible strings matched by the look-behind pattern must not be unbounded (no * or + operators).

Operator	Description
(?<! ...)	Negative look-behind assertion. True if the parenthesized pattern does not match text preceding the current input position, with the last character of the match being the input character just before the current position. Does not alter the input position. The length of possible strings matched by the look-behind pattern must not be unbounded (no * or + operators).
(?ismx-ismx: ...)	Flag settings. Evaluate the parenthesized expression with the specified flags enabled or disabled.
(?ismx-ismx)	Flag settings. Change the flag settings. Changes apply to the portion of the pattern following the setting. For example, (?i) changes to a case-insensitive match.

Use the flags in the following table for the `<flag>` argument.

Flag Options	Description
'CASE_INSENSITIVE'	If set, matching takes place in a case-insensitive manner.
'COMMENTS'	If set, allows use of white space and #comments within patterns.
'DOTALL'	If set, a "." in a pattern matches a line terminator in the input text. By default, it will not. Note that a carriage return or line feed pair in text behave as a single line terminator and match a single "." in a regular expression pattern.
'MULTILINE'	If set, the function treats the input string as multiple lines instead of a single line. The '^' and '$' characters apply to each line in the input string instead of the entire input string.
'NO_PUSHDOWN'	If set, the match_regex function is processed but is never pushed down to the database.

Example

Use the match_regex function in the Validation transform by accessing the Smart Editor or function wizard or in a WHERE clause of a Query transform. The input string can be from sources such as columns, variables, or constant strings.

Use Case	Pattern	Function Call
To match phone numbers in (408)-933-6000 format.	'([0-9]{3}-[0-9]{3}-[0-9]{4}'	`match_regex (pho_number,'([0-9]{3}-[0-9]{3}-[0-9]{4}',NULL)`

Use Case	Pattern	Function Call
To match a string that starts with "topicA" regardless of case.	'topicA.*'	`match_regex (subject, 'topicA.*','CASE_INSENSITIVE')`

6.3.5.70 match_simple

Use the match_simple function to match a whole input string to simple patterns supported by the software for this function.

Syntax

```
match_simple(<input_string>,<pattern_string>)
```

Return Value

int

Returns:

- 1 = Pattern matches
- 0 = Pattern does not match

Where

`<input_string>`	String to be matched. Supports UNICODE characters.
`<pattern_string>`	Pattern you want to find in a whole input string.

Details

The match_simple function does not match substrings.

Use the characters in the following table for `<pattern_string>`.

.	Represents any single character.
*	Represents any character zero or more times.

#	Represents any single alphabetic character including non-English letters.
$	Represents any alphabetic character, including non-English letters, zero or more times.
+	Matches the previous character one or more times.
(string)+	Matches the string one or more times.
[number1..number2]	Numeric range (integers only). Matches any number between number1 and number2.
\	Escape character
;	OR operator. If the data matches any of the identified patterns, the result is TRUE. Enclose the list with curly brackets {}. Example: `{ABC+;XYZ*}` If the data matches either ABC+ or XYZ*, the result is TRUE.
<>	NOT operator. Specify the pattern after the <>. Example: `<><pattern>`
{EMPTY} and {empty}	Special predefined patterns that match empty data.
{NULL} and {null}	Special predefined patterns that match NULL data.

If the pattern is empty, then the function matches all data.

If the value of a pattern column is NULL, then the function does not match with any value.

All other characters represent themselves. If you want to specify a special character as itself, then use an escape character.

Example

Example patterns

Example Value	Pattern string
ACCT1234567	ACCT*
ZIP10000 to ZIP99999	ZIP[10000..99999]
ACCT123 or ACCOUNT234	{ACCT*;ACCOUNT*}
www.anything.com	www.$.com

6.3.5.71 max

Use the max function to return the maximum value from a list.

Syntax

```
max(<value_list>)
```

Return value

Any type

The maximum value of the column values. The return type is the same as the values in `<value_list>`.

Where

`<value_list>`	The source values for which to identify a maximum.

Details

Example

To calculate the maximum value in the salary column of a table, use the max function in a query:

- In the *Mapping* tab of the query editor, enter:

  ```
  max(SALARY)
  ```

- In the *Group By* tab in the query editor, specify the columns for which you want to find the maximum salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the maximum salary.

6.3.5.72 min

Use the min function to return the minimum value from a list.

Syntax

```
min(<value_list>)
```

Return value

Any type

The minimum value of the column values. The return type is the same as the values in `<value_list>`.

Where

`<value_list>`	The source values for which to identify a minimum.

Details

> **Example**
>
> To calculate the minimum value in the salary column of a table, use the min function in a query:
>
> - In the *Mapping* tab of the query editor, enter:
>
>  ```
>  min(SALARY)
>  ```
>
> - In the *Group By* tab in the query editor, specify the columns for which you want to find the minimum salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the minimum salary.

6.3.5.73 mod

Use the mod function to return the remainder when one number is divided by another.

> **Syntax**
>
> ```
> mod(<numerator>, <denominator>)
> ```

Return Value

integer

Where

`<numerator>`	Integer to be divided.
`<denominator>`	Divisor of first integer.

Details

> **Note**
>
> The `%` operator from SAP Information Steward syntax produces the same result.

> **Example**

Function	Result
`mod(10,3)`	1
`mod(17,5)`	2
`mod(10,5)`	0

6.3.5.74 month

Use the month function to determine the month in which the given date falls.

> **Syntax**

```
month(<date1>)
```

Return value

int

The number from 1 to 12 that represents the month component of `<date1>`.

Where

`<date1>`	The source date.

Details

Example

Function	Results
`month(to_date('Jan 22, 1997', 'mon dd, yyyy'))`	1
`month(to_date('3/97', 'mm/yy'))`	3

6.3.5.75 nvl

Use the nvl function to replace NULL values with a given value.

Syntax

```
nvl(<expression1>, <replacement_value>)
```

Return value

Any type

The value of `<expression1>` if not NULL, otherwise, the value of `<replacement_value>`.

Where

`<expression1>`	The value to be tested for NULL.
`<replacement_value>`	The value to replace `<expression1>` if `<expression1>` is NULL. Ensure that `<replacement_value>` is the same data type as `<expression1>`.

Details

Example

Function	Results
`nvl(modification_date, sysdate())`	If the column `modification_date` for a row hasn't been set, this function inserts the current date.
`nvl(lookup(r3..vbpa, kunnr, 'NULL', vbeln, vbak.vbeln, posnr, vbap.posnr, parvw, 'RE'), lookup(r3..vbpa, kunnr, 'NULL', vbeln, vbak.vbeln, posnr, vbap.posnr, parvw, 'RG'))`	Both expressions are determined by the result of lookup functions.

6.3.5.76 power

Use the power function to return the value of the given expression to the specified power.

Syntax

```
power(<num>, <num>)
```

Return Value

Float data type

Where

`<num>`	Numeric expression representing a base number.
`<num>`	Numeric expression representing the power.

Details

Example

Function	Results
`power(2.2,3));`	10.648000

6.3.5.77 previous_row_value

Use the previous_row_value function to return the column value of the previous row.

Syntax

```
previous_row_value(<expression>)
```

Return Value

Data type of the input parameter. First row always returns `NULL`.

Where

`<expression>`	Valid Input expression.

Details

Each call to the previous_row_value() function returns the value stored during the previous call of this function. If the function is not called for each row, the results of this function might not be what you expect because it may not be the previous row value.

Example

This scenario can happen for example, if you use the previous_row_value() inside an ifthenelse() function: `If_then_else (table1.status = 'new', 0 , previous_row_value(table1.value))`

A better solution to the scenario is to use the following expression: `If_then_else (table1.status = 'new', 0 , 1) * previous_row_value(table1.value)`

Alternately, use two queries: One for the previous_row_value() and one for the final result including the if_then_else().

Example

The previous_row_value function is useful in Query transform. For example, the input stream of the column might be `1;2;3;4` for the first four rows. The function returns `NULL;1;2;3`.

Example

The following is a list of records of sales figures for a series of days. Each record lists the record number, date, and revenue.

Date Revenue

rec 1 1/1/2005 1000
rec 2 1/2/2005 1100
rec 3 1/3/2005 900
rec 4 1/4/2005 1200

The requirement is to calculate the delta of the revenue with the previous day. So the query uses "order by Date" and subtracts the previous row revenue from the current row revenue.

Results:

Date Revenue Delta = Revenue - Previous_Row_Value

rec 1 1/1/2005 1000 NULL
rec 2 1/2/2005 1100 +100
rec 3 1/3/2005 900 -200
rec 4 1/4/2005 1200 +300

6.3.5.78 print

Use the print function to print a given string to the trace log.

Syntax

```
print('<input_string>')
```

Return value

int

Value is `<input_string>` when the string contains valid data. Value is NULL and no string prints when the string contains NULL data.

Where

`<input_string>`	The message to be written to the trace log.

Details

Example

Function	Results
`print('Reached decision point for running full or incremental data flows')`	Writes "Reached decision point for running full or incremental flows" to trace log and returns `<input_string>`.
`print('The date is: [$start_date]')`	Writes "The date is 2000.06.03" to trace log and returns `<input_string>`.
`print('[$month_sal*12]')`	Writes "48000" to trace log and returns `<input_string>`.
`print('Total Sal is: [$month_sal*12]');`	Writes "Total Sal is: 48000" to trace log and returns `<input_string>`.
`print('The return value from the SQL() function is > [$y]');`	Writes "The return value from the SQL() function is > 23456" to trace log and returns `<input_string>`.

6.3.5.79 quarter

Use the quarter function to determine the quarter in which the given date falls.

Syntax

```
quarter(<date1>)
```

Return value

int

The number from 1 to 4 that represents the quarter component of `<date1>`.

Where

`<date1>`	The source date.

Details

Example

Function	Results
`quarter(to_date('Jan 22, 1997', 'mon dd, yyyy'))`	1
`quarter(to_date('5/97', 'mm/yy'))`	2

6.3.5.80 raise_exception

Use the raise_exception function to generate an exception message for the Job Server error log..

Syntax

```
raise_exception(<error_msg>)
```

Return Value

int

Always returns 1.

Where

`<error_msg>`	The string that the software writes to the Job Server error log.

Details

If you surround the function with a try—catch block, the work flow or job may or may not terminate based on how you set the block.

Example

```
ifthenelse(sal < 1000000, 0, raise_exception('Salary exceeds 1 million
dollars.'))
```

6.3.5.81 raise_exception_ext

Use the raise_exception_ext function generates an exception with an exit code.

Syntax

```
raise_exception_ext(<error_msg>, <exit_code>)
```

Return Value

int

Always returns 1.

Where

`<error_msg>`	The string that the software writes to the Job Server error log.
`<exit_code>`	If the exception is not caught in a try—catch block, the code with which the job exits. Use a number in range 1 to 255 (zero means "success" to all operating systems).

Details

The software may or may not terminate the work flow or job may based on whether a try-catch block surrounds the call.

Example

```
ifthenelse(sal < 1000000, 0, raise_exception_ext('Salary exceeds 1 million
dollars.', sal/1000000 + 1))
```

6.3.5.82 rand

Use the rand function to return a random number between 0 and 1.

Syntax

```
rand()
```

Return value

real

The random number between 0 and 1.

Example

Function	Results
`100 * rand()`	The function multiplies the random number by 100. The result is a random number between 0 and 100.

6.3.5.83 rand_ext

Use the rand_ext to return a random number between 0 inclusive and 1 exclusive.

Syntax

```
real rand_ext(<seed>)
```

Return value

real

The random number. The return value is between 0 and 1.

Where

`<seed>`	(Optional) Can be any positive integer greater than or equal to 0. If unspecified, the software uses the current time to create a seed.

Details

Similar to, and more powerful than the rand function. This function uses the linear-congruential generator (LCG) algorithm:

$x_n = (ax_{n-1} + b) \bmod m$ where:

x_n is an integer from 0 to m-1 and the initial value of x_n is called the "seed" (x_0).

For each call to the random number generator, the software calculates a new x_n by taking the value of the previous result x_{n-1}, multiplying by a, adding b, then taking the remainder mod m.

SAP Cloud Integration for data services uses this formula to generate an integer from 0 to m-1. After SAP Cloud Integration for data services calculates x_n, it divides that number by m to obtain a number equal to or greater than 0 and less than 1.

By specifying the same seed number, you can regenerate an exact number sequence. Specifying the same seed number is useful in repeat experiments.

Example

Function	Results
`100 * rand_ext()`	A random number between 0 and 100.

6.3.5.84 replace_substr

Use the replace_substr function to replace each occurrence of a specified substring with a different substring.

Syntax

```
replace_substr(<in_str>, <search_str>, <replace_str>)
```

Return value

varchar

Where

`<in_str>`	The input string that contains the substring to be changed. If `<in_str>` is NULL, the software returns NULL.
`<search_str>`	Substring to be replaced. If `<search_str>` is NULL, the software returns the string in `<in_str>`.
`<replace_str>`	Substring to use in place of `<search_str>`. If `<replace_str>` is blank or NULL, the software removes all occurrences of `<search_str>` from the `<in_str>`.

Details

Example

Function	Result
`replace_substr('a penny saved is a penny earned', 'penny', 'million')'`	Replaces the word "penny" with "million." `'a million saved is a million earned'`

6.3.5.85 replace_substr_ext

Use the replace_substr_ext function to replace each occurrence of a specified substring with a replacement string. The specified substring can contain hexadecimals that refer to a UNICODE character, or non printable character references such as form feed or new line.

Syntax

```
replace_substr_ext(<in_str>, <search_str>, <replace_str>,
<start_at_occurance>, <number_of_occurances>)
```

Return Value

varchar

Where

in_str	The input string that contains the substring to be changed. If <in_str> is NULL, the software returns NULL.
search_str	Substring to be replaced. If <search_str> is NULL, the software returns the string in <in_str>. You can use /x0000 to specify the hexadecimal value for a special character. For example, if you use /x000A, then if SAP Cloud Integration for data services encounters /x it converts the next 4 characters to a hexadecimal value. This function converts the hexadecimal value to a UNICODE character. This option provides more flexibility when you use a search string. You can also represent special characters using the escape character '/'. The software supports the following characters: /a Bell (alert) /b Backspace /f Formfeed /n New line /r Carriage return /t Horizontal tab /v Vertical tab To include the escape character '/' in the search string, escape it using '//'. For example, if the input is 'abc/de', SAP Cloud Integration for data services converts search_str to 'abcde'. If the input is 'abc//de', SAP Cloud Integration for data services converts search_str to 'abc/de'. If search_str is NULL, SAP Cloud Integration for data services returns a varchar with the data in in_str.
replace_str	Substring to use in place of <search_str>. If <replace_str> is blank or NULL, the software removes all occurrences of <search_str> from the <in_str>.
start_at_occurrence	Occurrence of the <search_str> with which to start replacing. If NULL, start at the 1st occurrence. For example, enter 2 to replace or remove the second occurrence of a search_str.
number_of_occurrences	Number of occurrences to replace. If NULL, replace all occurrences. For example, enter 2 to replace or remove two sequential occurrences of the search_str.

Details

Example

Function	Result
`replace_substr_ext('ayyyayyyayyyayyy', 'a', 'B', 2, 2)'`	Replaces 'a' with 'B' starting from second occurrence and replaces two occurrences. 'ayyyByyyByyyayyy'
`replace_substr_ext('ayyya</n>yyya</n>yyyayyy', 'a/n', 'B', 2, 2)`	Searches a string containing 'a' followed by a new line; replaces it with 'B' starting from second occurrence; replaces two occurrences. 'ayyyByyyByyyayyy'
`replace_substr_ext('ayyya</n>yyya</n>yyyayyy', 'a/x000a', 'B', 2, 2)`	Searches a string containing 'a' followed by a new line; replaces it with 'B' starting from second occurrence; replaces two occurrences. 'ayyyByyyByyyayyy'

6.3.5.86 round

Use the round function to round a given number to a specified precision.

Syntax

```
round(<num1>, <precision>)
```

Return value

decimal, double, int, or real

The rounded number using the same data type as the original number, `<num1>`.

Where

`<num1>`	The source number.

`<precision>`	An integer indicating the number of decimals in the result. If `<precision>` is negative, the software rounds the digits left of the decimal point.

Details

Example

Function	Results
`round(120.12345, 2)`	120.12
`round(120.12999, 2)`	120.13
`round(120, -2)`	100
`round(120.123, 5)`	120.12300

6.3.5.87 rpad

Use the rpad function to pad a string of characters from a given pattern.

Syntax

```
rpad(<input_string>, <size>, '<pad_string>')
```

Return value

varchar

The new string.

Where

`<input_string>`	The source string.
`<size>`	An integer value indicating the number of characters in the resulting string.

`<pad_string>`	A character or set of characters that this function concatenates to `<input_string>`.

Details

The function repeats the pattern at the end of the input string until the final string is the appropriate length. If the input string is already longer than the expected length, the function truncates the string.

Example

Function	Results
`rpad('Tanaka',15,' ')`	`'Tanaka '`
`rpad(last_name,25,' ')`	The value in the column `last_name`, padded with spaces to 25 characters, or truncated to 25 characters.

6.3.5.88 rpad_ext

Use the rpad_ext function to pad a string with logical characters from a given pattern.

Syntax

```
rpad_ext(<input_string>, <size>, '<pad_string>')
```

Return value

varchar

The new string.

Where

`<input_string>`	The source string.
`<size>`	An integer value indicating the number of characters in the resulting string.

`<pad_string>`	A logical character or set of logical characters that this function concatenates to `<input_string>`.

Details

> **Note**
>
> The logical characters prohibit this function from getting pushed down to an Oracle database.

The function repeats the pattern at the end of the input string until the final string is the appropriate length. If the input string is already longer than the expected length, this function truncates the string.

> **Example**

Function	Results
`rpad_ext('Tanaka',15,' ')`	`'Tanaka '`
`rpad_ext(last_name,25,' ')`	The value in the column `last_name`, padded with spaces to 25 characters, or truncated to 25 characters.

The rpad_ext and rpad functions exhibit the same behavior when the software evaluates the functions. In situations where the function is pushed down to the database, the database behavior may differ when `<input_string>` and—or `<pad_string>` parameters contain multibyte characters.

Function	Input	Output
rpad	("abc¶¶", 10,' ')	`'abc¶¶ '`
rpad_ext	("abc¶¶", 10,' ')	`'abc¶¶ '`
rpad	("abc¶¶", 10,'')	`'abc¶¶'`
rpad_ext	("abc¶¶", 10,'')	`'abc¶¶'`
rpad	("abcd", 10, '¶')	`'abcd¶¶¶'`
rpad_ext	("abcd", 10, '¶')	`'abcd¶¶¶¶¶¶'`

6.3.5.89 rtrim

Use the rtrim function to remove specified characters from the end of a string.

> **Syntax**

```
rtrim('<input_string>', '<trim_string>')
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`	The string to be modified.
`<trim_string>`	The characters to remove from `<input_string>`.

Details

The function scans `<input_string>` from right to left removing all characters that appear in `<trim_string>` until it reaches a character not in `<trim_string>`.

Removes trailing blanks only if `<trim_string>` contains trailing blanks. If the length of the modified string becomes zero after trimming, the function returns '' (empty string).

To remove all trailing blanks in a string, use the rtrim_blanks function.

Example

Function	Results
`rtrim('Marilyn ', ' ')`	'Marilyn'
`rtrim('ZABCABC', 'ABC')`	'Z'
`rtrim('ZABCABC', 'EFG')`	'ZABCABC'

You may also use the rtrim_blanks or rtrim_blanks_ext functions for this.

6.3.5.90 rtrim_blanks

Use the rtrim_blanks function to remove blank characters from the end of a string.

Syntax

```
rtrim_blanks(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`	The string to be modified.

Details

If the length of the modified string becomes zero after trimming, the function returns '' (empty string).

Example

Function	Results
`rtrim_blanks('Marilyn ')`	'Marilyn'
`rtrim_blanks(last_name)`	The value contained in the column `last_name` with trailing blanks removed.

6.3.5.91 rtrim_blanks_ext

Use the rtrim_blanks_ext function to remove blank and control characters from the end of a string.

Syntax

```
rtrim_blanks_ext(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`	The string to be modified.

Details

If the length of the modified string becomes zero after trimming, the function returns '' (empty string).

Example

Function	Results
`rtrim_blanks('Marilyn ')`	'Marilyn'
`rtrim_blanks(last_name)`	The value contained in the column `last_name` with trailing blanks and control characters removed.

6.3.5.92 save_data

Use the save_data function to create and store a persistent variable with a name, which could be the task name or any other string, and any piece of data. This data could be the end date timestamp of the most current load.

Syntax

```
save_data ('<task_name>', <date>)
```

Where

`<task_name>`	The name created for the task.
`<date>`	The most current load date.

Details

Both `<task_name>` and `<date>` must be varchar. The maximum data size is 255 characters.

Example

Functions	Results
`save_data ('hello_world', to char(sysdate(), 'yyyy-mm-dd hh24:mi:ss'))`	SAP Cloud Integration for data services saves the most current load date of hello_world.

Related Information

Change Data Capture (Delta Loads) [page 158]

6.3.5.93 sleep

Use the sleep function to suspend the execution of the calling data flow or work flow.

Syntax

```
sleep(<num_millisecs>)
```

Return Value

int

Always returns 1.

Where

`<num_millisecs>`	The number of milliseconds to "sleep".

Details

Calling this function causes the thread that executes this function to halt operations for the given number of milliseconds. To force a task or process to halt operations until a condition becomes true, call this function in a work flow, not in a data flow.

Example

The following example invokes sleep for one second when a file exists in a directory called 'c'.

```
while (file_exists('c:/temp.msg') == 0)
begin
sleep(1000);
end
```

6.3.5.94 sqrt

Use the sqrt function to return the square root of the given expression.

Syntax

```
sqrt(<num>)
```

Return Value

Float

Where

<num>	The number for which you want the square root.

Details

Return value is NULL if the input is negative.

Example

Function	Results
`sqrt(625.25);`	25.005000

6.3.5.95 substr

Use the substr function to return a specific portion of a string starting at a given point in the string.

Syntax

```
substr(<input_string>, <start>, <length>)
```

Return value

varchar

The modified string. The return data type is the `<input_string>`. If the length is a constant, then it is a varchar of the given length.

Where

`<input_string>`	The string to be modified.
`<start>`	The position in the `<input_string>` where the function obtains the first character of the new string. The function counts characters from the beginning of `<input_string>`. • In normal data flows, the first character is position number 1. • If `<start>` is 0, the new string begins with the first character (position 1). • If `<start>` is negative, the function counts characters from the end of `<input_string>`. The new string begins with the character in that position from the end of the string. The function returns NULL or an empty string under the following circumstances: • If `<start>` is greater than the number of characters in `<input_string>`, the function returns NULL. • If `<length>` is less than 1, the function returns an empty string.

`<length>`	The number of characters in the resulting string. • If `<length>` is 0 or negative, the function returns an empty string. • If `<length>` is greater than the number of characters remaining in `<input_string>` after `<start>` , the function returns only the remaining characters. The function keeps the trailing blanks in the remaining `<input_string>` after `<start>`.

For information about how Data Services uses the substr function with HANA, see SAP Note 2808903.

Details

Example

Function	Results
`substr('94025-3373', 1, 5)`	'94025'
`substr('94025-3373', 7, 4)`	'3373'
`substr('94025', 7, 4)`	NULL
`substr('Dr. Schultz', 4, 18)`	'Schultz'
`substr('San Francisco, CA',-4, 18)`	', CA'

6.3.5.96 sum

Use the sum function to calculate the sum of a given set of values.

Syntax

```
sum(<value_list>)
```

Return value

decimal, double, int, or real

The total of the values. The return type is the same as the values in `<value_list>`.

Where

`<value_list>`	The source values to sum.

Details

Example

To calculate the sum of values in the salary column of a table, use the sum function in a query:

- In the *Mapping* tab of the query editor, enter:

  ```
  sum(SALARY)
  ```

- In the *Group By* tab in the query editor, specify the columns for which you want to find the total salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the sum of the salary.

6.3.5.97 sy

Returns the value of an SAP system variable at run time. This function is only available through query transforms in ABAP data flows.

Syntax

Syntax

```
sy('<SAP_variable>')
```

Return value

varchar(255): The value of the SAP system variable. You may need to recast the return value to the actual data type of the system variable in SAP.

Where

`<SAP_variable>`: A string value containing the name of the SAP system variable. This value is not case sensitive. Enclose the name in single quotation marks (').

When the sy function is executed, the software generates the appropriate function call in the ABAP for the ABAP data flow (appends `SY-` to the `<SAP_variable >` that you specify) and returns the result of the function call at run time.

The table SYST in SAP lists the available system variable, their data types, and descriptions.

If the given `<SAP_variable >` does not exist in SAP, a run-time error will be returned:

```
ABAP program <Generated ABAP Program> syntax error: <The data object "SY" has no
component called "ABC">.
```

No value is returned when this error occurs.

Example: Functions and results

Function	Results
`sy('SUBRC')`	0 if the subroutine call returned successfully
`sy('mandt')`	The client ID number of the current SAP application client
`sy('UNAME')`	The login name of the user
`sy('DATUM')`	The current date from the SAP application server
`sy('UZEIT')`	The current time from the SAP application server
`sy('TCODE')`	The current transaction code

6.3.5.98 sysdate

Use the sysdate function to return the current date as listed by the system.

Syntax

```
sysdate()
```

Return value

date

Today's date.

Details

Returns the current date as listed by the operating system of the server where the Agent is installed.

> **Note**
>
> The value that the sysdate function returns is a datetime value. Internally SAP Cloud Integration for data services reads both the date and the time when it runs a sysdate function. The data that is used by the task depends on the data type of a particular column. For example, if the data type of a column in a query is `date`, SAP Cloud Integration for data services only uses the date for calculations. The time data is ignored. If you change the data type to `datetime`, both a date and a time are used.

> **Example**

Function	Results
`isweekend(sysdate())`	Tests whether today is a Saturday or Sunday.
`to_char(sysdate(), 'yyyy.mm.dd')`	Converts the sysdate function `datetime` value to a string that displays only the date. Use this, for example, to exclude part of the `datetime` data by providing only a format for the data to display in a report. To convert a `datetime` value to a string containing only the date, use this expression and change the column data type to `varchar`.

6.3.5.99 systime

Use the systime function to return the current time as listed by the system.

> **Syntax**

```
systime()
```

Return value

time

The current time.

Details

Returns the current time as listed by the operating system of the server where the Agent is installed.

6.3.5.100 sysutcdate

Use the sysutcdate function to return the current UTC date as listed by the operating system of the server where the Agent is installed.

> **i** Note
>
> The value that the sysutcdate function returns is a UTC datetime value. Internally SAP Cloud Integration for data services reads both the date and the time when it runs a sysutcdate function. The data that is used by the task depends on the data type of a particular column. For example, if the data type of a column in a query is `date`, SAP Cloud Integration for data services only uses the date for calculations. The time data is ignored. If you change the data type to `datetime`, both a date and a time are used.

> Syntax
>
```
sysutcdate()
```

Return value

date

Today's date.

Example

Function	Results
`isweekend(sysutcdate())`	Tests whether today (UTC) is a Saturday or Sunday.

Function	Results
`to_char(sysutcdate(), 'yyyy.mm.dd')`	Converts the sysutcdate function's `datetime` value to a string that displays only the date (UTC). For example, you can use this to exclude part of the `datetime` data by only providing a format for the data you want to display in a report. To convert a `datetime` value to a string containing only the date, use this expression and change the column's data type to `varchar`.

6.3.5.101 to_char

Use the to_char function to convert a date or numeric data type to a string.

Syntax

```
to_char(<date or numeric_expression>,'<format>')
```

Return

varchar

A formatted string that describes the specified `<numeric_expression>`.

Where

`<numeric expression>`	The source int, real, double, or decimal data value.

`<format>`

A string indicating the format of the generated string.

> **Note**
>
> Provide format to ensure correct results.

Choose from the codes listed in the following Format codes table.

Table 30: Format codes

Format	Description	Example
9	Number. Suppresses leading and trailing zeros from the results. Includes a leading minus sign (-) for negative numbers or one leading space for pos numbers.	to_char(123,'9999') = ' 123'
0	Number. Includes leading and trailing zeros.	to_char(123,'09999') = ' 0123' to_char(123,'9999D.00') = ' 123.00'
D<.|,>	Sets position of decimal point followed by character to use as decimal separator. Currently the software supports only dot (.) and comma (,) as decimal separators.	to_char(12.34,'99D.99') = ' 12.34'
G<.|,|space >	Position of group separator followed by character to be used as group separator. Currently the software supports only dot (.), comma (,), and space (' ') as group separators.	to_char(1234,' 9G,999') = ' 1,234'
x	Lower case "x." String containing unsigned hexadecimal integer, using "abcdef". If the number is not 2 bytes long, the software does not pad the output.	to_char(123,'xx') = ' 7b' to_char(12,'x') = ' c'
X	Upper case "X." String containing unsigned hexadecimal integer, using "ABCDEF". If the number is not 2 bytes long, the software does not pad the output.	to_char(123,'XX') = ' 7B' to_char(12,'X') = ' C'

Format	Description	Example
0	String containing unsigned octal integer. This option is not case-sensitive. If the number is not 2 bytes long, the software does not pad the output.	to_char(12,'oo') = ' 14' to_char(1,'o') = ' 1'

Where

> **i Note**
>
> The to_char function supports the Oracle 9i timestamp data type up to 9 digits precision for sub-seconds.

`<date>`	The source date, time, or datetime value.
`<format>`	A string indicating the format of the generated string. Choose from the following codes: DD: 2-digit day of the month. MM: 2-digit month. MONTH: Full name of month. MON: 3-character name of month. YY: 2-digit year. YYYY: 4-digit year. HH24: 2-digit hour of the day (00-23). MI: 2-digit minute (00-59). SS: 2-digit second (00-59). FF: Up to 9-digit sub-seconds. Other values included in `<format>` appear unchanged in the result.

Details

Example

Function	Results
`to_char(call_date, 'dd-mon-yy hh24:mi:ss.ff')`	The date value from the call_date column formatted as a string. Result: `28-FEB-97 13:45:23.32`

The software reproduces the hyphens and spaces in the `<format>` parameter. The software recognizes all the other characters as part of a parameter string from the Date string table and substitutes with appropriate current values.

6.3.5.102 to_date

Use the to_date function to convert an input string to a date type based on the input format.

Syntax

```
to_date(<input_string>,'<format>')
```

Return value

date, time, or datetime

A date, time, or both representing the original string.

Where

`<input_string>`	The source string.

`<format>`	A string indicating the format of the source string. Choose from the following codes: DD: 2-digit day of the month MM: 2-digit month MONTH: Full name of month MON: 3-character name of month YY: 2-digit year YYYY: 4-digit year HH24: 2-digit hour of the day (00-23) MI: 2-digit minute (00-59) SS: 2-digit second (00-59) FF: Up to 9-digit sub-seconds **Note** Ensure that you set a format. If you do not set a format, the results may be incorrect.

Details

If the input string has more characters than the format string, the software ignores the extra characters in the input string and initializes to the default value.

Example

The software converts the following expression but ignores and initializes the extra characters to zero in the time part of the input string:

`to_date('10.02.2007 13:25:45', 'DD.MM.YYYY')` converts to `10.02.2007 00.00.00`

This function also supports the Oracle 9i `timestamp` data type. Its precision allows up to 9 digits for sub-seconds.

Example

Function	Results
`to_date('Jan 8, 1968', 'mon dd, yyyy')`	`1968.01.08` stored as a date.

6.3.5.103 to_decimal

Use the to_decimal function to convert a varchar to a decimal.

Syntax

```
to_decimal('<in_str>','<decimal_sep>','<thousand_sep>',<scale>)
```

Return Value

decimal

Uses a precision of 28 and the given scale.

Where

`<in_str>`	The number string. Null implies a NULL return. If `<in_str>` is invalid, the software returns a 0.
`<decimal_sep>`	The character that separates the decimal component from the whole number component.
`<thousand_sep>`	The character that separates thousands from hundreds in the whole number component.
`<scale>`	The number of digits to the right of the decimal point in the returned value.

Details

Example

Function	Result
`to_decimal('99,567.99', '.', ',', 3)`	`99567.990`

6.3.5.104 to_decimal_ext

Use the to_decimal_ext function to convert a varchar to a decimal and includes precision as a parameter.

Syntax

```
to_decimal_ext('<in_str>','<decimal_sep>','<thousand_sep>',<precision>,<scale>
)
```

Return Value

decimal

Uses the given precision and scale.

Where

`<in_str>`	The number string. Null implies a NULL return.
`<decimal_sep>`	The character that separates the decimal component from the whole number component.
`<thousand_sep>`	The character that separates thousands from hundreds in the whole number component.
`<precision>`	The total number of digits in the returned value.
`<scale>`	The number of digits to the right of the decimal point in the returned value.

Details

The to_decimal_ext function supports the use of DECIMAL data types with up to 96 precision.

Example

Function	Result
`to_decimal_ext('99,567.99', '.', ',',38,3)`	`99567.990`

6.3.5.105 translate

Use the translate function to translate selected characters of an input string into other specified characters.

Syntax

```
translate(<input string>, <from string>, <to string>)
```

Return Value

String

Returns the input string translated in the following way: The software replaces all occurrences of each character in the `<from string>` with the corresponding character in the `<to string>`.

Where

`<input string>`	The string to be translated.
`<from string>`	The characters to be replaced from the `<input string>`. • The software does not replace the characters in the `<input string>` that are not in the `<from string>`. • The software removes the characters in the `<from string>` that do not have a corresponding character in the `<to string>` from the `<input string>`.
`<to string>`	The corresponding characters to replace the characters in `<from string>`.

Details

If the `<from string>` or `<to string>` is null, then the software returns null. This function is case sensitive with parameter values.

Example

Function	Results
`translate('Business Objects','sne','xyz')`	'Buxiyzxx Objzctx'

Function	Results
`translate('Business Objects','st','x')`	'Buxinexx Objecx'
`translate('Business Objects','s','')`	'Buine Object'
`translate('Business Objects','abcd',NULL)`	NULL

6.3.5.106 trunc

Use the trunc function to truncate a given number to the specified precision without rounding the value.

Syntax

```
trunc(<num1>, <precision>)
```

Return value

decimal, double, int, or real

The truncated number. The return type is the same as the original number, `<num1>`.

Where

`<num1>`	The source number.
`<precision>`	An integer indicating the number of decimals in the result. If `<precision>` is negative, digits to the left of the decimal point are truncated and the value is padded with zeros.

Details

Example

Function	Results
`trunc(120.12345, 2)`	`120.12`
`trunc(120.12999, 2)`	`120.12`
`trunc(180, -2)`	`100`
`trunc(120.123, 5)`	`120.12300`

6.3.5.107 upper

Use the upper function to change the characters in a string to uppercase.

Syntax

```
upper(<value>,'<locale>')
```

Return value

varchar

The uppercase string. The return type is the same as `<value>`. The software does not change the characters that are not letters.

Where

`<value>`	The string to be modified.
`<locale>`	Optional. A locale to which the function converts the string. **Note** The software supports ISO 639 language code and ISO 3166 country code formats.

Details

Example

Function	Results
`upper('Accounting101')`	`'ACCOUNTING101'`
`upper(substr(LastName,1,1))|| lower(substr(LastName,2,LENGTH(LastName )))`	The value in column `LastName` with the first letter uppercase and the rest of the value lowercase. Note that this example does not account for last names with two words.
`upper(LastName, 'tr')`	The value in column `LastName` is converted to all uppercase. It is also converted to the Turkish locale, using the ISO 639 language code.

6.3.5.108 utc_to_local

Use the utc_to_local function to convert an input that is in Coordinated Universal Time (UTC) to the set time zone value.

Syntax

```
utc_to_local(<input datetime>, <timezone to convert with UTC offset>)
```

Return Value

datetime

Details

Converts the input in UTC to the desired time zone value. The second parameter UTC offset is a constant value. If the UTC offset is not provided, then the software uses the time zone of the agent host to calculate the UTC offset.

Example

Function	Results
`utc_to_local('2014.01.31 15:30:00', 'UTC+08:30')`	'2014.02.01 00:00:00'

6.3.5.109 wait_for_file

Use the wait_for_file function to look for a specified file pattern in a file system, polling for the file at intervals, until the job timeout is reached.

Syntax

wait_for_file (file_name_pattern, timeout, poll_interval,max_match, file_name_list, list_size, list_separator)

Return Values

int

Values are:

- 0 - No file matched.
- 1 - At least one file is matched.
- -1 - Timed out.
- -2 - At least one input value is illegal.

Where

`<file_name_pattern>`	The file name and path, relative to where the Agent is running. It can be an absolute or relative path. File name can contain wildcard characters.
`<timeout>`	Wait timeout in milliseconds. • If timeout is 0, then the function doesn't block. • If timeout is -1, then the function waits indefinitely for at least one file to exist that matches the file pattern. If you enter any other negative value, the software considers it illegal. On a computer where millisecond timing accuracy isn't available, timeout is rounded up to the nearest legal value available on that system.
`<poll_interval>`	Polling interval in milliseconds to look for the existence of the file. On a computer where millisecond timing accuracy isn't available, the polling interval is rounded up to the nearest legal value available on that system. If the poll interval exceeds the timeout value, then, it is rounded up to time out value.
`<max_match >`	Optional. Specifies the maximum number of matched file names that the function returns. The default value is 0. -1 specifies that the function return all the matched file names.

`< file_name_list >`	Optional. Output varchar variable that returns the list of matched file names. Order of the file names in the list is determined by the way the operating system returns the file names.
`< list_size>`	Optional. Output integer variable that returns the list size.
`<list_separator>`	Optional. File name list separator character(s). Default value is comma (,).

Details

This function looks for the specified file pattern in the file system. If it doesn't find the file(s), it waits for the specified timeout period, polling for the file(s) at every polling interval. The value specified in poll_interval determines how often to poll for the file pattern until timeout is reached. After timeout, the task or process stops, and polling for the file ceases.

This function waits a maximum of up to timeout interval for at least one file to exist that matches the pattern. Poll interval determines how often to poll for files.

Example

This function is used in a script at the beginning of a task. In a process, the script containing this function is often added right before a source file. A task or process suspends until a file is present, as shown in the following business use case example:

During the night, an external program puts source files in a central location that SAP Cloud Integration for data services can access. The process is usually complete at 1:00 AM or later. Tonight, however, you schedule the job to start at 1:00 AM. You include a script in the first step of the job that checks for the existence of the last file. If the last file doesn't exist, the job waits for an interval of time and tries again. Once the file is present, the job finds the file and continues with the rest of the process. You set a timeout so that the job stops if the file is still not found at 9:00 tomorrow morning.

6.3.5.110 week_in_month

Use the week_in_month function to determine the week number of the month in which the given date falls.

Syntax

```
week_in_month(<date1>)
```

Return value

int

The number from 1 to 5 that represents which week in the month that `<date1>` occurs.

This function considers the first week of the month to be first seven days. The day of the week is ignored when calculating the weeks.

Where

`<date1>`	The source date.

Details

Example

The following examples use the to_date function to convert the input date to a date type.

Function	Results
`week_in_month(to_date('Jan 22, 1997', 'mon dd, yyyy'))`	4
`week_in_month(to_date('Jan 21, 1997', 'mon dd, yyyy'))`	3

6.3.5.111 week_in_year

Use the week_in_year function to return the week in the year in which the given date falls.

Syntax

```
week_in_year(<inputdate>,'<weektype>')
```

Return value

int

Returns from 1 to 53.

Where

`<inputdate>`	The source date
`<weektype>`	Optional. This function returns the week in the year in two ways based on your setting: • 'WW' - Absolute week number of the given date. • 'IW' - ISO week number of the given date.

Details

Consider the following information when you use this function:

- This function considers the first week of the year to be the first seven days when it determines the absolute week number.
- Under the ISO standard, a week always begins on a Monday, and ends on a Sunday.
- The first week of a year is that week which contains the first Thursday of the year.
- An ISO week number may be between 1 and 53.
- Under the ISO standard, week 1 always has at least 4 days.
- If 1-Jan falls on a Friday, Saturday, or Sunday, the first few days of the year are defined as being in the last (52nd or 53rd) week of the previous year.

Example

Some business applications use week numbers to categorize dates. For example, a business may report sales amounts by week, and identify each period as "9912", representing the 12th week of 1999. An ISO week is more meaningful than an absolute week for such a purpose.)

Following are more example results for week_in_year applied to three different input dates:

Function	Results
`week_in_year(to_date('Jan 01, 2001','mon dd, yyyy'))`	`1`
`week_in_year(to_date('2005.01.01', 'yyyy.mm.dd'),'WW')`	`1`
`week_in_year(to_date('2005.01.01', 'yyyy.mm.dd'),'IW')`	`53`

6.3.5.112 word

Use the word function to return one word out of a given string.

Syntax

```
word(<input_string>, <word_num>)
```

Return value

varchar

A string containing the indicated word. The return type is the same as <input_string>.

Where

<input_string>	The source string.
<word_num>	A nonnegative integer specifying the index of the target word in the string. The first word in a string is word number 1. If <word_num> is 0 or greater than the number of words in <input_string>, then the word function returns a NULL string.

Details

A word is defined to be any string of consecutive non-white space characters terminated by white space, or the beginning and end of <input_string>. White space characters are the following:

- Space
- Horizontal or vertical tab
- Newline
- Linefeed

Example

Function	Results
word('Accounting Department', 1)	'Accounting'
word('Accounting', 1)	'Accounting'

Function	Results
`word('Accounting', 2)`	`NULL`

6.3.5.113 word_ext

Use the word_ext function to return a word that you identify by a position in a delimited string.

Syntax

```
word_ext('<string>', <word_num>, '<separator(s)>')
```

Return value

varchar

A string containing the indicated word. Return type is the same as `<string>`.

Where

`<string>`	The source string.
`<word_num>`	A nonnegative integer specifying the index of the target word in the string. The first word in a string is word number 1. If `<word_num>` is 0 or greater than the number of words in `<string>`, then the word function returns a NULL string.
`<separator(s)>`	Any character specified.

The function considers a word to be:

- Any string of consecutive characters, not including white spaces
- String of characters terminated by white space or the beginnning and end of `<string>`.

The function considers white space characters to be:

- Space
- Horizontal or vertical tab
- Newline
- Linefeed

Details

This function is useful for parsing Web log URLs or file names.

Example

Function	Results
`word_ext('www.sap.com',2,'.')`	`'sap'`
`word_ext('www.cs.wisc.edu', -2, '.')`	`'wisc'` A negative word number, -2 in the example, means the function counts from right to left.
`word_ext('www.cs.wisc.edu', 5, '.')`	`NULL`
`word_ext('aaa+=bbb+=ccc+zz=dd', 4, '+=')`	`'zz'` If 2 separators are specified (+=), the function looks for either one.
`word_ext(',,,,,aaa,,,,bb,,,c ', 2, ',')`	`'bb'` This function skips consecutive delimiters.

6.3.5.114 year

Use the year function to determine the year in which the given date falls.

Syntax

```
year(<date1>)
```

Return value

int

The number that represents the year component of `<date1>`.

Where

`<date1>`	The source date.

Details

Example

Function	Results
`year(to_date('Jan 22, 1997','mon dd, yyyy'))`	`1997`
`year(to_date('03/97', 'mm/yy'))`	`1997`
`year(to_date('03/19', 'mm/yy'))`	`2019`

7 Administration

The administration section provides information about additional settings and configurations within SAP Cloud Integration for data services.

Agents [page 384]
: An agent provides connectivity to on-premise sources in your system landscape.

Configure Email Notification [page 385]
: Email notifications can be sent based on the results of scheduled task and process runs or due to agent downtime.

User Profile [page 387]
: In the user profile tab, you can configure your preferred display language.

Activate or Deactivate Multiple Schedules [page 387]
: You can select and activate or deactivate multiple schedules at one time.

Create Custom Calendars [page 388]
: Custom calendars allow you to specify a customized schedule for running tasks or processes.

Create Specialized Monthly Schedules [page 389]
: You can create schedules that run on a monthly basis on the last day of the month, or the first or last workday of the month.

Related Information

7.1 Agents

An agent provides connectivity to on-premise sources in your system landscape.

At design-time, the agent is used to provide metadata browsing functionality for on-premise sources to the web-based user interface. At run-time, the agent manages the secure data transfer from your on-premise sources to your cloud-based target application.

Agent groups ensure high-availability by clustering one or more agents and making sure tasks and processes get assigned only to available agents in the group.

Parent topic: Administration [page 384]

Related Information

Create an Agent [page 385]
Configure Email Notification [page 385]
User Profile [page 387]
Activate or Deactivate Multiple Schedules [page 387]
Create Custom Calendars [page 388]
Create Specialized Monthly Schedules [page 389]
SAP Data Services Agent
SAP Data Services Agent Guide

7.1.1 Create an Agent

You create an agent to provide basic metadata before configuring it to then connect to on-premise sources in your system landscape.

1. On the *Agents* tab, select *New Agent*.
2. Provide a name for the agent.
3. (Optional) Enter the agent's description.
4. (Optional) Indicate whether you want to receive notifications by email if this agent goes down.
5. Add the agent to an existing group or create a new group to which to add it. This is required; doing so supports high-availability for your production tasks.
6. Click *Save*.

The list of agents displays the group names alphabetically and, within each group, the agents alphabetically.

> → Remember
>
> After you create an agent, the agent is not ready for you to use until you configure it. For more information, see the SAP Data Services Agent Guide, in particular the section Configuring the SAP Data Services Agent.

Related Information

About Agent Groups

7.2 Configure Email Notification

Email notifications can be sent based on the results of scheduled task and process runs or due to agent downtime.

You must have Administrator permissions to configure notifications.

Notifications can be sent to any valid email address.

Task overview: Administration [page 384]

Related Information

Agents [page 384]
User Profile [page 387]
Activate or Deactivate Multiple Schedules [page 387]
Create Custom Calendars [page 388]
Create Specialized Monthly Schedules [page 389]

Configure task or process notifications

Tasks and processes must already be set up to run on a scheduled basis.

> **i** Note
>
> Email notifications for tasks or processes can be set for the Production environments. Notifications are not available for Sandbox.

Email notifications about the status of tasks and processes are captured in the security log.

1. On the *Administration* tab, select *Notifications*.
2. Enter the email addresses that should receive notifications about tasks or processes. Use a semicolon to separate multiple email addresses.
3. Select the type of email notifications to be sent, then click *Save*.

Configure agent notifications

Agent downtime notifications are sent for all environments including sandbox, production, and additional environments such as development or test.

Downtime is a period of five minutes or longer. The server checks every 15 minutes.

In addition to creating an email notification list, in the Edit Agent dialog for each applicable agent you must select the *Receive Downtime Notifications* checkbox. To do this, on the Agents tab, click *Actions* > *Edit*.

1. On the *Administration* tab, select *Notifications*.
2. Enter the email addresses that should receive notifications about agent downtime. Use a semicolon to separate multiple email addresses.

7.3 User Profile

In the user profile tab, you can configure your preferred display language.

> ! Restriction
>
> The current version of Cloud Integration for data services supports only English.

Parent topic: Administration [page 384]

Related Information

Agents [page 384]
Configure Email Notification [page 385]
Activate or Deactivate Multiple Schedules [page 387]
Create Custom Calendars [page 388]
Create Specialized Monthly Schedules [page 389]

7.4 Activate or Deactivate Multiple Schedules

You can select and activate or deactivate multiple schedules at one time.

1. In the *Dashboard* tab, click *Schedules*
2. Do one of the following:
   - Select individual schedules as needed.
   - To select all schedules, click the box in the upper left corner of the table.
   - To select a group of schedules, press `Shift` on your keyboard .

     > → Tip
     >
     > You can click the *Active* tab and sort schedules that are active or inactive.
3. Click *Activate* or *Deactivate* and confirm the action.

Task overview: Administration [page 384]

Related Information

Agents [page 384]

Configure Email Notification [page 385]
User Profile [page 387]
Create Custom Calendars [page 388]
Create Specialized Monthly Schedules [page 389]

7.5 Create Custom Calendars

Custom calendars allow you to specify a customized schedule for running tasks or processes.

With the *Administrator* role, you can create a custom calendar that specifies the dates you want a task or process to run. Once saved, the custom calendar becomes available to all users in a schedule dialog when *Run Frequency* is set to *Custom*.

1. In the *Administration* tab, click *Calendars*.
2. Click the plus button ( ) to create a new custom calendar.
3. Enter a name and optionally a description for your calendar.
4. Add the dates you want a task or process to run by doing one of the following:

Option	Description
Manually enter the dates	Type the dates in the *Run On* field. The dates must be of the format **`YYYY.MM.DD`**. You must separate two dates by a comma or by entering the second date on a new line.
Select dates by using the calendar button	Click the calendar button ( ) and select dates. The dates are automatically added onto new lines.
Upload a *Calendar File*	*Browse* your local system and select a CSV file that defines your business calendar. Click *Open*, and the dates in the file will be automatically populated into the *Run On* field. **i Note** In the CSV file, the dates must also follow the **`YYYY.MM.DD`** format, and be separated with commas or on new lines.

5. Click *Save*.

Task overview: Administration [page 384]

Related Information

Agents [page 384]
Configure Email Notification [page 385]

User Profile [page 387]
Activate or Deactivate Multiple Schedules [page 387]
Create Specialized Monthly Schedules [page 389]

7.6 Create Specialized Monthly Schedules

You can create schedules that run on a monthly basis on the last day of the month, or the first or last workday of the month.

1. In the *Projects* tab, select a task or process. Click *Schedule* > *New*.
2. Set *Run Frequency* to *Monthly*.
3. Enter the *Start Time* to determine when the schedule comes into effect.
4. Choose a *Specific Day* to determine on which day the schedule runs during a month. Options are as follows:

Option	First run time
Leave the field blank as default	The task or process starts running at the time and on the day you select in *Start Time*.
Choose *Last Day of Month*	The task or process starts running at the time and on the last day of the month you select in *Start Time*.
Choose *First Workday of Month* **and specify whether your** *Workweek Starts On* **Monday or Sunday**	The task or process starts running at the time and on the first workday of the month you select in *Start Time*.
Choose *Last Workday of Month* **and specify whether your** *Workweek Starts On* **Monday or Sunday**	The task or process starts running at the time and on the last workday of the month you select in *Start Time*.

> **i** Note
>
> At the time you submit the schedule, if your local time has passed the time you set in *Start Time*, the first run will start the following month.

5. In the *Repeat Every N Months* field, type a positive integer to define the run period. Valid values are 1, 2, 3, 4, 6 and 12. The value is set to 1 by default.

   For example, if N is set to 2 and the first run starts at 9:00am on April 1, then the second and third run will start at 9:00am on June 1 and August 1 respectively.
6. Enter the *End Time* to determine when the schedule expires.

Task overview: Administration [page 384]

Related Information

Agents [page 384]
Configure Email Notification [page 385]

User Profile [page 387]
Activate or Deactivate Multiple Schedules [page 387]
Create Custom Calendars [page 388]

8 Security

The security sections provides information about settings and configurations relevant to operating your SAP Cloud Integration for data services account in a secure manner.

User Roles [page 392]
: Control access to SAP Cloud Integration for data services functionality by assigning roles to your standard (non-web services) users.

Enable Access for SAP Support [page 394]
: To efficiently troubleshoot an issue you are experiencing with SAP Cloud Integration for data services, you can allow SAP Support to temporarily access your system. Access for SAP Support users is secure and audited.

Disable SAP Support Access and Users [page 395]
: At any time you can disable SAP Support access or individual SAP Support users.

Security Log [page 396]
: The security log provides information about occurrences of user-related events, datastore updates, and task or process actions.

Set the Security Log Retention Period [page 397]
: The security log displays sensitive user data such as email addresses. Data in the security log is deleted after a predefined retention period.

Cryptographic Keys [page 398]
: Within SAP Cloud Integration for data services, certain data is encrypted to ensure privacy, keep it free from corruption, and maintain access control. Cryptographic keys are used to encrypt and decrypt this sensitive data.

Transfer Your Identity Provider (IdP) [page 399]
: Users are authenticated by the SAP Cloud Identity Service. If you have configured a corporate tenant within SAP Cloud Identity Service or have a third-party corporate identity provider and use SAP Cloud Identity Service as a proxy, you can transfer the identity provider for SAP Cloud Integration for data services.

Related Information

8.1 User Roles

Control access to SAP Cloud Integration for data services functionality by assigning roles to your standard (non-web services) users.

> **i Note**
>
> You must have Security Administrator permissions to create users and assign roles.

SAP Cloud Integration for data services supports the following user roles:

Role	Authorizations
Production Operator	• Executes and schedules tasks and processes in the Production environment • Views tasks, processes, data flows, and datastore connections • Monitors running and finished tasks and processes from the *Projects* and *Dashboard* tabs
Administrator	• Has all the abilities of a Production Operator • Manages the registration of Data Services Agent instances • Creates or modifies datastore connection information in the Production environment • Promotes tasks and processes between environments, for example from Sandbox to Production • Can reset the cache of tasks and processes in Production
Developer	• Creates and modifies tasks, processes, data flows, and datastore connections • Executes and schedules tasks and processes in non-production environments such as Sandbox and views data to verify the results • Can access only non-production environments, such as Sandbox
Security Administrator	• Creates, activates, and deletes users • Assigns roles to users • Views security log
SAP Support	• The SAP Support user role provides limited access to Sandbox and Production environments. Members of the SAP Support team are automatically assigned to this role to facilitate troubleshooting. For details, see SAP Support user role permissions [page 393]. • The Security Administrator cannot assign or unassign users to this role, but can add additional roles to the user.

Permissions of the SAP Support user role

Activity	Sandbox	Production
Log on	Yes	Yes
View projects, processes, tasks, data flows and their configurations	Yes	Yes
Edit projects, processes, tasks, data flows and their configurations	Yes	No
All deletion activities except sources, transforms, targets, and imported tables in Sandbox	No	No
Export tasks	Yes	Yes
Import and promote tasks	No	No
System Configuration: view	Yes	Yes
System Configuration: edit	Yes	No
Datastore: view	Yes	Yes
Datastore: edit, including import tables	Yes	No
Schedule: view configuration	Yes	Yes
Schedule: activate, deactivate, update, and delete	No	No
Task: "Run Now"	Yes	No
Design-time data (JIT): execution	Yes	No
History: view	Yes	Yes
History: clear	No	No
Manage task version	No	No
Task execution logs (run in debug mode)	Yes	Yes
Datastore test connection	Yes	Yes
Web Services: views	Yes	Yes
Web Services: run task	Yes	No

Activity	Sandbox	Production
Administration tab (all subtabs)	No	No
Can reset the cache of tasks and processes in Production	Yes	Yes

Parent topic: Security [page 391]

Related Information

Enable Access for SAP Support [page 394]
Disable SAP Support Access and Users [page 395]
Security Log [page 396]
Set the Security Log Retention Period [page 397]
Cryptographic Keys [page 398]
Transfer Your Identity Provider (IdP) [page 399]
Enable Access for SAP Support [page 394]

8.2 Enable Access for SAP Support

To efficiently troubleshoot an issue you are experiencing with SAP Cloud Integration for data services, you can allow SAP Support to temporarily access your system. Access for SAP Support users is secure and audited.

After the security administrator grants access, an SAP Support user can be created by SAP. In your users list, SAP Support users are indicated by a wrench icon ( ) and assigned the SAP Support role. The SAP Support user role provides limited access to Sandbox and Production environments and should be sufficient to diagnose most issues.

> **i Note**
>
> In the list of user roles (*Administration tab* > *Users*), the SAP Support user role is for information only and cannot be edited. Members of the SAP Support team who access your system are automatically assigned to this role. You cannot unassign the role, but you can add additional roles to the user.

To enable access:

1. In the *Administration* tab, click *Settings*.
2. Select *Allow SAP Support Access*.

Inform SAP Support that you have enabled access.

Task overview: Security [page 391]

Related Information

User Roles [page 392]
Disable SAP Support Access and Users [page 395]
Security Log [page 396]
Set the Security Log Retention Period [page 397]
Cryptographic Keys [page 398]
Transfer Your Identity Provider (IdP) [page 399]
User Roles [page 392]
Security Log [page 396]
Disable SAP Support Access and Users [page 395]
Disable SAP Support Access and Users [page 395]
Security Log [page 396]
Set the Security Log Retention Period [page 397]

8.3 Disable SAP Support Access and Users

At any time you can disable SAP Support access or individual SAP Support users.

- Do one of the following:

Result	Action
Disable SAP Support access. This action disables all SAP Support users.	1. In the *Administration* tab, click *Settings*. 2. Deselect *Allow SAP Support access*.
Disable or delete a specific SAP Support user	1. In the *Administration* tab, click *Users*. 2. Do one of the following: • To **disable** an SAP Support user, select the user and deselect *Active User*. • To **delete** an SAP Support user, select the user and click *Delete*.

Task overview: Security [page 391]

Related Information

User Roles [page 392]
Enable Access for SAP Support [page 394]
Security Log [page 396]
Set the Security Log Retention Period [page 397]
Cryptographic Keys [page 398]
Transfer Your Identity Provider (IdP) [page 399]

Enable Access for SAP Support [page 394]

8.4 Security Log

The security log provides information about occurrences of user-related events, datastore updates, and task or process actions.

In SAP Cloud Integration for data services, the security log can be accessed under *Administration* > *Security Log*. You must have Security Administrator permissions to view the security log.

The security log includes occurrences of the following events:

Security events

- Create, modify or delete a user
- User activation email sent
- Grant or revoke a user role
- View user details or roles
- Successful logins
- Create or delete an agent
- Cloud key store certificate added or deleted
- Enter or reset a datastore password
- View datastore configurations
- View datastore tables
- Access to SAP Cloud Integration for data services inbound web services (including IP address)
- View the security log

Datastore updates

- Create, update or delete datastores
- Add, reimport or delete tables
- Create, save, update, or delete system configurations

Task or process actions

- Create a task
- Edit a task or process name
- Edit task global variables
- Edit a task script
- Edit task data flows
- Edit a process
- Delete a data flow from a task
- Copy a task data flow
- Copy a task data flow to a new target
- Replicate a task or process
- Reset task or process cache

- Execute sandbox tasks
- Schedule a production task or process
- Activate, delete, deactivate, or modify a task or process schedule
- Run a task or process in Production
- Promote a task or process to Production
- Task status email notifications

> **i** Note
>
> Configuration data consists primarily of task definitions (mappings, filters, transformations, rules, connection information, and so on). Task or process definitions cannot be modified in the production environment.

Parent topic: Security [page 391]

Related Information

User Roles [page 392]
Enable Access for SAP Support [page 394]
Disable SAP Support Access and Users [page 395]
Set the Security Log Retention Period [page 397]
Cryptographic Keys [page 398]
Transfer Your Identity Provider (IdP) [page 399]

8.5 Set the Security Log Retention Period

The security log displays sensitive user data such as email addresses. Data in the security log is deleted after a predefined retention period.

You must have the Security Administrator role in order to change the log retention period.

At the end of the specifed log retention period, the expired data in the security log is automatically deleted. The default retention period is 60 months (five years).

1. Navigate to *Administration* > *Settings*.
2. Enter the number of months to retain the data in the security log.

   The number must be a positive integer between 1 and 60.

Task overview: Security [page 391]

Related Information

User Roles [page 392]
Enable Access for SAP Support [page 394]
Disable SAP Support Access and Users [page 395]
Security Log [page 396]
Cryptographic Keys [page 398]
Transfer Your Identity Provider (IdP) [page 399]

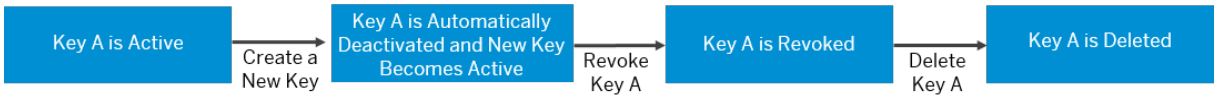
8.6 Cryptographic Keys

Within SAP Cloud Integration for data services, certain data is encrypted to ensure privacy, keep it free from corruption, and maintain access control. Cryptographic keys are used to encrypt and decrypt this sensitive data.

A cryptographic key is generated for each environment in an organization. In the default organization that contains Sandbox and Production environments, two keys are generated. Cryptographic keys need to be replaced regularly to minimize the risk of being compromised. The security officer manages the keys based on the organization's security guidelines and procedures.

Key lifecycle management

At any given time, only one cryptographic key can be designated as active.
Each cryptographic key moves through a lifecycle illustrated in the following diagram:



The available statuses of a cryptographic key are explained in the following table:

Status	Definition
Active	The active key is used to encrypt current sensitive data. The key is also used to decrypt all sensitive data. When a new cryptographic key is created, the current active key moves to a deactivated state. An active key cannot be deleted from the system.
Deactivated	A deactivated key can no longer be used to encrypt data. It can however be used to decrypt all data encrypted when the key was active. You cannot reactivate a key once it has been deactivated. A deactivated key cannot be deleted directly from the system. Its status must first be changed to revoked before it can be deleted.
Revoked	When a cryptographic key is revoked, a process is launched in which all data encrypted with the key is decrypted and then re-encrypted with the current active cryptographic key. This process may take some time. Once a key is revoked it can safely be deleted from the system. The revocation mechanism ensures that encrypted data can always be decrypted. There is no way to reactivate a key once it has been revoked.

Status	Definition
Deleted	The deleted key is no longer displayed and can be safely removed from the database or file system.

> **i** Note
>
> You must have Security Administrator permissions to manage cryptographic keys.

Parent topic: Security [page 391]

Related Information

User Roles [page 392]
Enable Access for SAP Support [page 394]
Disable SAP Support Access and Users [page 395]
Security Log [page 396]
Set the Security Log Retention Period [page 397]
Transfer Your Identity Provider (IdP) [page 399]

8.7 Transfer Your Identity Provider (IdP)

Users are authenticated by the SAP Cloud Identity Service. If you have configured a corporate tenant within SAP Cloud Identity Service or have a third-party corporate identity provider and use SAP Cloud Identity Service as a proxy, you can transfer the identity provider for SAP Cloud Integration for data services.

Before you transfer your identity provider, consider the following items:

- The Security Administrator must perform the transfer.
- When creating assertion attributes, you will needed to create or modify an "E-mail" user attribute with a corresponding assertion attribute of **mail** or **email**. The assertion attribute is not case-sensitive, however it must be either **mail** or **email**. Your company may have an approval process required for new assertion attributes. Please ensure that the attribute is approved, if needed.
- SAP recommends that you have a planned downtime period for your users while you transfer your identity provider. Tasks and processes that are running or scheduled to run are not affected by the change in authentication service provider.
- Authentication of individual users for web services is not supported by SAP Cloud Identity. It is recommended that you use the *WebServicesUser* available in SAP Cloud Integration for data services instead.
  The WebServicesUser is always authenticated by the default authentication for SAP Cloud Integration for data services.
- When you use SAP Identity Authentication (IAS) on its own or you use IAS along with a third-party identity provider (IdP), SAP Cloud Integration for data services does not send newly created users to IAS nor to

your IdP. You must create new users there separately in addition to creating them in SAP Cloud Integration for data services. Also, when your system is configured this way, the *Send Activation Email* button does **not** send the courtesy email to new users. Note that this email is not required for new users to access the application. Separately, you should provide the system URL in the form of `https://<Server URL>/DSoD/session/logon/<ORG>` to these new users, who can access the system as soon as you have added them to SAP Cloud Integration for data services, to IAS, and to your IdP, depending on your setup.

To transfer to a new identity provider, follow the tasks below.

1. Download the Service Provider (SP) Metadata File [page 401]
   Dowload the Service Provider (SP) metadata file from SAP Cloud Integration for data services to use when configuring SAML 2.0 trust for the new identity provider (IdP).
2. Create a New Application for SAP Cloud Integration for data services [page 401]
   In the SAP Cloud Identity Authentication Administration Console, create an application for your SAP Cloud Integration for data services.
3. Configure the SAML 2.0 Trust With the Service Provider [page 401]
   Use the service provider (SP) metadata file to configure SAML 2.0 trust.
4. Define Assertion Attributes [page 402]
   The method you follow to define assertion attributes depends on the type of identity provider your company uses.
5. Update the Identity Provider (IdP) Metadata in SAP Cloud Integration for data services [page 404]
   Download the Identity Provider (IdP) metadata file from the SAP Cloud Platform Identity Authentication Administraton console and then update the IdP setting SAP Cloud Integration for data services.

Parent topic: Security [page 391]

Related Information

User Roles [page 392]
Enable Access for SAP Support [page 394]
Disable SAP Support Access and Users [page 395]
Security Log [page 396]
Set the Security Log Retention Period [page 397]
Cryptographic Keys [page 398]
Activate the Pre-defined User WebServicesUser
Finish Setting Up a Standard User

8.7.1 Download the Service Provider (SP) Metadata File

Dowload the Service Provider (SP) metadata file from SAP Cloud Integration for data services to use when configuring SAML 2.0 trust for the new identity provider (IdP).

> **i** Note
>
> You must have the Security Administrator role to complete this action.

1. Log into SAP Cloud Integration for data services.
2. Click the *Administration* tab and then *Identity Provider*.
3. Click *Get SP Metadata*.
4. Download and save the .zip file.

Task overview: Transfer Your Identity Provider (IdP) [page 399]

Next task: Create a New Application for SAP Cloud Integration for data services [page 401]

8.7.2 Create a New Application for SAP Cloud Integration for data services

In the SAP Cloud Identity Authentication Administration Console, create an application for your SAP Cloud Integration for data services.

1. Log into SAP Cloud Platform Identity Authenticaton Administration Console.
2. Select the *Applications* tile.
3. Select the *+Add* button at the bottom of the left-hand panel in order to add a new application to the list. Follow your naming convention and enter a name for the application.

   The name of the application is displayed on the login and registration pages.

Task overview: Transfer Your Identity Provider (IdP) [page 399]

Previous task: Download the Service Provider (SP) Metadata File [page 401]

Next task: Configure the SAML 2.0 Trust With the Service Provider [page 401]

8.7.3 Configure the SAML 2.0 Trust With the Service Provider

Use the service provider (SP) metadata file to configure SAML 2.0 trust.

- You have created an aplication for SAP Cloud Integration for data services in the SAP Cloud Identity Administration Console.

- You have downloaded and saved the service provider (SP) metadata XML file from the *Administration* tab in SAP Cloud Integration for data services

1. If needed, log into SAP Cloud Identity Administration Console and select the *Applications* tile.
2. Select the SAP Cloud Integration for data services application from the left-hand panel.
3. In the Application panel, choose the *Trust* tab.
4. Click *SAML 2.0 Configuration*.
5. In *Define from Metadata*, browse to the location of the service provider (SP) metadata XML file you downloaded previously.
6. Select *Save* in the lower right corner.

Task overview: Transfer Your Identity Provider (IdP) [page 399]

Previous task: Create a New Application for SAP Cloud Integration for data services [page 401]

Next: Define Assertion Attributes [page 402]

8.7.4 Define Assertion Attributes

The method you follow to define assertion attributes depends on the type of identity provider your company uses.

Choose the appropriate method to define assertion attributes:

Define Assertion Attributes When Using SAP Cloud Identity Services as Your Identity Provider [page 403]
: If you have a tenant within SAP Cloud Identity Services and use it as your main identity provider (IdP), define the assertion attributes directly in the SAP Cloud Identity Services Administration Console.

Change the Identity Provider and Define Assertion Attributes When Using a Corporate Identity Provider [page 403]
: If you use a corporate identity provider and have configured SAP Cloud Platform Identity Authentication service as a proxy, change to your corporate identity provider and then define the assertion attributes.

Parent topic: Transfer Your Identity Provider (IdP) [page 399]

Previous task: Configure the SAML 2.0 Trust With the Service Provider [page 401]

Next task: Update the Identity Provider (IdP) Metadata in SAP Cloud Integration for data services [page 404]

Related Information

8.7.4.1 Define Assertion Attributes When Using SAP Cloud Identity Services as Your Identity Provider

If you have a tenant within SAP Cloud Identity Services and use it as your main identity provider (IdP), define the assertion attributes directly in the SAP Cloud Identity Services Administration Console.

You have created an application for SAP Cloud Integration for data services.

1. If needed, log into SAP Cloud Identity Services Administration Console and navigate to your SAP Cloud Integration for data services application:
   a. Select the *Applications* tile.
   b. Select your SAP Cloud Integration for data services application from the left-hand panel.
   c. In the Application panel, choose the *Trust* tab.
2. Click *Assertion Attributes*.
3. As needed, modify the names of the assertion attributes. Ensure that the following three attributes are available:

User Attribute	Assertion Attribute
First Name	`first_name`
Last Name	`last_name`

E-Mail	`mail`
	or
	`email`
	The attribute is not case-sensitive; Email, EMail, eMail are all accepted.

4. As required for your environment, add additional assertion attributes.

8.7.4.2 Change the Identity Provider and Define Assertion Attributes When Using a Corporate Identity Provider

If you use a corporate identity provider and have configured SAP Cloud Platform Identity Authentication service as a proxy, change to your corporate identity provider and then define the assertion attributes.

- You have created an application for SAP Cloud Integration for data services
- A corporate identity provider has already been configured in SAP Cloud Platform Identity Authentication Service.

This task should not be performed if you have a tenant within SAP Cloud Platform Identity Authentication service and use it as your main identity provider (IdP)

Change to Your Corporate Identity Provider

1. If needed, log into SAP Cloud Platform Identity Authentication Administration Console and navigate to your SAP Cloud Integration for data services application:
   a. Select the *Applications* tile.
   b. Select your SAP Cloud Integration for data services application from the left-hand panel.
   c. In the Application panel, choose the *Trust* tab.
2. Click *Identity Provider*.
3. Select the desired identity provider.

Define Assertion Attributes

- Ensure that SAML configuration of the third-party corporate identity provider includes the following assertion attributes:

User Attribute	Assertion Attribute
First Name	`first_name`
Last Name	`last_name`

E-Mail	`mail`
	or
	`email`
	The attribute is not case-sensitive; Email, EMail, eMail are all accepted.

8.7.5 Update the Identity Provider (IdP) Metadata in SAP Cloud Integration for data services

Download the Identity Provider (IdP) metadata file from the SAP Cloud Platform Identity Authentication Administraton console and then update the IdP setting SAP Cloud Integration for data services.

> → Tip
>
> Test the new connection before you log out of your current session.

1. Download the Identity Provider (IdP) metadata file.
   a. Log into SAP Cloud Platform Identity Authentication Administration Console.
   b. Under Applications & Resources, click *Tenant Settings*.
   c. Click *SAML 2.0 Configuration*.

d. At the bottom of the SAML 2.0 Configuration information, click Download Metadata File to download and save the IdP metadata file.

2. Update the IdP settings in SAP Cloud Integration for data services.

> **i Note**
>
> You must have the Security Administrator role.

   a. Log into SAP Cloud Integration for data services.
   b. Click the *Administration* tab and then *Identity Provider*.
   c. Click *Update IdP Settings*.
   d. In the Update IdP Settings dialog, update the settings in one of the following ways:
      - Browse to the IdP metadata file you previously downloaded and saved.
      - Manually enter the required information for the *IDP Name*, *SSO Endpoint*, *SLO Endpoint*, and *Signing Certificate*.
   e. Click *Save*.

3. **Test your new connection**.
   a. Before logging out of your current SAP Cloud Integration for data services session, open a new browser window in Chrome Incognito or Internet Explorer InPrivate Browsing.
   b. Log into SAP Cloud Integration for data services in the new browser window and confirm that you are redirected to your corporate identity provider.

> **→ Tip**
>
> If necessary, in the *Identity Provider* tab, use *Revert to Default IdP* to reset to the original identity provider.

The transfer of your identity provider is now complete.

Task overview: Transfer Your Identity Provider (IdP) [page 399]

Previous: Define Assertion Attributes [page 402]

9 Monitoring and Troubleshooting

The monitoring and troubleshooting sections provides information on the tasks and details related to the lifecycle of SAP Cloud Integration for data services.

Monitoring Production Status [page 406]
In the Dashboards, the production status displays whether your production tasks and processes succeeded or failed over a given period of time.

Troubleshooting Task or Process Failures [page 409]
Many errors are caused by simple configuration or connectivity errors on a data source, the agent host system, or the target cloud application. View the error log for details about a particular failure, and if necessary, contact another user to resolve the issue.

Troubleshooting During Task or Process Development [page 412]
Errors that occur during task or process execution can be caused by configuration errors or issues within the task, process, and data flow logic.

Related Information

9.1 Monitoring Production Status

In the Dashboards, the production status displays whether your production tasks and processes succeeded or failed over a given period of time.

From the production status, you can:

- Set the time period for which you want to analyze results.
- Click on an area of the pie chart to filter tasks and processes displayed in the table.
- Click on a task or process in the table to view its history and log data.

> **i** Note
>
> Hovering over the status column in the table displays the number of successful and failed runs in the specified time period.

Success and failure statuses

The icons for tasks or processes that include SAP Integrated Business Planning post-processing contain a '!' symbol. Statuses are reported as described in the following table:

State of *Treat 'Processes with Error' as success* checkbox	Data load status for task or process execution	Post-processing in SAP Integrated Business Planning completes as:	Status result	Web services status
Selected	Succeeded	Succeeded	Succeeded	`TASK:SUCCESS`
Selected	Succeeded	Failed	Succeeded	`TASK:SUCCESS_WITH_ERRORS_D`
Deselected	Succeeded	Failed	Failed	`TASK:SUCCESS_WITH_ERRORS_E`
Deselected	Failed	Not applicable	Failed	`TASK:ERROR`

Last run succeeded status

Last run succeeded is a status available only on the Dashboard (in the pie chart and table view) and is indicated by a yellow diamond-shaped icon ( ). The status is reported when a task or process has a successful run following a failed run. The purpose of the status is to make it easy to track the run results after changes are made to address issues that caused the failed run.

> **i** Note
>
> The *Last Run Succeeded* state is independent of how SAP Integrated Business Planning post-processing is treated or completes.

Parent topic: Monitoring and Troubleshooting [page 406]

Related Information

Log Files [page 408]
Troubleshooting Task or Process Failures [page 409]
Troubleshooting During Task or Process Development [page 412]
Troubleshooting Task or Process Failures [page 409]
Project Operations
Post-Processing for SAP Integrated Business Planning [page 160]

9.1.1 Log Files

Trace, monitor, and error logs show information about tasks that have been run.

To view these logs, go to the *Projects* tab, select a project, select a task, and select *View History*.

Trace Log

The trace log shows the execution progress of the job.

For unsuccessful jobs, use the trace log to see which components of a partially executed job completed or where an error occurred.

If the trace log ends after several *JOB* lines, the job did not execute successfully.

Trace logs show G_IBP_ global variables used in jobs. G_IBP_ global variables are supported only for WebRFC connections.

Monitor Log

The monitor log quantifies the activities of the components of the job. It lists the time spent in a given component of a job and the number of data rows which streamed through the component.

Use the monitor log to help tune the performance of a job.

Entry	Description
Path Name	+ means it's a datasource - means it's a target
State	Indicates the current status of the execution of the object. If you view the log while the job is running, this value changes as the status changes. The possible values are START, PROCEED, and STOP. In a successfully run job, all of these values are STOP to indicate that they finished successfully.
Row Count	Indicates the number of rows processed through this object.
Elapsed Time	Indicates the time (in seconds) since this object received its first row of data.
Absolute Time	Indicates the time (in seconds) since the execution of this entire data flow began.

Error Log

The error log lists errors generated during processing. If the error log is empty, the job completed successfully.

9.2 Troubleshooting Task or Process Failures

Many errors are caused by simple configuration or connectivity errors on a data source, the agent host system, or the target cloud application. View the error log for details about a particular failure, and if necessary, contact another user to resolve the issue.

Single task or process failure

When the dashboard indicates that a single task or process has failed, consider the following troubleshooting steps:

- Check the error log for details about the failure.
- If needed, send the detailed error message to the developer to correct the task or process.
- Prevent additional failures by stopping any schedules that include the task or process until the error is corrected.

> **i** Note
>
> *Last run succeeded* means that the most recent execution attempt succeeded, but that a previous attempt within the current time period failed.
>
> When a previous execution attempt has failed, you may wish to verify any delta loads and reload if necessary. Depending on the design of the task or process, a range of data may have been missed due to the failed execution attempt.
>
> You may need an administrator to view the data in the production datastore, and a developer or user may be required to validate the data.

Multiple task or process failure

If the dashboard indicates that many tasks or processes have failed, a configuration or connectivity problem with the SAP Data Services Agent or a data source is often the cause.

In addition to the suggested steps for single task or process failures, consider the following troubleshooting steps:

- Check the *Agent* tab to verify whether the agent is running and configured properly.
- Check whether other tasks or processes executed on the same agent also fail.
- If the tasks or processes share a common source, check for issues with the source and contact the database or basis administrator.

Common errors

Type of error	Who to contact
Invalid directory on the agent	Administrator responsible for managing the agent
Unable to connect to the source datastore	Database or basis administrator
A stored procedure failed to execute	Cloud application administrator

> **i** Note
>
> When you use SAP Business Suite applications as data sources, there are several other common reasons that a task or process may fail to execute:
>
> - The ABAP program was not transported to the production SAP system
> - SAP Data Services Agent failed to submit the job because the production SAP system was unreachable
> - The correct user authorizations are not configured on the production SAP system
> - The required functions are not installed on the production SAP system

For each of these error causes, you should contact your SAP basis administrator.

Parent topic: Monitoring and Troubleshooting [page 406]

Related Information

Reset Cache [page 410]
Change the Task Datastore during an SAP Integrated Business Planning for Supply Chain Migration [page 411]
Monitoring Production Status [page 406]
Troubleshooting During Task or Process Development [page 412]

9.2.1 Reset Cache

You can reset the cache of tasks and processes to ensure that the cached ATL matches the current configuration. For example, you might need to reset your cache if you make changes to a task because of a change in your environment, but the task is already cached with its prior configuration. You might also need to reset cache if troubleshooting finds there is a cache consistency issue.

To reset the cache in Production, you must be an Administrator or a member of the SAP Support team. However, anyone who has access to the system can reset cache in Sandbox.

You must select a job in the list for the *Reset Cache* menu option to appear in the *More Actions* dropdown.

To reset the cache, perform these steps:

1. On the *Projects* tab, select a task or process.
2. Click *More Actions* > *Reset Cache*.

A confirmation message appears to notify you that the cache will regenerate on the next run.

3. Click *Yes* to continue.

   The system processes the cache reset request. You receive a final message when the reset has processed successfully or a message with an error ID that you can provide to SAP Support if there is a problem.

The next time the task or process runs, the system regenerates the cache.

9.2.2 Change the Task Datastore during an SAP Integrated Business Planning for Supply Chain Migration

If you are using SAP Integrated Business Planning for Supply Chain, you are migrating from a JDBC connection type to a WebSocket RFC connection type, and you have an issue with a task during or after the migration, you can fall back to using the JDBC connection for that task so the task runs successfully and does not impact development or production runs.

Use this procedure to revert the specific problematic WebSocket RFC task back to JDBC without having to revert all tasks back to JDBC. Once the connection issue with WebSocket RFC is fixed, use this procedure again to change the datastore for the task to WebSocket RFC and then run the previously failed jobs.

> **i Note**
>
> This functionality applies only to your default configuration.
>
> If a data flow from a switched task is used by a process, all data flow tasks that the process consumes need to be switched.

Prerequisites:

- You are migrating from a JDBC connection type to a WebSocket RFC connection type for SAP Integrated Business Planning for Supply Chain.
- JDBC and WebSocket RFC connection types have been configured on your tenant. The *Change Datastore* button mentioned in the steps below appears only for customers that have both connection types configured.
- The WebSocket RFC datastore must contain at least the same tables as the JDBC datastore, meaning it can have additional tables, but at a minimum must have the tables that are in the JDBC datastore.

When your migration is completed successfully, the option to change the datastore for tasks will become unavailable.

If you have only a JDBC connection type or only a WebSocket RFC connection type to IBP or are not migrating as described above, you will not see the *Change Datastore* button in the user interface.

1. In your Sandbox environment, locate the task and go into Edit mode.
2. Switch to the *Connections* tab.
3. Choose *Source* or *Target*.
4. Click the *Change Datastore* button.
5. Choose the datastore to which you want to change, then click *OK*.
   If the datastore you chose does not contain at least the same tables as the JDBC datastore, a message appears asking you to add all of the original tables to the selected WebSocket RFC datastore and to repeat this procedure.

9.3 Troubleshooting During Task or Process Development

Errors that occur during task or process execution can be caused by configuration errors or issues within the task, process, and data flow logic.

From the *Projects* tab, select a task or process and select *View History*. History is stored for 90 days. Errors and possible resolutions are shown in the following table:

Error	Possible resolution
"<tablename> is an invalid ABAP program name. Program names must be less than 40 characters and start with 'Z' or 'Y'".	In the SAP application datastore, check if the ABAP execution option is set to *Execute preloaded*. If it is, make sure that the ABAP program has been installed on the SAP application server. For more information, see the *Agent Guide*
java.security.InvalidKeyException: Illegal key size	This error may occur when enabling PGP encryption. See SAP Note 1887289.
java.lang.SecurityException: Unsupported keysize or algorithm parameters	This error may occur when enabling PGP encryption. See SAP Note 1887289.
Other errors	See SAP Note 1858095.

Parent topic: Monitoring and Troubleshooting [page 406]

Related Information

Monitoring Production Status [page 406]
Troubleshooting Task or Process Failures [page 409]
Log Files [page 408]

10 Supplemental Information

View the topics in this supplement for additional useful information about SAP Cloud Integration for data services.

Related Information

10.1 Using SAP BW/4HANA as a Source and as a Target

You can use SAP BW/4HANA as a source and as a target. There are special setup considerations you must follow for each.

Related Information

10.1.1 SAP BW/4HANA as a Source

You can utilize SAP BW/4HANA as a source by using an SAP Business Suite Applications datastore. As indicated in the steps in this topic, you must set the ODP context to BW when you set up the SAP Business Suite Applications datastore.

All functionality of an SAP Business Suite Applications datastore is supported. The following import functionality is supported:

- Import by browsing for tables and ODP extractors
- Import by name of tables, functions, and ODP extractors, which requires an object's full name. Importing using a partial name is not supported.

> **i** Note
>
> Connecting to BW/4HANA using an SAP BW Source datastore is not supported. For more information, see SAP Note 3090468.

SAP BW/4HANA version 2 SP04 and above are supported.

To use BW/4HANA as a source, perform the following steps:

1. Begin creating a new datastore.
2. Set the *Datastore Type* to *SAP Business Suite Applications*.
3. Set *ODP Context* to *BW*.
4. Complete the rest of the datastore's options as indicated in SAP Business Suite Applications [page 87].
5. Use the datastore to import metadata objects for use in SAP Cloud Integration for data services dataflows and ABAP dataflows. Applicable metadata objects include the following:
   - SAP operational data provider (ODP) sources: used to read source data
   - Tables
   - Functions

Related Information

Data type mapping for BW/4HANA [page 414]
Import SAP BW/4HANA source metadata [page 416]
SAP Business Suite Applications [page 87]

10.1.1.1 Data type mapping for BW/4HANA

When you import data from your BW/4HANA data source, SAP Cloud Integration for data services converts data types to native data types.

After processing, SAP Cloud Integration for data services converts data types back to BW/4HANA data types when it loads data to the BW/4HANA targets.

The following table contains the data type conversion from BW targets or ODP objects to ABAP data types and to SAP Cloud Integration for data services data types.

Table 31: Conversion of BW to SAP Cloud Integration for data services

BW target or ODP object	ABAP	Data Services
CHAR	c	varchar
NUMC	n	numeric or varchar Dependent on NUMC_AS_VARCHAR flag in DSConfig.txt file, default=numeric. For BW/4HANA, if length is greater than 96, always varchar.
STRING	string/g	varchar
SSTRING	g	varchar
DATS	d	date
TIMS	t	time
INT1	b	int
INT2	s	int
INT4	i	int
INT8	8	int
DEC	p	decimal
DF16_RAW	a	decimal
DF16_DEC	a	decimal
DF34_RAW	e	decimal
DF34_DEC	e	decimal
FLTP	f	double
RAW	x	varchar

The following table contains data conversions when the input data is from SAP R/3, ECC, and BW sources.

Table 32: R/3, ECC, and BW sources to ABAP and SAP Cloud Integration for data services data types

SAP R/3, ECC, and BW sources	ABAP	Data Services table import unless specified
CHAR	c	varchar
NUMC	n	numeric or varchar Depends on NUMC_AS_VARCHAR flag in DSConfig.txt file: default=numeric.
LCHR	c	varchar

SAP R/3, ECC, and BW sources	ABAP	Data Services table import unless specified
STRING	string/g	long or varchar Depends on IMPORT_SAP_STRING_AS_CHAR in DSConfig.txt file: default=long For function import, varchar
SSTRING	g	varchar For function import, long
VARC	v	varchar
PREC	s	varchar
DATS	d	date
TIMS	t	time
INT1	b	int
INT2	s	int
INT4	i	int
INT8	8	int
DEC	p	decimal
FLTP	f	double
RAW	x	varchar

10.1.1.2 Import SAP BW/4HANA source metadata

After you create the SAP Business Suite Applications source datastore, import SAP BW/4HANA source metadata by browsing for them or by selecting them by name.

1. Select the datastore to which you want to import objects.
2. On the *Tables* tab, click on one of the following icons:
   - Import Objects: Browse for and select the objects you want to import, then click *Import*.
   - Import Objects by Name: Select the type of object and enter an object's name, then click *OK*.

10.1.2 SAP BW/4HANA as a Target

You can use BW/4HANA as a target by creating an SAP BW Target datastore.

When you set up the SAP BW Target datastore for BW/4HANA, be sure to do the following:

- On the *Import Object By Name* dialog box, use a system name of **BW4** and select *Advanced DSO*.

- On the *Import Objects* dialog box, choose the `BW4 ADSOs` folder.
- Be sure to enable Secure Network Communications as described in the topic Enable Secure Network Communications (SNC) in BW [page 136]. For additional important information, refer to the topic Configuring the Use of the SAP Cryptographic Library for SNC . Ensure that you select the version of the document that matches your SAP NetWeaver installed version.

Related Information

Import SAP BW/4HANA target metadata [page 417]

SAP BW Target [page 103]

10.1.2.1 Import SAP BW/4HANA target metadata

Import SAP BW/4HANA target metadata by name, by browsing, and by searching.

After you create the SAP BW target datastore, follow the same procedure to import objects as you do for SAP Business Warehouse target objects. In addition, use the Search feature to find BW/4HANA target objects for import.

> **i** Note
>
> To access ADSOs with the BW target datastore, you must be using SAP BW/4HANA 2.0 or later versions.

SAP Cloud Integration for data services stores imported ADSOs and InfoObjects under the *BW/4HANA DataStore Objects* node in the *Datastores* tab of the object library. ADSOs load generated data from a data flow into HANA.

Related Information

Importing BW/4HANA metadata by searching [page 417]

10.1.2.1.1 Importing BW/4HANA metadata by searching

When you don't know the full name of an SAP BW/4HANA Advanced DataStore Object (ADSO), but you know that the name contains a word or string, use search criteria to find the ADSO to import.

Log in to SAP Data Services Designer.

1. In the object library, open the *Datastores* tab.
2. Right-click the applicable datastore and select *Search* from the dropdown menu.

   The *Search* dialog box opens.

3. Verify that the correct datastore name appears in *Look in*.
4. Select *External* for the next text box.
5. Select the applicable repository.
6. Select *BW/4HANA DataStore Objects* from the *Object Type* dropdown list.
7. Complete the *Name*, *Description*, and *Search all* search criteria options as applicable.
8. Click *Search*.

   A list of ADSOs that match your search criteria appears in the lower pane of the *Search* dialog box.
9. Right-click the name of the applicable ADSO and select *Import* from the dropdown list.

Use the imported ADSOs as targets in a regular batch data flow.

10.2 IBM DB2 iSeries Reader Support

IBM iSeries support in SAP Cloud Integration for data services is available through DB2 datastores.

When downloading from IBM , search for package name **db2 connect**. Be sure to install **DB2 Connect Server**. Note that the DB Connect Server for iSeries driver is different than the DB Connect driver. Contact your System Administrator if you need more information.

IBM iSeries support in SAP Cloud Integration for data services through DB2 datastores functions via a DSN connection type. For information about configuring a DSN connection, see DB2 [page 28].

The following table contains the data type conversion from iSeries targets to SAP Cloud Integration for data services data types:

DB2 Target data type	SAP Cloud Integration for data services data type
ADT_VARCHAR	varchar(5)
ADT_CHAR	varchar(50)
ADT_BLOB	blob
ADT_CLOB	long
ADT_DATE	date
ADT_DECIMAL	decimal(18,2)
ADT_DOUBLE	double
ADT_FLOAT21	real
ADT_FLOAT53	double
ADT_INTEGER	int
ADT_LONGVARCHAR	long
ADT_REAL	real
ADT_SMALLINT	int
ADT_TIME	time
ADT_TIMESTAMP	datetime

DB2 Target data type	SAP Cloud Integration for data services data type
ADT_UNIQ	int

10.3 Accessibility Features in SAP Cloud Integration for data services

To optimize your experience of SAP Cloud Integration for data services, the service provides features and settings that help you use the software efficiently.

SAP Cloud Integration for data services is based on SAPUI5. For this reason, some accessibility features for SAPUI5 are available. See the accessibility documentation for SAPUI5 on SAP Help Portal at SAPUI5 Accessibility for End Users.

SAP Cloud Integration for data services is part of SAP BTP. Therefore, accessibility features for SAP BTP also apply, which are described in Accessibility Features in SAP BTP Cockpit.

You may experience the following exceptions:

- Instances in which a screen reader may read icons as "Graphic" rather than by an identifying name.
- Instances in which there is no title or header on a pane.
- Instances in which a screen reader reads all the information from the top of the page before reading the label of a selected button.
- With a screen reader on, the *Actions* menu options when editing cannot be performed.
- There is no keyboard support provided for users to navigate the graphical layout in the *Edit data flow* screen.
- Labels are not associated with *Edit* fields in the *Details* menu.
- In forward navigation, the focus goes to the toolbar, but in backward navigation the focus goes to the *Action* label in the toolbar.
- There is no tooltip provided for a checked icon in the *Promoted* column in the table.
- When creating a data flow, drag and drop is supported only by mouse click; there is no keyboard support.
- There is no visible focus inside the *Input* and *Output* data view in the data flow editor.
- Navigation via keyboard is not possible for mappings presented as a table in the data flow wizard.
- With screen reader support, a user is not able to navigate the data flow wizard screen using a keyboard; the system becomes slow and there is no system reaction.
- The application uses scripting languages to display content, but the information provided by the script is not readable by assistive technology.

> **i Note**
>
> These are issues that persist throughout the application on screens similar to the ones listed.

11 Glossary

SAP Cloud Integration for data services terms and their definitions are listed below:

agent	An entity that provides connectivity between on-premise sources and targets in the cloud.
change data capture	The process of identifying only new or modified data and loading the changes to a target system.
data flow	An object which contains the steps to define the transformation of data from source to target.
data type	The format used to store a value, which can imply a default format for displaying and entering the value.
datastore	A logical channel connecting SAP Cloud Integration for data services to a source or target database or application.
datastore configuration	The definition of a connection to a particular database from a single datastore. A datastore may have more than one configuration.
extract, transform, and load (ETL)	The process of migrating data from a source to a target.
file location	A file location object is a special type of datastore, which contains connection information to remote file locations. The file location object is not used to connect to the location, but is used by other datastores instead to provide the appropriate connection information.
filter	The *Filter* tab under *Transform Details* in the data flow editor allows you to restrict the rows of data that will be considered in your query processing. Columns can be dropped in to the filter tab and values or conditions can be applied to those columns to limit the data that is considered.
global variable	Global variables are symbolic placeholders. When a task or process runs, these placeholders can be populated with values that can be used by the task or process data flow.
join	The *Join* tab under *Transform Details* in the data flow editor allows you to join two or more source tables in your query. The join is specified via join pairs and join conditions based on primary or foreign keys and column names, thus emulating typical SQL join statements via a graphical user interface.
mapping	The *Mapping* tab under *Transform Details* in the data flow editor allows you to map input to output columns in your query.
order by	The *Order By* tab under *Transform Details* in the data flow editor allows you to adjust the sort order of your query output data by dropping in columns that need to be sorted and applying ascending or descending sort orders.
organization	An organization is the high-level grouping of your data within the SAP Cloud Integration for data services cloud instance. An organization itself is subdivided into Sandbox

	and Production areas, where testing and finalized data and processes can be stored respectively.
process	A process is an executable object that allows you to control the order in which your data is loaded.
project	A project is a container that is used to group related tasks.
script	A step in a task or process that allows you to calculate values to pass to other parts of the task or process by calling functions, executing if-then-else statements, and assigning values to variables.
source	The data in a database or file that you want the application to process.
system configuration	A set of datastore configurations that you want to use together when running a task or process.
target	The database or application where the data is loaded.
task	A set of steps that are executed together. A task can be run on-demand or scheduled for execution.
template	A task containing predefined content which serves as the starting point for populating a data integration project.
transform	A step in a data flow that acts on a data set. The transform takes one or more data sets as input and produces an output data set.

12 FAQs

Here you can find solutions to frequently asked questions.

General questions

Q: Did the product name change?

A: Yes. SAP Cloud Integration for data services was formerly called SAP Cloud Platform Integration for data services.

Q: Is it possible to configure the setting for the timeout period?

A: No. Your session will automatically time out. This feature is to protect the security of your data.

Q: How do I contact SAP Support to report a problem?

A: Go to http://support.sap.com. If necessary, refer to component **`LOD-HCI-DS`**.

Q: What time zone is set for the times that display in the projects page, schedule, and so on?

A: UTC time zone (Coordinated Universal Time) is displayed in all locations except the *Schedule* dialog. In the *Schedule* dialog, task and process execution schedules are always set at the UTC offset. For example, Pacific Time is considered to be UTC - 8:00 hours year-round.

Q: On the Projects tab, why isn't the execution status updated?

A: Click the *Refresh* button in the upper-right corner of the page to see an updated status.

Q: While a task or process is running, why aren't the logs in the History updated?

A: The Trace and Monitor logs are refreshed every 10 seconds while the task or process is running. Click the *Refresh* button in the upper-right corner of the page to update the Error Log.

Q: Why can't I do certain tasks?

A: You may not have the necessary privileges. SAP Cloud Integration for data services has a role-based architecture. Your Security Administrator can tell you what roles you've been assigned. For more information, see User roles [page 392].

Q: I am using the SuccessFactors Adapter and the XSD is incompatible or out of date. How can I update the XSD used by **SAP Cloud Integration for data services***?*

A: Refer to SAP Note 1900616.

Q: Is it possible to use my own Identity Provider for user authentication and management?

A: Yes. Your Security Administrator can take care of that. See Transfer Your Identity Provider (IdP) [page 399].

Task, process, and data flow questions

Q: How do I view my data after I've run a task or process?

A: From the *Datastores* tab, select your target datastore and then the target object. Click the *View Data* icon ().

> **i Note**
>
> View Data is available only for SAP HANA application cloud datastores that are in non-production environments. If you do not see the View Data icon in your target datastores, contact SAP Support and request that they activate View Data functionality on your target application.

Q: Why can't I add a new transform after the Target Query transform?

A: The Target Query transform must be the final transform in the data flow. The columns in the Output pane reflect the schema for the target object.

Q: In a task that I created from a template, there are columns in the Output pane of the Target Query that are not mapped. Is this a problem?

A: The templates were created to cover a broad range of requirements. Columns that are not mapped in the Target Query may not be relevant. You may need to verify your specific requirements. Unmapped columns in the Output pane of the Target Query are OK and will not result in runtime errors.

Q: A task or process that I want to edit is locked by another user. How do I unlock it?

A: Only one user at a time may edit a task or process. If necessary, ask your administrator to unlock a task process that someone inadvertently left locked.

> **→ Tip**
>
> After the task or process has been unlocked, if needed, refresh the *Projects* tab.

Q: My task fails to run. The following message displays: "<tablename> is an invalid ABAP program name. Program names must be less than 40 characters and start with 'Z' or 'Y'". What should I do?

A: In the SAP application datastore, check if the ABAP execution option is set to *Execute preloaded*. If it is, make sure that the ABAP program has been installed on the SAP application server. For more information, see Configuring SAP Business Suite connectivity.

Q: My Integrated Business Planning for Sales and Operations task fails with the following error message: " # records failed with error, Special characters are not allowed". What should I do?

A: You can use an SAP Cloud Integration for data services function to remove the special characters. For more information, see SAP Note 2007254.

Q: I call an SAP web service in my data flow. I have mapped all input schemas correctly, but no data is returned from the web service call. What should I do?

A: SAP web services have some schemas that are optional for the web service request since they are intended for response structures. You must map at least one column in this optional schema for the web service to provide a result.

Q: When I run a task containing multiple data flows, in what order are the data flows executed?

A: The data flows will be executed sequentially following the order in which the targets are listed in the task. You can modify the data flow execution order by choosing *Manage target order* from the *Actions* menu in the task editor. Alternatively, you can create a process to define the execution order of data flows from the same or different tasks.

Q: Can File format datastore be selected both as source and target?

A: Yes, you can simultaneously select File format datastore both as source and target.

Q: Can I use a web service as a source?

A: You can call a web service function to retrieve source data by using the Web Service transform type within your data flow.

After you choose the web service transform type, click *Select Web Service Function* in the Output actions. Select the function from the available web service datastores, and the request and response schemas will be added to your data flow automatically.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP