PUBLIC

SAP HANA Smart Data Integration and SAP HANA Smart Data Quality 1.0 SP03
Document Version: 1.0 – 2017-11-03

Administration Guide

Content

1 Administration Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality

This guide describes the common tasks and concepts necessary for the ongoing operation, administration, and monitoring of SAP HANA smart data integration and SAP HANA smart data quality.

The following areas are covered:

- Monitoring
- Administration and maintenance tasks
- Troubleshooting and recovery operations

For information about the initial installation and configuration of SAP HANA smart data integration and SAP HANA smart data quality, refer to the *Installation and Configuration Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality*.

For information about administration of the overall SAP HANA system, refer to the *SAP HANA Administration Guide*.

2 Monitoring Data Provisioning

Monitor the Data Provisioning Server availability and key performance indicators regularly.

The following monitoring tools are available:

- The Data Provisioning Monitor provides a browser based interface to monitor agents, tasks, design time objects (flowgraphs and replication tasks), and remote subscriptions created in the SAP HANA system. After you take steps to set up this Data Provisioning Monitor, you can view the monitors in the following ways:
  - Enter the URL address of each monitor directly into a web browser.
  - Access Data Provisioning tiles in SAP HANA Cockpit.
- The SAP HANA Administration and Monitoring group of tiles in SAP HANA Cockpit provides a Database Status application in which you monitor the Data Provisioning Server. For more information, see the *SAP HANA Administration Guide*, section "Analyze Status and Resource Usage of Services".

2.1 SAP HANA Tile Catalog for Data Provisioning

The tile catalog defines the set of all tiles available in the SAP HANA cockpit. Within the main tile catalog, tiles are grouped into sub-catalogs according to functional area. The tile group *SAP HANA Data Provisioning* contains the tiles for monitoring agents, remote subscriptions, tasks, and design time objects, as well as a tile to reference the documentation on the SAP Help Portal. The tiles are dynamic and update automatically with information from each monitor.

2.2 Access the Data Provisioning Monitors

You can access the Data Provisioning Monitors through either the SAP HANA Cockpit or by typing the URLs directly into your web browsers.

Prerequisites

- The user must have the following roles or privileges to use the data provisioning monitors.

  Table 1: Roles and Privileges

Action	Role or Privilege
Access the monitors	Role: sap.hana.im.dp.monitor.roles::Monitoring
Perform administration tasks from the monitors	Role: sap.hana.im.dp.monitor.roles::Operations

- Your Web browser supports the SAPUI5 library `sap.m` (for example, Internet Explorer 9).
  For more information about SAPUI5 browser support, see SAP Note 1716423 and the Product Availability Matrix (PAM) for SAPUI5.

Procedure

- To open the Data Provisioning monitors through the SAP HANA Cockpit:
  a. Enter the SAP HANA cockpit URL in your browser.

     `http://<host>:<port>/sap/hana/admin/cockpit`
  b. If required, enter your database user name and password.
  c. Go to the group of tiles that contain the Data Provisioning monitors.
- To open the Data Provisioning monitors directly, enter the following URLs in your browser.

Option	Description
Data Provisioning Agent Monitor	`<host name>:80<2 digit instance number>/sap/hana/im/dp/monitor/index.html?view=DPAgentMonitor`
Data Provisioning Subscription Monitor	`<host name>:80<2 digit instance number>/sap/hana/im/dp/monitor/index.html?view=DPSubscriptionMonitor`
Data Provisioning Task Monitor	`<host name>:80<2 digit instance number>/sap/hana/im/dp/monitor/index.html?view=IMTaskMonitor`
Data Provisioning Design Time Object Monitor	`<host name>:80<2 digit instance number>/sap/hana/im/dp/monitor/index.html?view=IMDesignTimeObjectMonitor`

2.3 Configure User Settings Profiles

By creating user settings profiles, you can quickly switch between different monitor layouts.Settings profiles contain information about visible columns, column order, column width, column filters, table visibility, and slider positions.

Context

You can create, modify, or remove settings profiles in each Data Provisioning Monitor by clicking the *Settings* button.

Procedure

- To add a new settings profile, click *Add*.
  a. Specify a name for the profile and whether to make it the default profile.
  b. Click *Add*.

  A new profile is created using the current layout and column display settings.
- To switch to an existing settings profile, select the profile and click *Load*.
  The current layout and column display settings are updated from the settings saved in the profile.
- To modify an existing settings profile, select the profile and click *Update*.
  a. If you want to make the profile the default profile, select *Default*.
  b. Click *Update*.

  The selected profile is updated with the current layout and column display settings.
- To remove an existing settings profile, select the profile and click *Delete*.
  The selected profile is removed from the *Profiles* table.

2.4 Monitoring Data Provisioning Agents

You can monitor basic system information of an agent (such as CPU, memory, timestamps) and the time it last connected with the Data Provisioning Server.

When you first open the *Data Provisioning Agent Monitor*, it displays information for all agents and all adapters. You can select an agent to show information only for that agent.

Related Information

Manage Agents from the Data Provisioning Agent Monitor [page 27]
Manage Adapters from the Data Provisioning Agent Monitor [page 28]

Information Available on the Data Provisioning Agent Monitor

The *Data Provisioning Agent Monitor* displays information in two tables *Agent Monitor* and *Adapter Agent Mapping*. If you select an agent in the first table, the second table shows only the adapters registered to that agent.

Errors, warnings, and action execution information are displayed in the *Status Console*.

Table 2: Information Available in Agent Monitor Table

Column	Description
Agent Name	Name of the Data Provisioning Agent.
Agent Host	Name of the host on which the agent is running
Agent Port	Port that the agent uses to communicate with the Data Provisioning Server.
State	Indicates the state of the agent. The following states are possible: • CONNECTING • DISCONNECTED • CONNECTED
Last Connect	Elapsed time since the last connection from the Data Provisioning Server to the Data Provisioning Agent.
Last Connect Time	The last connect time from the Data Provisioning Server to the Data Provisioning Agent.
Adapters	Number of adapters defined for this Data Provisioning Agent.

Column	Description
Protocol	Type of network protocol used between the Data Provisioning Agent and the Data Provisioning Server. The following protocols are possible: • TCP • HTTP
Used Memory	Amount of memory currently used by the agent.
Used Swap Space	Amount of swap space currently used by the agent.
Free Memory	Amount of free memory on the agent host.
Free Swap Space	Amount of free swap space on the agent host.

Table 3: Information Available in Adapter Agent Mapping Table

Column	Description
Adapter Name	Name of the adapter registered to the Data Provisioning agent.
Agent Name	Name of the Data Provisioning Agent.

2.5 Monitoring Remote Subscriptions

The *Data Provisioning Remote Subscription Monitor* tile in the SAP HANA Cockpit provides you with information about how well data is being replicated to the Data Provisioning Server.

When you first open the *Data Provisioning Remote Subscription Monitor*, it displays information for all remote sources, all remote subscriptions, and all remote subscription statistics. You can select a remote source to view its corresponding remote subscriptions and remote subscription statistics.

Related Information

Remote Subscription States for Data Provisioning [page 13]
Information Available on the Data Provisioning Remote Subscription Monitor [page 13]
Processing Remote Source or Remote Subscription Exceptions [page 48]
Suspend and Resume Remote Sources [page 44]
Manage Remote Subscriptions [page 46]

2.5.1 Remote Subscription States for Data Provisioning

You create remote subscriptions when you want to listen for real-time changes to data that you replicated into the Data Provisioning Server. A remote subscription can have seven different states.

The following diagram shows where the states occur during the initial load and subsequent updates. The state names and their descriptions are in the table below.

Table 4: Remote Subscription States

State Name	Description
CREATED	Remote subscription is created by the replication task.
MAT_START_BEG_MARKER	The receiver is waiting for the begin marker that indicates the first changed data to queue while the initial load is running.
MAT_START_END_MARKER	The receiver queues the rows and is waiting for the end marker that indicates the last row of the initial load.
MAT_COMP_BEG_MARKER	The receiver is waiting for the begin marker that indicates the first row to queue after the initial load has completed.
MAT_COMP_END_MARKER	The receiver queues the changed rows and is waiting for the end marker that indicates the last row of the initial load. The initial load has completed and the end marker is sent to the adapter. If the state does not change to AUTO_CORRECT_CHANGE_DATA, the adapter or source system is slow in capturing the changes.
AUTO_CORRECT_CHANGE_DATA	When the end marker is received for the initial load, the applier loads the changed data captured (and queued during the initial load) to the target. If a lot of changes occurred after the initial load started, this state might take a long time to change to APPLY_CHANGE_DATA.
APPLY_CHANGE_DATA	All of the changes captured while the initial load was running have completed and are now loaded to the target.

2.5.2 Information Available on the Data Provisioning Remote Subscription Monitor

The Data Provisioning Remote Subscription Monitor provides information in tables.

The *Data Provisioning Remote Subscription Monitor* displays information in three tables *Remote Source Monitor*, *Remote Subscription Monitor* (for all sources or one source), and *Remote Subscription Statistics* (for all sources or one source). If you select a remote source in the first table, the second and third tables show only information for that remote source.

Errors, warnings, and action execution information are displayed in the *Status Console*.

Table 5: Information Available in Remote Source Monitor Table

Column	Description
Remote Source Name	Name of the remote source.
Adapter Name	Name of the adapter.

Column	Description
Location	Port that the agent uses to communicate with the Data Provisioning Server.
Agent Name	Name of the agent for this remote source.
CDC Status	The status of changed-data capture on this remote source. The following status values are possible: • OK • ERROR • SUSPENDED
Subscriptions	Number of subscriptions defined for this remote source.

Table 6: Information Available in Remote Subscription Monitor Table

Column	Description
Subscription Name	Name of the remote subscription.
Schema Name	Name of the schema (user name) in the remote source.
Remote Source Name	Name of the remote source for which this subscription is defined.
Design Time Name	Name of corresponding Design Time Object. **Tip** You can click the name of a design time object to open that object in the *Data Provisioning Design Time Object Monitor*.
Design Time Type	Type of corresponding Design Time Object. Values would be hdbflowgraph, hdbreptask, or empty.
Valid	Whether or not the remote subscription is valid.
State	Name of the state of the remote subscription. For more information, see Remote Subscription States for Data Provisioning [page 13].
Last Processed	Elapsed time since the last changed data was processed.
Last Processed Transaction Time	Time the last changed data was processed.
Subscription Type	Type of subscription. The following values are possible: • TABLE • VIRTUAL TABLE
Target Type	Type of target. The following values are possible: • TABLE • VIRTUAL TABLE

Table 7: Information Available in Remote Subscription Statistics Table

Column	Description
Subscription Name	Name of the remote subscription.
Schema Name	Name of the schema (user name) in the remote source.
Received Count	Total number of messages received by the Data Provisioning server.
Applied Count	Total number of messages applied.
Received Size	Total size of messages received by the Data Provisioning server.
Applied Size	Total size of messages applied.
Rejected Count	Total number of messages rejected.
Receiver Latency	Duration
Applier Latency	Duration
Last Message Received	Time the last message was received.
Last Message Applied	Time the last message was applied.

2.6 Monitoring Design Time Objects

The *Data Provisioning Design Time Object Monitor* tile in the SAP HANA cockpit provides you with information about your design time objects, including flowgraphs and replication tasks.For example, you can see the duration of a task execution for a flowgraph and how many records have been processed.

When you first open the *Data Provisioning Design Time Object Monitor*, it displays information for all design time objects. You can select a flowgraph or replication task to show information for only that object.

You can also sort and hide individual columns by right-clicking a row and selecting your display preferences.

Related Information

Execute Flowgraphs and Replication Tasks [page 49]
Schedule Flowgraphs and Replication Tasks [page 50]
Stop Non-Realtime Flowgraph Executions [page 51]

Information Available on the Data Provisioning Design Time Object Monitor

The *Data Provisioning Task Monitor* displays information in three tables: *Design Time Objects*, *Task Monitor*, and *Remote Subscription Monitor*. If you select a design time object in the first table, the other tables show information for only that object.

Errors, warnings, and action execution information are displayed in the *Status Console*.

Table 8: Information Available in Design Time Objects Table

Column	Description
Package	Name of the package to which the design time object belongs.
Object Name	Name of the design time object.
Object Type	Type of design time object, either “hdbreptask” or “hdbflowgraph”.
Schema Name	Name of the schema in which the object was created.
Realtime	'TRUE' if the object is a realtime object, else 'FALSE'.
Has Table Type Input	'TRUE' if the object is modeled with a table type as input. This means data would need to be passed (pushed) at execution time.

Table 9: Information Available in Task Monitor Table

Column	Description
Task Name	Name of the data provisioning task. ➔ Tip You can click the name of a task to open that task in the *Data Provisioning Task Monitor*.
Last Start Time	Day, date, and time when the last execution of the task started.
Last Duration	Total elapsed time from start to end for COMPLETED or FAILED tasks. Current elapsed time for RUNNING tasks.
Last Status	Current status of the last execution of the task. The following values are possible: • STARTING • RUNNING • COMPLETED • FAILED • CANCELLED (or, CANCELLING) ➔ Tip You can click the status to display additional information about the task execution.
Last Processed Records	Number of records that the last execution of the task has processed.

Table 10: Information Available in Remote Subscription Monitor Table

Column	Description
Subscription Name	Name of the remote subscription. ➔ Tip You can click the name of a remote subscription to open that subscription in the *Data Provisioning Remote Subscription Monitor*.
State	Name of the state of the remote subscription. For more information, see Remote Subscription States for Data Provisioning [page 13].
Status	Either 'ERROR' or 'SUCCESSFUL'. ➔ Tip You can click on the status of a remote subscription to process any exceptions for the subscription. For more information, see Processing Remote Source or Remote Subscription Exceptions [page 48].
Received Count	Total number of messages received by the Data Provisioning server.
Rejected Count	Total number of messages rejected.
Last Message Received Time	Time the last message was received.

2.7 Monitoring Data Provisioning Tasks

The *Data Provisioning Task Monitor* tile in the SAP HANA cockpit provides you with information about your replication tasks and transformation tasks.For example, you can see the duration of a task and how many records have been processed.

When you first open the *Data Provisioning Task Monitor*, it displays information for all tasks. You can select a task to show information for only that task.

You can also sort and hide individual columns by right-clicking a row and selecting your display preferences.

Related Information

Change Retention Period for Data Provisioning Task Monitor [page 21]

Start and Stop Data Provisioning Tasks [page 52]

Schedule Data Provisioning Tasks [page 53]

Information Available on the Data Provisioning Task Monitor

The *Data Provisioning Task Monitor* displays information in three tables: *Task Overview*, *Task Execution Monitor* and *Task Operation Execution Monitor*. If you select a task in the first table, the other tables show information for only that task.

> **Note**
>
> By default, the *Task Execution Monitor* and *Task Operation Execution Monitor* tables are limited to 500 entries, and the limit for each table can be controlled individually. After selecting a different task, you may need to refresh the tables to see all corresponding information.

Errors, warnings, and action execution information are displayed in the *Status Console*.

Table 11: Information Available in Task Overview Table

Column	Description
Task Type	Type of task. Derived from task plan
Task Name	Name of the data provisioning task
Task ID	Unique identifier for a task
SQL Security	Security model for the task, either 'DEFINER' or 'INVOKER'
Schema Name	Name of the schema in which the task was created.
Procedure Schema	If the task was created with a procedure instead of a plan, this attribute will contain the schema name of the stored procedure
Procedure Name	If the task was created with a procedure instead of a plan, this attribute will contain the name of the stored procedure
Plan Version	Version of the task plan
Owner Name	Owner of the task
Output Parameter Count	Number of output (tableType) parameters
Memory Size	Memory size of loaded task
Is Valid	'TRUE' if the task is in a valid state, 'FALSE' if it has been invalidated by a dependency
Is Realtime Task	'TRUE' if the task is a realtime task, else 'FALSE'
Is Read-Only	'TRUE' if the task is read only (has only table type outputs), 'FALSE' if it writes to non-table-type outputs
Input Parameter Count	Number of input (tableType) parameters
Has Table Type Input	'TRUE' if the task is modeled with a table type as input. This means data would need to be passed (pushed) at execution time
Has SDQ	'TRUE' if the task contains SDQ (Smart Data Quality) functionality

Column	Description
Design Time Type	Type of corresponding Design Time Object. Values would be hdbflowgraph, hdbreptask, or empty.
Design Time Name	Name of corresponding Design Time Object. ➔ Tip You can click the name of a design time oject to open that object in the *Data Provisioning Design Time Object Monitor*.
Create Time	The time that the task was created.
Comments	Description of the task, from the task plan

Table 12: Information Available in Task Execution Monitor Table

Column	Description
Task Name	Name of the Data Provisioning task.
Schema Name	Name of the schema in which the task was created.
Host	Name of the host on which the task is running.
Port	Port number that the task uses to communicate with the Data Provisioning Server.
Task Execution ID	Unique identifier for the task.
Partition Count	Number of logical partitions used for parallel processing in the task. If no partitions are defined for the task, the partition count is 1. ➔ Tip You can click the partition count to display additional information about the partitions used in the task execution.
Start Time	Day, date, and time when the task started.
End Time	Day, date, and time when the task ended.
Duration	Total elapsed time from start to end for COMPLETED or FAILED tasks. Current elapsed time for RUNNING tasks.

Column	Description
Status	Current status of the task. The following values are possible: • STARTING • RUNNING • COMPLETED • FAILED • CANCELLED (or, CANCELLING) **Tip** You can click the status to display additional information about the task execution.
Total Progress	Percentage completed.
Processed Records	Number of records that the task has processed.
Async	Indicates whether or not the task is running as a background task. Possible values are: • TRUE • FALSE
Parameters	Parameters passed to the task.
Hana User	Name of the user that started the execution of this task.
Application	Application user set in the session context.

Tip

Use the *Number of rows* menu to limit the number of task executions displayed at one time in the *Task Execution Monitor* table.

Table 13: Information Available in Task Operation Execution Monitor Table

Column	Description
Operation	Name of the operation that the task is currently performing. For a transformation task, the name of the operation is the name of the node that appears in the flowgraph.
Schema Name	Name of the schema (user name) in the remote source.
Task Name	Name of the task.
Task Execution ID	Identification number of the task.
Partition ID	Identification number of the logical partition used for parallel processing within the task.
Partition Name	Name of the logical partition used for parallel processing within the task.
Host	Name of the host on which the task is running.
Port	Port number that the task uses to communicate with the Data Provisioning Server.

Column	Description
Operation Type	Current type of operation. For example, the operation type can be Table Writer, Adapter, Projection, and so forth.
Start Time	Day, date, and time when the task started.
End Time	Day, date, and time when the task ended.
Duration	Total elapsed time from start to end.
Status	Current status of the task. The following values are possible: • STARTING • RUNNING • COMPLETED • FAILED
Progress	Percentage completed.
Processed Records	Number of records that the task has processed.
Side Effects	Indicates whether or not this operation generates side effect statistics. Possible values are: • TRUE • FALSE

➔ Tip

Use the *Number of rows* menu to limit the number of operations displayed at one time in the *Task Execution Operations Monitor* table.

2.7.1 Change Retention Period for Data Provisioning Task Monitor

Change the retention period for Data Provisioning statistics tables if you want to retain the data in the Task Execution Monitor and Task Operation Execution Monitor for longer than five minutes.

Context

The following parameters specify how long to keep the statistics data and when to delete them:

- The `task_data_retention_period` parameter specifies the period of time the data is kept in the statistics tables. This period is calculated from the time the task reached the COMPLETED, FAILED, or CANCELLED status. The default value is -1. A value of -1, means never delete the data.
- The `task_data_retention_period_check_interval` parameter specifies how often the data is actually deleted by the garbage collection thread. The default value is 300 seconds (five minutes).

To change the default values of these parameters, you must add a new section named *task_framework* to each of the `indexserver.ini`, `scriptserver.ini`, and the `xsengin.ini` files.

> **Note**
>
> If you need to change these options you can do so by running the following:

```
ALTER SYSTEM ALTER CONFIGURATION ('<server type>.ini', 'SYSTEM')
SET ('task_framework', ' task_data_retention_period_check_interval')='<in secs>'
WITH RECONFIGURE;
ALTER SYSTEM ALTER CONFIGURATION ('<server type>.ini', 'SYSTEM')
SET ('task_framework', ' task_data_retention_period')='<in secs>'
WITH RECONFIGURE;
```

Procedure

1. Login to SAP HANA studio as SYSTEM user.
2. In the *Systems* view, right-click the name of your SAP HANA server and choose *Configuration and Monitoring* > *Open Administration*.
3. Click the *Configuration* tab.
4. Select *indexserver.ini*, right-click and choose *Add Section*.
5. On the *Section Name* screen of the *Add Section Wizard*, enter **`task_framework`** for the section name, and click *Next*.
6. On the *Scope Selection* screen, select *System* from the drop-down list of *Assign Values to*, and click *Next*.
7. On the *Key Value Pairs* screen, enter **`task_data_retention_period`** in the *Key* field, and enter the number of seconds you want the statistics data to be kept in the statistics tables.

   > **Note**
   >
   - If you set the retention period to 0 (zero) or less than zero, statistics data will not be deleted.
   - If you set the retention period to a negative number, the default value (-1) will be used.

8. Click *Finish*.
9. Select *indexserver.ini*, right-click and choose *Add Parameter*.
10. On the *Add New Parameter* screen, enter **`task_data_retention_period_check_interval`** in the *Key* field, and enter the time interval for the Task Data Cleanup Process to run.

    > **Note**
    >
    If you set the task_data_retention_period_check_interval to less than 60 seconds, the default value (five minutes) will be used.

11. Select *scriptserver.ini* and do the following substeps.
    a. Right-click and choose *Add Section*.
    b. Repeat steps 5 through 10 for *scriptserver.ini*.
12. If your tasks will be called by the XS application, select *xsengine.ini* and do the following substeps.

a. Right-click and choose *Add Section*.
b. Repeat steps 5 through 10 for *xsengine.ini*.

2.8 Create Status Notifications

Create email notifications for various task, remote subscription, and design time object statuses.

Prerequisites

The user must have the following roles or privileges to create status notifications:

Table 14: Roles and Privileges

Action	Role or Privilege
Enable users to schedule task	Role: sap.hana.xs.admin.roles::JobSchedulerAdministrator
Configure SMTP	Role: sap.hana.xs.admin.roles::SMTPDestAdministrator
Create status notifications	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::NotificationAdministration

To activate status notifications, the following must occur:

- Scheduling must be enabled in the XS Job Admin Dashboard at `http://<host>:<port>/sap/hana/xs/admin/jobs`.
  The user that enables other users to schedule needs the role *sap.hana.xs.admin.roles::JobSchedulerAdministrator*.
- To create notifications, the Job *sap.hana.im.dp.monitor.jobs/checkNotifications* must be enabled in the XS Job Details page: `http://<host>:<port>/sap/hana/xs/admin/jobs/#/package/sap.hana.im.dp.monitor.jobs/job/checkNotifications`.
- A sender address must be configured in the monitor. If no sender address is configured, notification emails will not be sent.
  To configure the sender address, click *Settings* > *Set sender email address* in the monitor.
- The SAP HANA SMTP mail client must be correctly configured in the SMTP Configurations page at `http://<host>:<port>/sap/hana/xs/admin/index.html#smtp`.
  The user that configures the SMTP client needs the role *sap.hana.xs.admin.roles::SMTPDestAdministrator*

Context

You can create notifications for the following statuses:

Table 15: Supported Statuses

Object Type	Supported Statuses
Task execution	• COMPLETED • FAILED • CANCELLED
Remote subscription	• ERROR • WARNING
Design time object	• COMPLETED • FAILED • CANCELLED • ERROR • WARNING

Procedure

1. In the *Task Overview*, *Remote Subscription Monitor*, or *Design Time Objects* table, select the object for which you want to create a notification.
2. Click the *Notifications* button.
   The list of notifications for the object is displayed.
3. Click the *Add* button to create a new notification.
4. Specify a name, status conditions, and recipient email addresses for the notification.
5. If you want to enable the notification immediately, select *Is active*.
6. Click *Create Notification*.
   The new notification is added to the list of notifications for the object. When the conditions for the notification are met, users in the recipient list are sent an email containing details about the event that triggered the notification.

2.9 Create Monitoring Alerts

Create monitoring alerts for various functions.

Context

You can receive alerts for the following functions:

- Agent availability
- Agent memory usage
- Remote subscription exception

- Data Quality reference data
- Long-running tasks

The process for creating alerts is the same for each of these alerts in HANA studio.

> **Note**
>
> You can also configure alerts in the SAP HANA Cockpit by adding and clicking the *Configure Alerts* tile.

Procedure

1. In HANA studio, click the Alerts tab, and click the *Configure* button.
2. Configure the sender and recipient information.
3. Select the Alerts for which you want to receive email notifications.
4. Click OK.

3 Administering Data Provisioning

This section describes common tasks related to the ongoing administration of SAP HANA smart data integration and SAP HANA smart data quality.

3.1 Managing Agents and Adapters

The Data Provisioning Agent provides secure connectivity between the SAP HANA database and your on-premise sources. Adapters are registered on the agent and manage the connection to your source.

3.1.1 Manage Agents from the Data Provisioning Agent Monitor

You can use the *Data Provisioning Agent Monitor* to perform basic administration tasks such as registering, altering, or dropping Data Provisioning Agents.

Prerequisites

The user must have the following roles or privileges to manage agents.

Table 16: Roles and Privileges

Action	Role or Privilege
Add Data Provisioning Agent	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::CreateAgent • System privilege: AGENT ADMIN
Alter Data Provisioning Agent	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AlterAgent • System privilege: AGENT ADMIN
Remove Data Provisioning Agent	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::DropAgent • System privilege: AGENT ADMIN

Context

Use the buttons in the *Agent Monitor* table to perform the action.

Procedure

- Click *Create Agent* to register a new agent with the SAP HANA system.
  a. Specify the name of the agent and relevant connection information.
  b. If the agent uses a secure SSL connection, check *Enable SSL*.
  c. If you want to assign the agent to an existing agent group, select the group under *Agent Group*.
  d. Click *Create Agent*.

  The new agent appears in the *Agent Monitor* table.
- Click *Alter Agent* to make connection configuration changes on an agent already registered in the SAP HANA system.
  a. Specify the new connection information for the agent. You cannot change the name or connection protocol for an existing agent.
  b. If the agent uses a secure SSL connection, check *Enable SSL*.

c. If you want to assign the agent to an existing agent group, select the group under *Agent Group*.
d. Click *Alter Agent*.

The updated agent information appears in the *Agent Monitor* table.

- Click *Drop Agent* to remove an agent from the SAP HANA system.
  a. To automatically drop any dependent objects such as registered adapters, choose the *CASCADE option*. You cannot remove an agent while it has dependent objects such as registered adapters. First manually remove the adapters from the agent, or check the *CASCADE option*.
  b. Click *Drop Agent*.

  The agent is removed from the *Agent Monitor* table. If the agent was assigned to an agent group, it is also removed from the agent group.

Related Information

Smart Data Integration: ALTER AGENT [page 134]
Smart Data Integration: CREATE AGENT [page 143]
Smart Data Integration: CREATE AGENT GROUP [page 145]
Smart Data Integration: DROP AGENT [page 153]
Smart Data Integration: DROP AGENT GROUP [page 154]

3.1.2 Manage Adapters from the Data Provisioning Agent Monitor

You can use the Data Provisioning Agent Monitor to perform basic administration tasks such as adding adapters to or removing adapters from a Data Provisioning Agent instance.

Prerequisites

The user must have the following roles or privileges to manage adapters.

Table 17: Roles and Privileges

Action	Role or Privilege
Add adapter	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AddLocationToAdapter • System privilege: ADAPTER ADMIN
Remove adapter	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::RemoveLocationFromAdapter • System privilege: ADAPTER ADMIN

Action	Role or Privilege
Update adapters	• Role: sap.hana.im.dp.monitor.roles::Operations • System privilege: ADAPTER ADMIN

Context

Use the buttons in the *Agent Monitor* and *Agent Adapter Mapping* tables to perform the action.

Procedure

- To add adapters to an agent instance, select the agent and click *Add Adapters* in the *Agent Monitor* table.
  a. Select the desired adapters from the list of adapters deployed on the agent instance.
  b. Click *Add Adapters*.

  The selected adapters appear in the *Adapter Agent Mapping* table.
- To remove an adapter from an agent instance, select the adapter and click *Remove Location* in the *Adapter Agent Mapping* table.
  a. If the adapter is registered on only one agent instance, you can choose to remove with the *CASCADE option*.
  b. Click *Remove Location*.

  The adapter is removed from the *Adapter Agent Mapping* table.
- To update all adapters for an agent, select the agent and click *Update Adapters* in the *Agent Monitor*.
  All adapters registered for the selected agent are refreshed, and any new capabilities can be used by SAP HANA.
- To update a single adapter, select the adapter and click *Update* in the *Adapter Agent Mapping* table.
  The selected adapter is refreshed, and any new capabilities can be used by SAP HANA.

Related Information

Smart Data Integration: CREATE ADAPTER [page 142]
Smart Data Integration: DROP ADAPTER [page 152]

3.1.3 Back Up the Data Provisioning Agent Configuration

You can back up your Data Provisioning Agent configuration by copying key static configuration files to a secure location. You can use this backup to restore communication between the SAP HANA server and the Data Provisioning Agent.

> **Note**
>
> This backup can be restored only to an agent host with the same fully-qualified domain name as the original agent. You cannot use the backup to transport configuration settings between agents with different fully-qualified domain names.
>
> For example, you cannot use a backup from an agent on **host1**.mydomain.com to restore settings to an agent on **host2**.mydomain.com.

> **Restriction**
>
> CDC state information for Log Reader adapters cannot be backed up and restored.

Unless specified, all files and directories that you need to back up are located under `<DPAgent_root>`:

- `dpagent.ini`
- `dpagentconfig.ini`
- `sec`
- `secure_storage`
- `ssl/cacerts`
- `configuration/com.sap.hana.dp.adapterframework`
- `lib/`
- `camel/`
- `LogReader/config`
- `LogReader/sybfilter/system/<platform>/LogPath.cfg`

3.1.4 Uninstall the Data Provisioning Agent

You can uninstall the Data Provisioning Agent in graphical mode on Windows and Linux.

Procedure

- On Windows, call the uninstallation program from Control Panel.

  Programs and Features > *SAP HANA Data Provisioning Agent* > *Uninstall*
- On Linux, call the uninstallation program from the command line.
  a. Navigate to the `<DPAgent_root>/install` directory.

     For example, `/usr/sap/dataprovagent/install`.
  b. Run **`./hdbuninst --main SDB::Install::App::Gui::Uninstallation::main --path "<DPAgent_root>"`**.

Results

The Data Provisioning Agent is uninstalled from the system.

Next Steps

After uninstalling the agent, several files and directories generated by the agent during runtime may be left in place. If you choose, you can safely remove these remaining files and directories manually.

Remove the following files and directories from `<DPAgent_root>`:

- `configTool/`
- `configuration/`
- `install/`
- `log/`
- `LogReader/`
- `workspace/`

Related Information

Uninstall from the Command Line [page 31]

3.1.4.1 Uninstall from the Command Line

If you cannot use or do not want to use the graphical uninstallation mode, you can uninstall the Data Provisioning Agent using the command line.

Procedure

1. Navigate to the `<DPAgent_root>/install` directory.

   For example, `C:\usr\sap\dataprovagent\install` or `/usr/sap/dataprovagent/install`.
2. Call the uninstallation program.
   - On Windows, run **`hdbuninst.exe --path "<DPAgent_root>"`**.
   - On Linux, run **`./hdbuninst --path "<DPAgent_root>"`**.

Results

The Data Provisioning Agent is uninstalled from the system without displaying the graphical uninstallation manager.

3.2 Managing Agent Groups

Agent grouping provides fail-over capabilities by combining individual Data Provisioning Agents installed on separate host systems.

> ⚠ Restriction
>
> Data Provisioning Agent groups do not provide load balancing.

Planning considerations

Before configuring agents in a group, review the following considerations and limitations:

- Each agent in a group must be installed on a different host system.
- All agents in a group must have identical adapter configurations.
- All agents in a group must use the same communication protocol. You cannot mix on-premise agents (TCP) and cloud-based agents (HTTP) in a single group.

Agent Group Failover Behavior [page 33]
When an agent node in an agent group is inaccessible for a time longer than the configured heart beat time limit, the Data Provisioning Server chooses a new active agent within the group and resumes replication for any remote subscriptions active on the original agent.

Create or Remove an Agent Group [page 34]
You can create a new agent group or remove an existing group in the *Data Provisioning Agent Monitor*.

Manage Agent Nodes in an Agent Group [page 35]
You can manage the agent nodes that belong to an agent group in the *Data Provisioning Agent Monitor*.

Add Adapters to an Agent Group [page 36]
Before you can create remote sources on an agent group, you must add adapters to the group in the SAP HANA Web-based Development Workbench.

Configure Remote Sources in an Agent Group [page 37]
To receive the benefits of fail over from an agent group, you must configure your remote sources on the agent group.

3.2.1 Agent Group Failover Behavior

When an agent node in an agent group is inaccessible for a time longer than the configured heart beat time limit, the Data Provisioning Server chooses a new active agent within the group and resumes replication for any remote subscriptions active on the original agent.

Initial and batch load requests to a remote source configured on the agent group are routed to the first accessible agent in the group.

> ⚠ Restriction
>
> Failover is not supported for initial and batch load requests. you must restart the initial load following a failure due to agent unavailability.

Although no user action is required for automatic failover within an agent group, you may choose to monitor the current master and slave agent node information.

- To query the current master agent node name for a remote source:

```
SELECT AGENT_NAME FROM "SYS"."M_REMOTE_SOURCES_" WHERE "REMOTE_SOURCE_OID" =
(SELECT REMOTE_SOURCE_OID FROM "SYS"."REMOTE_SOURCES_" WHERE REMOTE_SOURCE_NAME
= '<remote_source_name>');
```

- To query a list of all agent and agent group names:

```
SELECT AGENT_NAME,AGENT_GROUP_NAME FROM SYS."AGENTS";
```

> ⚠ Caution
>
> If all nodes in an agent group are down, replication cannot continue and must be recovered after one or more agent nodes are available.

Restarting Agent Nodes in an Agent Group

Restarting nodes in an agent group does not impact active replication tasks.

For the master agent node, stopping or restarting the agent triggers the agent group's failover behavior, and a new active master node is selected.

3.2.2 Create or Remove an Agent Group

You can create a new agent group or remove an existing group in the *Data Provisioning Agent Monitor*.

Prerequisites

The user who creates or removes the agent group must have the following roles or privileges.

Table 18: Roles and Privileges

Action	Role or Privilege
Create agent group	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::CreateAgentGroup • System privilege: AGENT ADMIN
Remove agent group	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::DropAgentGroup • System privilege: AGENT ADMIN

Context

Use the buttons in the *Agent Group* table to create or remove an agent group.

Procedure

- Click *Create* to create a new agent group.
  Specify the name for the new agent group, and click *Create Agent Group*.
  The new agent group appears in the *Agent Group* table.
- Select the agent group and click *Drop* to remove an existing agent group.

  > **Note**
  > When you remove an agent group, any agent nodes for the group will first be removed from the group. Agents cannot be removed from the group if there are active remote subscriptions.

  Any agent nodes are removed from the group, and the group is removed from the *Agent Group* table.

Related Information

Smart Data Integration: CREATE AGENT GROUP [page 145]

Smart Data Integration: DROP AGENT GROUP [page 154]
Smart Data Integration: CREATE AGENT GROUP [page 145]
Smart Data Integration: DROP AGENT GROUP [page 154]

3.2.3 Manage Agent Nodes in an Agent Group

You can manage the agent nodes that belong to an agent group in the *Data Provisioning Agent Monitor*.

Prerequisites

The user must have the following roles or privileges to manage agent nodes.

Table 19: Roles and Privileges

Action	Role or Privilege
Create agent	• Role: sap.hana.im.dp.roles::Operations • Application privilege: sap.hana.im.dp.monitor::CreateAgent • System privilege: AGENT ADMIN
Add agent to agent group	• Role: sap.hana.im.dp.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AlterAgent • System privilege: AGENT ADMIN
Remove agent from agent group	• Role: sap.hana.im.dp.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AlterAgent • System privilege: AGENT ADMIN

Context

Use the buttons in the *Agent Monitor* and *Agent Group* tables to perform the action.

> **Tip**
>
> Select an agent group in the *Agent Group* table to display its nodes in the *Agent Monitor* table.

Procedure

- To register a new agent with the SAP HANA system and add it to an existing agent group, click *Create Agent*. When specifying the parameters for the agent, select the agent group from the *Agent Group* list.

  The new agent appears in the *Agent Monitor* table.

- To modify the group assignment for an existing agent, click *Alter Agent*.
  - Select the new agent group from the *Agent Group* list.
    If you are assigning the agent to a different group, select the empty entry for *Enable SSL* to avoid connection issues when the group is changed.
  - To remove the agent from an agent group, select the empty entry from the *Agent Group* list.

  The group for the agent is displayed in the *Agent Monitor* table.
- To add multiple existing agents to an agent group, select the group in the *Agent Group* table and click *Add Agents*.
  a. Select the agents that you want to add to the group.
  b. Click *Add Agents*.

  The selected agents are assigned to the agent group, and all associated entries in the *Agent Monitor* and *Agent Group* tables are updated.

Related Information

Smart Data Integration: CREATE AGENT [page 143]
Smart Data Integration: ALTER AGENT [page 134]
Manage Agents from the Data Provisioning Agent Monitor [page 27]
Smart Data Integration: CREATE AGENT [page 143]
Smart Data Integration: ALTER AGENT [page 134]

3.2.4 Add Adapters to an Agent Group

Before you can create remote sources on an agent group, you must add adapters to the group in the SAP HANA Web-based Development Workbench.

Prerequisites

The user who adds an adapter must have the following roles or privileges.

Table 20: Roles and Privileges

Action	Role or Privilege
Add adapter to agent group	• System privilege: `ADAPTER ADMIN`

Procedure

1. Open the SQL console in the SAP HANA Web-based Development Workbench.

2. If you do not know the agent names, query the system for for a list of agents and agent groups.

```
SELECT AGENT_NAME,AGENT_GROUP_NAME FROM SYS."AGENTS";
```

3. Create the adapter on the first agent node.

```
CREATE ADAPTER "<adapter_name>" AT location agent "<agent1_name>";
```

4. Add the agent to each additional agent node in the agent group.

```
ALTER ADAPTER "<adapter_name>" ADD location agent "<agent#_name>";
```

Related Information

Smart Data Integration: CREATE ADAPTER [page 142]
Smart Data Integration: ALTER ADAPTER [page 132]

3.2.5 Configure Remote Sources in an Agent Group

To receive the benefits of fail over from an agent group, you must configure your remote sources on the agent group.

Related Information

Smart Data Integration: CREATE REMOTE SOURCE [page 147]
Smart Data Integration: ALTER REMOTE SOURCE [page 135]

Configure Remote Sources in the Web-based Development Workbench

Procedure

- To create a new remote source in an agent group:
  a. In the Catalog editor, right-click the *Provisioning* > *Remote Sources* folder, and choose *New Remote Source*.
  b. Enter the required configuration information for the remote source, including the adapter name.
  c. In the *Location* dropdown, choose *agent group*, and select the agent group name.
  d. Click *Save*.
- To add an existing remote source to an agent group:

a. In the Catalog editor, select the remote source in the *Provisioning* > *Remote Sources* folder.
b. In the *Location* dropdown, choose *agent group*, and select the agent group name.
c. Click *Save*.

Related Information

Create a Remote Source in the Web-based Development Workbench [page 41]

Configure Remote Sources in a SQL Console

Procedure

1. Open the SQL console in the SAP HANA studio or Web-based Development Workbench.
2. Execute the CREATE or ALTER REMOTE SOURCE statement in the SQL console.
   - To create a new remote source in the group:

     ```
     CREATE REMOTE SOURCE <source_name> ADAPTER <adapter_name> AT LOCATION AGENT
     GROUP <group_name> <configuration_clause> <credential_clause>
     ```

   - To add an existing remote source to the group:

     ```
     ALTER REMOTE SOURCE <source_name> ADAPTER <adapter_name> AT LOCATION AGENT
     GROUP <group_name> <configuration_clause> <credential_clause>
     ```

     > **Note**
     >
     > If you are changing only the location for the remote source, you can omit the ADAPTER and CONFIGURATION clauses:
     >
     > ```
     > ALTER REMOTE SOURCE <source_name> AT LOCATION AGENT GROUP <group_name>
     > <credential_clause>
     > ```

   For more information about the CREATE REMOTE SOURCE and ALTER REMOTE SOURCE statements, see the *SAP HANA SQL and System Views Reference*.

Alter Remote Source Clauses

When you use ALTER REMOTE SOURCE to modify a remote source, you must specify the configuration and credential details as XML strings.

Example Credential Clause

```
WITH CREDENTIAL TYPE 'PASSWORD' USING '<CredentialEntry name="credential">
     <user><username></user>
     <password><password></password>
</CredentialEntry>'
```

Example Configuration Clause

```
CONFIGURATION '<?xml version="1.0" encoding="UTF-8"?>
    <ConnectionProperties name="configurations">
        <PropertyGroup name="generic">
            <PropertyEntry name="map_char_types_to_unicode">false</PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="database">
            <PropertyEntry name="cdb_enabled">false</PropertyEntry>
            <PropertyEntry name="pds_use_tnsnames">false</PropertyEntry>
            <PropertyEntry name="pds_host_name"><db_hostname></PropertyEntry>
            <PropertyEntry name="pds_port_number">1521</PropertyEntry>
            <PropertyEntry name="pds_database_name">ORCL</PropertyEntry>
            <PropertyEntry name="cdb_service_name"></PropertyEntry>
            <PropertyEntry name="pds_service_name"></PropertyEntry>
            <PropertyEntry name="pds_tns_filename"></PropertyEntry>
            <PropertyEntry name="pds_tns_connection"></PropertyEntry>
            <PropertyEntry name="cdb_tns_connection"></PropertyEntry>
            <PropertyEntry name="_pds_tns_connection_with_cdb_enabled"></
PropertyEntry>
            <PropertyEntry name="pds_byte_order"></PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="schema_alias_replacements">
            <PropertyEntry name="schema_alias"></PropertyEntry>
            <PropertyEntry name="schema_alias_replacement"></PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="security">
            <PropertyEntry name="pds_use_ssl">false</PropertyEntry>
            <PropertyEntry name="pds_ssl_sc_dn"></PropertyEntry>
            <PropertyEntry name="_enable_ssl_client_auth">false</PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="jdbc_flags">
            <PropertyEntry name="remarksReporting">false</PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="cdc">
            <PropertyGroup name="databaseconf">
                <PropertyEntry name="pdb_archive_path"></PropertyEntry>
                <PropertyEntry name="pdb_supplemental_logging_level">table</
PropertyEntry>
            </PropertyGroup>
            <PropertyGroup name="parallelscan">
                <PropertyEntry name="lr_parallel_scan">false</PropertyEntry>
                <PropertyEntry name="lr_parallel_scanner_count"></PropertyEntry>
```

```
                <PropertyEntry name="lr_parallel_scan_queue_size"></PropertyEntry>
                <PropertyEntry name="lr_parallel_scan_range"></PropertyEntry>
            </PropertyGroup>
            <PropertyGroup name="logreader">
                <PropertyEntry name="skip_lr_errors">false</PropertyEntry>
                <PropertyEntry name="lr_max_op_queue_size">1000</PropertyEntry>
                <PropertyEntry name="lr_max_scan_queue_size">1000</PropertyEntry>
                <PropertyEntry name="lr_max_session_cache_size">1000</PropertyEntry>
                <PropertyEntry name="scan_fetch_size">10</PropertyEntry>
                <PropertyEntry name="pdb_dflt_column_repl">true</PropertyEntry>
                <PropertyEntry name="pdb_ignore_unsupported_anydata">false</
PropertyEntry>
                <PropertyEntry name="pds_sql_connection_pool_size">15</
PropertyEntry>
                <PropertyEntry name="pds_retry_count">5</PropertyEntry>
                <PropertyEntry name="pds_retry_timeout">10</PropertyEntry>
            </PropertyGroup>
        </PropertyGroup>
</ConnectionProperties>'
```

> **Note**
>
> Changing user names are not allowed when remote source is suspended.

3.3 Managing Remote Sources and Subscriptions

Remote sources establish the connection between a data provisioning adapter and your source system. Remote subscriptions monitor a remote source for real-time changes to data replicated into the Data Provisioning Server.

Remote sources are generally created by an administrator, and can then be used for remote subscriptions in replication tasks and flowgraphs created by a data provisioning modeler.

Create a Remote Source [page 41]
Using SAP HANA smart data integration, you set up an adapter that can connect to your source database, then create a remote source to establish the connection.

Suspend and Resume Remote Sources [page 44]
You can suspend and resume capture and distribution for remote sources within the *Data Provisioning Remote Subscription Monitor*.

Alter Remote Source Parameters [page 46]
You can modify some remote source parameters while the remote source is suspended.

Manage Remote Subscriptions [page 46]
You can drop, queue, distribute, and reset remote subscriptions within the *Data Provisioning Remote Subscription Monitor*.

Processing Remote Source or Remote Subscription Exceptions [page 48]
Remote subscription exceptions need to be evaluated and resolved.

3.3.1 Create a Remote Source

Using SAP HANA smart data integration, you set up an adapter that can connect to your source database, then create a remote source to establish the connection.

Prerequisites

- The user who creates the remote source must have the following roles or privileges.

  Table 21: Roles and Privileges

Action	Role or Privilege
Create a remote source	System privilege: `CREATE REMOTE SOURCE`

- The Data Provisioning Server must be enabled.
- The Data Provisioning Agent must be installed and configured.
- The adapter must be configured and registered with SAP HANA.

Context

You can create a remote source in more than one way.

Related Information

Create a Remote Source in the Web-based Development Workbench [page 41]
Create a Remote Source in the SQL Console [page 42]
Create Credentials for a Secondary User [page 44]

3.3.1.1 Create a Remote Source in the Web-based Development Workbench

How to create a remote source in SAP HANA smart data integration with the Web-based Development Workbench user interface.

Prerequisites

The user who creates the remote source must have the following roles or privileges.

Table 22: Roles and Privileges

Action	Role or Privilege
Create a remote source	• System privilege: `CREATE REMOTE SOURCE`

Procedure

1. In the Web-based Development Workbench Catalog editor, expand the *Provisioning* node.
2. Right-click the *Remote Sources* folder and choose *New Remote Source*.
3. Enter the required information including the adapter and Data Provisioning Agent names.

   Regarding user credentials, observe the following requirements:
   - A remote source created with a secondary user can only be used for querying virtual tables.
   - If the remote source will be used for designing a `.hdbreptask` or `.hdbflowgraph` enabled for real time, use **`technical user`**
   - If you create a remote subscription using the CREATE REMOTE SUBSCRIPTION SQL statement, use **`technical user`**.
4. Select *Save*.

Related Information

CREATE REMOTE SOURCE Statement [Smart Data Integration] (SAP HANA SQL and System Views Reference) [page 147]

3.3.1.2 Create a Remote Source in the SQL Console

In SAP HANA smart data integration, in addition to using the Web-based Development Workbench to create a remote source, you can also do so using the SQL console.

Prerequisites

The user who creates the remote source must have the following roles or privileges.

Table 23: Roles and Privileges

Action	Role or Privilege
Create a remote source	• System privilege: `CREATE REMOTE SOURCE`

Context

To create an remote source using the SQL console, you need to know the connection information that is particular to your source. For an existing remote source, the connection information is in an XML string in the CONFIGURATION statement.

For your adapter, refer to the remote source configuration topic for that adapter in this guide to see its sample SQL code. Change the variables to the correct values for your remote source.

The example at the end of this topic illustrates the basic CONFIGURATION connection information XML string for a Microsoft SQL Server adapter.

After you create the remote source:

- Note that if you have recently updated the Data Provisioning Agent, the connection information XML string might have also been updated for your adapter. Therefore, refresh the adapter to get up-to-date connection information.
- To view the connection information for an existing remote source, execute `SELECT * FROM "PUBLIC"."REMOTE_SOURCES"`. In the resulting view, look in the CONNECTION_INFO column. (To ensure you can view the entire XML string in the CONNECTION_INFO column, in SAP HANA preferences enable the setting *Enable zoom of LOB columns*.)
- To view all of the configuration parameters for a given adapter type, execute `SELECT * FROM "PUBLIC"."ADAPTERS"`. In the resulting view, look in the CONFIGURATION column. This information can be useful if you want to, for example, determine the PropertyEntry name for a given parameter in the user interface (displayName). For example:

```
<PropertyEntry name="pds_database_name" displayName="Database
Name"><database_name></PropertyEntry>
<PropertyEntry name="pdb_dcmode" displayName="Database Data Capture Mode">MSCDC</
PropertyEntry>
```

Example

Sample Code

```
CREATE REMOTE SOURCE "MySQLServerSource" ADAPTER "MssqlLogReaderAdapter" AT
LOCATION AGENT "MyAgent"
CONFIGURATION
'<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ConnectionProperties name="configurations">
<PropertyGroup name="data_type_conversion" displayName="Data Type Conversion">
<PropertyEntry name="map_char_types_to_unicode" displayName="Always Map
Character Types to Unicode">false</PropertyEntry>
<PropertyEntry name="map_time_to_timestamp" displayName="Map SQL Server Data
Type Time to Timestamp">true</PropertyEntry>
</PropertyGroup>
<PropertyGroup name="database" displayName="Database">
<PropertyEntry name="pds_server_name" displayName="Host">myserver.sap.corp</
PropertyEntry>
<PropertyEntry name="pds_port_number" displayName="Port Number">1433</
PropertyEntry>
<PropertyEntry name="pds_database_name" displayName="Database Name">mydb</
PropertyEntry>
<PropertyEntry name="pdb_dcmode" displayName="Database Data Capture
Mode">MSCDC</PropertyEntry>
</PropertyGroup>
<PropertyGroup name="cdc" displayName="CDC Properties">
<PropertyGroup name="logreader" displayName="LogReader">
```

```
<PropertyEntry name="skip_lr_errors" displayName="Ignore log record processing
errors">false</PropertyEntry>
</PropertyGroup>
</PropertyGroup>
</ConnectionProperties>
'WITH CREDENTIAL TYPE 'PASSWORD' USING
'<CredentialEntry name="credential">
<user>myuser</user>
<password>mypassword</password>
</CredentialEntry>'
```

Related Information

CREATE REMOTE SOURCE Statement [Smart Data Integration] (SAP HANA SQL and System Views Reference) [page 147]

3.3.1.3 Create Credentials for a Secondary User

The syntax for creating secondary user credentials for SAP HANA smart data integration adapters is different than the syntax for SAP HANA system adapters.

The syntax for creating secondary user credentials for SAP HANA smart data integration adapters is as follows.

```
create credential for user <user_name> component 'SAPHANAFEDERATION'
purpose <remote_source_name> type 'PASSWORD' using
<CredentialEntry name="credential">
      <user><user_name></user>
      <password><password></password>
</CredentialEntry>
```

3.3.2 Suspend and Resume Remote Sources

You can suspend and resume capture and distribution for remote sources within the *Data Provisioning Remote Subscription Monitor*.

Prerequisites

The user must have the following roles or privileges to suspend and resume capture and distribution:

Table 24: Roles and Privileges

Action	Role or Privilege
Suspend capture or distribution	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AlterRemoteSource • Object privilege: ALTER on the remote source
Resume capture or distribution	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AlterRemoteSource • Object privilege: ALTER on the remote source

Context

Use the *Alter Remote Source* button in the monitor to perform the action.

Procedure

1. Select the remote source in the *Remote Source Monitor* table and click *Alter Remote Source*.
2. Click *Suspend* or *Resume* for CAPTURE or DISTRIBUTION.
   Confirmation of the action is displayed in the status console.
3. Close the *Alter Remote Source* dialog.

Results

Capture or distribution for the selected remote source is suspended or resumed.

Related Information

Smart Data Integration: ALTER REMOTE SOURCE [page 135]

3.3.3 Alter Remote Source Parameters

You can modify some remote source parameters while the remote source is suspended.

Context

In the *Installation and Configuration Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality*, see each adapter's remote source description topic regarding which parameters you can modify when a remote source is suspended.

> **Note**
>
> You cannot change the User Name parameter when the remote source is suspended.

Procedure

1. In the *Data Provisioning Remote Subscription Monitor*, suspend capture on the remote source.
2. In the SAP HANA Web-based Development Workbench catalog, change the intended remote source parameters.
3. Re-enter the credentials for the remote source and save the changes.
4. Resume capture on the remote source.

3.3.4 Manage Remote Subscriptions

You can drop, queue, distribute, and reset remote subscriptions within the *Data Provisioning Remote Subscription Monitor*.

Prerequisites

The user must have the following roles or privileges to manage remote subscriptions:

Table 25: Roles and Privileges

Action	Role or Privilege
Reset, queue, or distribute remote subscription	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::AlterRemoteSubscription • Object privilege: ALTER on the remote subscription

Action	Role or Privilege
Drop remote subscription	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::DropRemoteSubscription • Object privilege: DROP on the remote subscription

Context

Use the buttons in the *Remote Subscription Monitor* table to perform the action.

Procedure

1. Select the remote subscription in the *Remote Subscription Monitor* table.
2. Click *Queue*, *Distribute*, *Reset*, or *Drop*.

> **Note**
>
> A warning appears if you attempt to drop a remote subscription that is used by any flowgraphs or replication tasks. Click *Drop* if you want to continue and drop the remote subscription anyway.

Results

The remote subscription is queued, distributed, or reset. If you drop a remote subscription, the subscription is removed from the *Remote Subscription Monitor* table.

Related Information

Smart Data Integration: ALTER REMOTE SUBSCRIPTION [page 139]
Smart Data Integration: CREATE REMOTE SUBSCRIPTION [page 148]
Smart Data Integration: DROP REMOTE SUBSCRIPTION [page 155]

3.3.5 Processing Remote Source or Remote Subscription Exceptions

Remote subscription exceptions need to be evaluated and resolved.

In the case of an error or if the row count on the target table does not match, for example, you will need to look at the Exceptions table and process the entries.

Use the following syntax:

```
PROCESS REMOTE SUBSCRIPTION EXCEPTION <exception_oid> IGNORE|RETRY;
```

To process a remote source or remote subscription exception using the monitoring UI:

1. Click the status of the remote source or remote subscription.
2. Select the error.
3. Click either *Retry Operations* or *Ignore Error.*

Related Information

Smart Data Integration: PROCESS REMOTE SUBSCRIPTION EXCEPTION [page 158]

3.4 Managing Design Time Objects

Design time objects such as flowgraphs and replication tasks manage the replication and transformation of data in SAP HANA smart data integration and SAP HANA smart data quality.

Execute Flowgraphs and Replication Tasks [page 49]
You can execute design time objects including flowgraphs and replication tasks from the *Data Provisioning Design Time Object Monitor*.

Schedule Flowgraphs and Replication Tasks [page 50]
You can schedule design time objects including flowgraphs and replication tasks within the *Data Provisioning Design Time Object Monitor*.

Stop Non-Realtime Flowgraph Executions [page 51]
You can stop the execution of non-realtime flowgraphs within the *Data Provisioning Design Time Object Monitor*.

Start and Stop Data Provisioning Tasks [page 52]
You can start and stop tasks within the *Data Provisioning Task Monitor*.

Schedule Data Provisioning Tasks [page 53]
You can schedule tasks within the *Data Provisioning Task Monitor*.

3.4.1 Execute Flowgraphs and Replication Tasks

You can execute design time objects including flowgraphs and replication tasks from the *Data Provisioning Design Time Object Monitor*.

> ⚠ Restriction
>
> Realtime flowgraphs and replication tasks cannot be executed from the *Data Provisioning Design Time Object Monitor*.

Prerequisites

The user must have the following roles or privileges to schedule flowgraphs and replication tasks.

Table 26: Roles and Privileges

Action	Role or Privilege
Execute flowgraph or replication task	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::ExecuteDesignTimeObject • Object privilege: EXECUTE on the object schema • Object privilege: Any additional object privileges needed within the task (for example, ALTER, CREATE ANY, DELETE, DROP, EXECUTE, INDEX, INSERT, and so on.)

Procedure

1. Select the flowgraph or replication task in the *Design Time Objects* table.
2. Click *Execute*.
   a. If the object uses table type parameters, select the tables to use when executing the object.
   b. If the object uses variable parameters, specify the values to use when executing the object.
   c. Click *Execute*.

Results

The object execution begins and the task appears in the *Task Monitor* table.

Related Information

Smart Data Integration: START TASK [page 160]

3.4.2 Schedule Flowgraphs and Replication Tasks

You can schedule design time objects including flowgraphs and replication tasks within the *Data Provisioning Design Time Object Monitor*.

> ⚠ Restriction
>
> Realtime flowgraphs and replication tasks cannot be scheduled from the *Data Provisioning Design Time Object Monitor*.

Prerequisites

The user must have the following roles or privileges to schedule flowgraphs and replication tasks.

Table 27: Roles and Privileges

Action	Role or Privilege
Enable users to schedule design time objects	Role: sap.hana.xs.admin.roles::JobSchedulerAdministrator
Schedule flowgraph or replication task	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::ScheduleDesignTimeObject

To activate task scheduling, the following must occur:

- Scheduling needs to be enabled via XS Job Admin Dashboard /sap/hana/xs/admin/jobs/ (The user that enables other users to schedule needs the role sap.hana.xs.admin.roles::JobSchedulerAdministrator).
- To schedule design time objects, the Job sap.hana.im.dp.monitor.jobs::scheduleTask needs to be enabled in the XS Job Details page: /sap/hana/xs/admin/jobs/#/package/sap.hana.im.dp.monitor.jobs/job/scheduleTask.

Procedure

1. Select the flowgraph or replication task in the *Design Time Objects* table.
2. Click the *Schedules* button.
   The *Schedules* dialog appears.
3. To create a new schedule for the task, click *Add*.
   a. Select the frequency (once or recurring), interval if recurring (year, month, week, day, hour, minute, second), and the time (local, not server time or UTC) for the object execution.
   b. If the object uses table type parameters, select the tables to use when executing the object.
   c. If the object uses variable parameters, specify the values to use when executing the object.
   d. Click *Schedule*.
      The new schedule is added to the list of schedules for the object.

4. To remove an existing schedule, select the schedule and click *Delete*.
   The schedule is removed from the list of schedules for the object.
5. Close the *Schedules* dialog.

Results

Your object executes as scheduled and you can monitor the results of each execution of the object.

3.4.3 Stop Non-Realtime Flowgraph Executions

You can stop the execution of non-realtime flowgraphs within the *Data Provisioning Design Time Object Monitor*.

Prerequisites

The user must have the following roles or privileges to stop flowgraph execution.

Table 28: Roles and Privileges

Action	Role or Privilege
Stop flowgraph execution	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::StopTask

Procedure

1. Select the task for the flowgraph in the *Task Monitor* table.
2. Click *Stop*.

Results

The selected flowgraph execution instance is stopped.

Related Information

Smart Data Integration: CANCEL TASK [page 140]

Smart Data Integration: START TASK [page 160]

3.4.4 Start and Stop Data Provisioning Tasks

You can start and stop tasks within the *Data Provisioning Task Monitor*.

Prerequisites

The user must have the following privileges to start or stop tasks:

Table 29: Privileges

Action	Privilege
Start task	sap.hana.im.dp.monitor::StartTask
Stop task	sap.hana.im.dp.monitor::StopTask

Procedure

- To start a task, select a task in the *Task Overview* table.
  a. Click *Start*.
  b. If the object uses table type parameters, select the tables to use when executing the object.
  c. If the object uses variable parameters, specify the values to use when executing the object.

  > **Note**
  >
  > Tasks that belong to realtime design time objects cannot be started or scheduled from the *Data Provisioning Task Monitor*.

- To stop a task, select the running task in the *Task Execution Monitor* table and click *Stop*.

  Note that there might be a delay in stopping the task depending on when the cancellation was initiated and the pending operation.

Related Information

Smart Data Integration: CANCEL TASK [page 140]
Smart Data Integration: START TASK [page 160]

3.4.5 Schedule Data Provisioning Tasks

You can schedule tasks within the *Data Provisioning Task Monitor*.

Prerequisites

The user must have the following roles or privileges to schedule tasks.

Table 30: Roles and Privileges

Action	Role or Privilege
Enable users to schedule task	Role: sap.hana.xs.admin.roles::JobSchedulerAdministrator
Schedule task	• Role: sap.hana.im.dp.monitor.roles::Operations • Application privilege: sap.hana.im.dp.monitor::ScheduleTask

To activate task scheduling, the following must occur:

- Scheduling needs to be enabled via XS Job Admin Dashboard /sap/hana/xs/admin/jobs/ (The user that enables other users to schedule needs the role sap.hana.xs.admin.roles::JobSchedulerAdministrator).
- To schedule tasks, the Job sap.hana.im.dp.monitor.jobs::scheduleTask needs to be enabled in the XS Job Details page: /sap/hana/xs/admin/jobs/#/package/sap.hana.im.dp.monitor.jobs/job/scheduleTask.

Procedure

1. Select the task in the Task Monitor.
2. Click the *Schedules* button.
   The *Schedules* dialog appears.
3. To create a new schedule for the task, click *Add*.
   a. Select the frequency (once or recurring), interval if recurring (year, month, week, day, hour, minute), and the time (local, not server time or UTC) for the task execution.
   b. If the task uses table type parameters, select the tables to use when executing the task.
   c. If the task uses variable parameters, specify the values to use when executing the task.
   d. Click *Schedule*.
      The new schedule is added to the list of schedules for the task.
4. To remove an existing schedule, select the schedule and click *Delete.*
   The schedule is removed from the list of schedules for the task.
5. Close the *Schedules* dialog.

Results

Your task executes as scheduled and you can monitor the results of each execution of the task.

3.5 Managing Enterprise Semantic Services

Use the SAP HANA Enterprise Semantic Services Administration browser-based application to administer and monitor artifacts for semantic services.

To launch the SAP HANA Enterprise Semantic Services Administration tool, enter the following URL in a web browser:

```
http://<<your_HANA_instance:port>>/sap/hana/im/ess/ui
```

The interface includes the following components (tiles):

- *Publication Schedules*
  Publish and unpublish artifacts. Schedule publishing and data profiling requests.
- *Publication Requests*
  View the details and status of all requests.
- *Published Artifacts*
  View and remove artifacts from the knowledge graph.
- *Data Profiling Blacklist*
  Prevent data profiling for selected artifacts.

Once you have drilled into a component, you can click the navigation menu in the upper-left corner to open other components or return to the *Home* page.

Related Information

Roles for Enterprise Semantic Services [page 55]
Enterprise Semantic Services Knowledge Graph and Publication Requests [page 55]
Publishing Artifacts [page 56]
Monitor the Status of Publication Requests [page 59]
Manage Published Artifacts [page 63]
Data Profiling [page 66]
Set Configuration Parameters [page 68]
Troubleshooting Enterprise Semantic Services [page 69]

3.5.1 Roles for Enterprise Semantic Services

SAP HANA role requirements for Enterprise Semantic Services (ESS).

The following database roles are used to control access to Enterprise Semantic Services.

Description	Role
To use the SAP HANA ESS Administration tool	sap.hana.im.ess.roles::Administrator
To update the SAP HANA ESS configuration	sap.hana.im.ess.roles::Configurator
To use the remote view (sap.hana.im.ess.services.views:REMOTE_OBJECTS) or the lineage table functions (sap.hana.im.ess.services.views.datalineage:GET_ALL_IMPACTING_TABLES, sap.hana.im.ess.services.views.datalineage:GET_IMPACTING_TABLES, sap.hana.im.ess.services.views.datalineage:GET_LINEAGE_FROM_VIEW, sap.hana.im.ess.services.views.datalineage:GET_LINEAGE_FROM_SCHEMA)	sap.hana.im.ess.roles::DataSteward
To use the publishing APIs	sap.hana.im.ess.roles::Publisher
To use the search, ctid API, or the remote view (sap.hana.im.ess.services.views:REMOTE_OBJECTS) or the secured data lineage table functions (sap.hana.im.ess.services.views.datalineage:GET_ACCESSIBLE_LINEAGE_FROM_VIEW, sap.hana.im.ess.services.views.datalineage:GET_ACCESSIBLE_LINEAGE)	sap.hana.im.ess.roles::User

3.5.2 Enterprise Semantic Services Knowledge Graph and Publication Requests

How Enterprise Semantic Services enables the searching and profiling of datasets.

Enterprise Semantic Services uses a knowledge graph that describes the semantics of the datasets that are available to users or applications connected to SAP HANA. It is natively stored in the SAP HANA database.

Datasets represented in the knowledge graph can include tables, SQL views, SAP HANA views, remote objects in remote sources, and virtual tables that refer to remote objects.

An Enterprise Semantic Services publication request extracts information from a resource and publishes it in the knowledge graph. When a user searches for an object based on its metadata and contents, the knowledge graph provides the results.

The knowledge graph becomes populated by one or more of the following methods:

- An SAP HANA administrator uses the Enterprise Semantic Services Administration tool to publish datasets
- An SAP HANA administrator configures the Enterprise Semantic Services REST API so that an application can publish datasets

- If an application has already been configured to call the Enterprise Semantic Services REST API, the application can populate the knowledge graph. For example in SAP HANA Agile Data Preparation, when you add a worksheet, the content publishes to the knowledge graph.

Related Information

Managing Enterprise Semantic Services [page 54]
SAP HANA Enterprise Semantic Services JavaScript API Reference
SAP HANA Enterprise Semantic Services REST API Reference

3.5.3 Publishing Artifacts

The SAP Enterprise Semantic Services (ESS) Administration tool lets you publish (or unpublish) artifacts.

You can publish or unpublish an artifact programmatically using the on-demand Enterprise Semantic Services API. This method is useful for applications that manage the life cycle of their artifacts, that is, applications that create, delete, and update SAP HANA artifacts. The application determines which artifacts to publish, republish, or unpublish to the Enterprise Semantic Service knowledge graph. An example is the SAP Agile Data Preparation application.

Administrators can also use the SAP HANA ESS Administration tool to publish or unpublish artifacts. This is useful for applications that manage the life cycle of SAP HANA artifacts but do not want (or cannot easily) integrate artifact management with Enterprise Semantic Services. An example is the SAP ERP application. In those cases, it is easier to delegate to an administrator the task of determining which artifacts should be published to Enterprise Semantic Services depending on the needs of application (for example, access to a semantic service like search).

The best practice is to separate the artifacts published by applications using the on-demand ESS API from those published by an administrator using the SAP ESS Administration tool. Therefore, the artifacts will belong to different publisher groups, as shown on the SAP ESS Administration tool Published Artifacts tile.

Related Information

Publish Artifacts [page 57]
Information Available in Publication Schedule [page 58]
Remove a Schedule for a Deleted Artifact [page 59]
Manage Published Artifacts [page 63]
Information Available in Published Artifacts [page 64]

3.5.3.1 Publish Artifacts

How to use the SAP HANA Enterprise Semantic Services (ESS) Administration tool to publish artifacts in the knowledge graph.

Procedure

1. Select the *Publication Schedules* tile.
2. In the Published Artifact Browser, expand the nodes and select an artifact.

   Note that in the following browsers, you can search for an object within a node using a *Filter*: Publication Schedules (catalog only), Published Artifacts, Data Profiling Blacklist, Entity Grid Tags.
   1. Right-click the object and select *Filter*. If the *Filter* option does not display for the object, select *Refresh*.
   2. In the *Filter* dialog box, start typing the object name or part of the name. The list filters as you type.
   3. Select *OK* to save the filter on the node. To clear the filter later, right-click the object and select *Remove filter*.

      > **Note**
      >
      > Selecting *Refresh* on an object removes all filters from the object and its children.
3. To include the artifact, select *Include for publication*.

   Children of the artifact inherit the INCLUDED configuration of the parent unless specifically excluded.
4. Configure the publication schedule as follows.
   a. For *Next Scheduled Date*, click in the box to select a date and time to next publish the artifact.

      If you do not enter a date-time, it is set to the current day and time.
   b. Enter a frequency *Period*.
5. Configure *Data profiling options*.
   a. Select *Discover content type* to include content types in the publication; however, this can impact performance and is not suggested for production scenarios.
   b. Select *Extract searchable values* to extract values; however, this can impact performance and is not suggested for production scenarios.
6. Select *Active* to enable the schedule.

   To deactivate a schedule, clear the check box and click *Save*.
7. Select *Save*.

   The schedule displays in the *Schedules* table.
8. To stop (cancel) the publication of an artifact: open Publication Requests monitor, refresh the view, select the *Stop* check box for the artifact(s), and select the *Stop* icon in the upper right corner of the window.
9. Optionally, from the navigation menu, open Publication Requests to confirm the status of the request (for example, *REQUEST_PENDING* or *DONE*).
10. To exclude objects, select the object, select *Exclude for publication*, and select *Save*.

Results

The objects display in the browser tree marked with a solid green plus symbol (for included objects) or a solid red minus symbol (for excluded objects). Inherited objects display an outlined green or red symbol. These markers indicate that a request action has been initiated but is independent of the actual status of the request. To view the status of the request, open the Publication Requests monitor. To view the results of requests, view the Published Artifacts monitor.

3.5.3.2 Information Available in Publication Schedule

Publication Schedules displays a table that describes all of the scheduled publication and data profiling requests.

The search field above the table lets you search for published artifacts.

Column Name	Description	Filterable
Publication Artifact	Fully qualified name of the published artifact	Yes
Publication	The mode selected for the schedule (INCLUDE, EXCLUDE, INHERIT)	Yes
Next Scheduled Date	The timestamp for when the schedule will next execute	No
Period	The selected frequency of the schedule	Yes
Active	Whether or not the schedule has been activated	No
Discover	Whether or not the option for *Discover content type* was selected	No
Extracted Searchable Values	Whether or not the option for *Extract searchable values* was selected	No
Warning	A warning indicates that a scheduled artifact has been deleted. The *Delete* option will be enabled on the schedule.	Yes
Delete	To delete a schedule for an artifact that has been deleted, select the check box and click *Delete*.	No

3.5.3.3 Remove a Schedule for a Deleted Artifact

Remove a publication schedule for an artifact that was deleted.

Context

When an artifact is configured as included for publication, a publication schedule displays in the table of schedules. If the artifact gets deleted, the publication schedule will remain until the crawler cannot detect it. A warning will then display for that schedule.

Procedure

1. Select the *Publication Schedules* tile.
2. For the artifact that displays a warning, the *Delete* option will be enabled on that schedule. Select the check box and click *Delete* above the column.

3.5.4 Monitor the Status of Publication Requests

Use *Publication Requests* to view and monitor publishing and profiling requests.

Context

To monitor the status of all requests, Select the *Publication Requests* tile.

A user can only search for catalog or remote objects that are described in the knowledge graph as a result of successful publication requests. However, if the name of an artifact unexpectedly does not appear in the search results, the publication of the corresponding artifact might have failed.

Related Information

Information Available on Publication Requests [page 60]

3.5.4.1 Information Available on Publication Requests

Enterprise Semantic Services Publication Requests displays the status and any error messages for each request.

The search field above the table lets you search for published artifacts.

Column Name	Description	Filterable
Detail	Click the magnifying glass icon to drill into the request for more details. The Detail page displays the following information and statistics: • Request type (see list below) • Submission timestamp • Request status (see list below) • Publisher (user) • Publisher Group • Number of basic artifacts to publish • Number of basic artifacts to unpublish • Number of basic artifacts to profile For the latter three statistics, you can see the associated number of requests that are *Successful*, *Failed*, *In progress*, or *Not started*. The table displays each *Publication Artifact* and its *Publication Status*, *Data Profiling Status*, *Error Code*, and *Error Message* if any. Requests with a FAILED status include a *Retry* check box. To retry the request, select the check box and click *Retry* at the top of the window. To retry all FAILED requests, select the *Retry* check box in the column heading.	No
ID	A number that helps identify a request in the list of requests. Note that this number might not be unique in some cases.	Yes
Request Date	Timestamp of when the request was executed	No
Publication Artifact	Fully qualified name of catalog object or repository object that was published	Yes
Artifact Type	The type of artifact to publish: • SAP HANA views in the repository: attributeview, analyticview, calculationview • SAP HANA catalog objects: table, view, columnview, virtualtable	Yes
Publisher Group	Indicates whether a publication was scheduled using the SAP HANA ESS Administration tool or using the REST API. In the former case, the predefined publisher group is sap.hana.im.ess.AdminPublisherGroup. In the latter case, a call to the publish() API must specify a publisherGroup parameter that defines the ownership of the specified publication in the knowledge graph.	Yes

Column Name	Description	Filterable
Publisher	Name of the SAP HANA user who submitted the request	Yes
Request Type	Request types on the Publication Requests monitor home page include: ON_DEMAND_PUBLISH ON_DEMAND_UNPUBLISH SCHEDULED_PUBLISH MONITORING_UNPUBLISH RETRY_ON_DEMAND_PUBLISH RETRY_ON_DEMAND_UNPUBLISH RETRY_SCHEDULED_PUBLISH Request types on the artifact Detail page include following: PUBLISH_NOT_STARTED UNPUBLISH_NOT_STARTED UNPUBLISH_IN_PROGRESS PUBLISH_DONE PUBLISH_FAILED UNPUBLISH_FAILED	Yes

Column Name	Description	Filterable
Status	Status values on the Publication Requests monitor home page include: REQUEST_PENDING IN_PROGRESS DONE DONE_WITH_ERRORS NOTHING_DONE STOPPING STOPPED Status values on the artifact Detail page include: PROFILING_NOT_STARTED PROFILING_IN_PROGRESS PROFILING_DONE PROFILING_FAILED INACTIVATED NOT PROFILABLE BLACKLISTED OBSOLETE PUBLICATION_FAILED STOPPED	Yes
Error Code	Error code when the status is FAILED. Each range of numbers indicates a specific area as follows: • 100-199: SAP HANA adapter errors • 200-399: Prepare, extract, load, and deploy jobs • 700-799: Miscellaneous	Yes
Error Message	Description of the error	No
Retry	To retry one or more requests, select the *Retry* check box for each, or select the *Retry* check box in the column heading to select all failed requests, and select the *Retry* button.	No

Column Name	Description	Filterable
Stop	To display requests that are currently in progress, select the Refresh icon to update the *Status* column. To move the latest publication requests to the top of the list, for *Request Date* select *Sort Descending*. To stop (cancel) one or more in-progress requests, select the *Stop* check box for each, or select the *Stop* check box in the column heading to select all requests, and select the *Stop* button.	No

3.5.5 Manage Published Artifacts

Use Published Artifacts to view the artifacts that have been published to the knowledge graph and remove (unpublish) them.

Context

The knowledge graph describes the semantics of published artifacts (datasets). Metadata crawlers and data profiling requests let you publish artifacts to the knowledge graph. Thereby, applications can search for and locate these objects and their metadata.

There are two ways to publish artifacts to the knowledge graph: The HTTP REST API `publish()` method and the SAP HANA Enterprise Semantic Services Administration tool. If the same artifact gets published by both mechanisms, the artifact is identified in the Published Artifacts monitor as belonging to a corresponding publisher group. Therefore, publisher groups define ownership of specific publications in the knowledge graph.

When an artifact is published with a specific publisher group, it can only be unpublished by that group. If the same artifact has been published with multiple publisher groups, it can only unpublished when all corresponding publisher groups unpublish it. This control helps avoid conflicts between applications and an administrator using the Administration tool. Otherwise, an application could publish an artifact and another application or administrator could unpublish it.

In the case of the HTTP `publish()` API, the publisher group name is specific to the application; for example for SAP HANA Agile Data Preparation, it could be com.sap.hana.im.adp. For the SAP HANA ESS Administration tool, the predefined publisher group name is sap.hana.im.ess.AdminPublisherGroup.

To limit the size of both extracted metadata elements and extracted searchable attribute values in knowledge graph, you can also select artifacts to unpublish.

Procedure

1. Select the *Published Artifacts* tile.
2. Expand the nodes on the Published Artifact Browser to find the object to view and select it.

Note that in the following browsers, you can search for an object within a node using a *Filter*: Publication Schedules (catalog only), Published Artifacts, Data Profiling Blacklist, Entity Grid Tags.

1. Right-click the object and select *Filter*. If the *Filter* option does not display for the object, select *Refresh*.
2. In the *Filter* dialog box, start typing the object name or part of the name. The list filters as you type.
3. Select *OK* to save the filter on the node. To clear the filter later, right-click the object and select *Remove filter*.

> **Note**
>
> Selecting *Refresh* on an object removes all filters from the object and its children.

3. Select the *Publisher Group* as necessary.
4. The table displays all of the published artifacts, when they were last refreshed, and the number of metadata elements in each.
5. To remove an artifact and its data profiling information, select its *Unpublish* check box and click *Save*. To unpublish all displayed artifacts, select the *Unpublish* check box in the column heading and click *Save*.

   View the *Publication Requests* monitor to confirm that the object was removed. For example, the Request Type would indicate MONITORING_UNPUBLISH.

Related Information

Information Available in Published Artifacts [page 64]

Information Available on Publication Requests [page 60]

3.5.5.1 Information Available in Published Artifacts

Enterprise Semantic Services Published Artifacts displays artifacts that have been published and also lets you remove (unpublish) artifacts from the knowledge graph.

The Published Artifact Browser displays all the published objects available in the Catalog, Content, and Remote Sources folders. The size of an artifact is measured as the total number of searchable metadata elements and searchable attribute values extracted from that artifact.

The search field above the table lets you search for published artifacts.

Name	Description	Filterable
Publisher Group	Indicates whether a publication was scheduled using the SAP HANA ESS Administration tool or using the REST HTTP `publish()` API. When an artifact is published with a specific publisher group, the artifact can only be unpublished by the same group. If a same artifact has been published with different publisher groups, the artifact will be unpublished when all associated groups have unpublished it. For the SAP HANA ESS Administration tool, the predefined publisher group name is sap.hana.im.ess.AdminPublisherGroup. For the REST HTTP `publish()` API, a call to the `publish()` API must specify a `publisherGroup` parameter that determines the name of the publisher group.	
Publication Artifact name	Name of the selected artifact.	
Number of published artifacts	Number of basic artifacts recursively contained in the selected artifact when the selected artifact is a container. If the selected artifact is a basic artifact, the number of published artifacts is equal to 1.	
Number of metadata elements	Total number of extracted metadata elements in the selected artifact.	
Number of extracted values	Total number of attribute values extracted in the selected artifact.	
Publication Artifact	Qualified name of an artifact that was published or contains published artifacts. The fully qualified name is described by three attributes: • Origin: Catalog or Content • Container: Schema name, package path, or virtual container name • Artifact: Name of the artifact (basic or container) in its parent container	Yes Wildcards
Oldest Refresh	Oldest date of updated basic artifacts in the corresponding container. This date is NULL in the case of a basic artifact.	No
Last Refresh	Most recent date of updated basic artifacts in the corresponding container. This date is the last update in the case of a basic artifact.	No
Basic Artifacts	Number of published basic artifacts recursively contained in the corresponding container. This value is 1 in the case of a basic artifact.	Yes
Removable Metadata	Number of non-shared metadata elements. It indicates the number of searchable metadata elements extracted from the corresponding published artifacts that are not shared with other published artifacts. This number gives an indication of how many metadata elements would be removed if you unpublished the artifact.	Yes

Name	Description	Filterable
Removable Values	Number of searchable attribute values extracted for the catalog object represented by the published artifact. It indicates the number of metadata elements that are not shared with other published artifacts. This number gives an indication of how many profiled values would be removed in the case of unpublishing.	Yes
Unpublish	To unpublish the artifact, select the *Unpublish* check box and click *Save*. To unpublish all displayed artifacts, select the *Unpublish* check box in the column heading and click *Save*.	No

3.5.6 Data Profiling

Enterprise Semantic Services can profile the contents of artifacts that have been published to the knowledge graph.

Data profiling is a process that analyzes the values contained in specific columns of a dataset (the columns to analyze are specified internally using ESS logic). Analysis of the data in a column discovers business types, and searchable values can then be extracted and indexed using SAP HANA full text index.

Enterprise Semantic Services can profile the contents of the following artifacts:

- SQL tables
- Column views issued from graphical Calculation views, Attribute, and Analytic views
- Virtual tables created from remote objects of a remote source with the PASSWORD credential type (see the topic "CREATE REMOTE SOURCE" in the *SAP HANA SQL and System Views Reference*).

> **Note**
>
> When requesting profiling of a catalog object that does not result from an activation, you must assign the role SELECT with grant option to the technical user _HANA_IM_ESS. (For activated objects, there is nothing to do.)

Current limitations include the following:

- Calculation views with dynamic privileges will not be profiled
- Views with required parameters will not be profiled
- Calculation views with dependencies to other views will only be profiled if they are referenced by exactly the same set of analytic privileges as their dependent views. It is advised in this version to only create a single analytic privilege that references all views. Future versions will handle dependencies with different privileges.

Related Information

Limiting Objects to Profile [page 67]
Information Available on the Data Profiling Blacklist [page 67]

3.5.6.1 Limiting Objects to Profile

You can prevent artifacts (limited to catalog objects) from being profiled.

Context

Limiting the artifacts to profile lets you control the volume of searchable attribute values or avoid extracting searchable values from datasets that hold sensitive or personal data.

In order to prevent an artifact from being profiled, an administrator can blacklist artifacts. When a catalog object that was previously profiled is blacklisted, all its extracted searchable attribute values are immediately removed from knowledge graph. The catalog object will never be profiled again, even if there is still a data profiling schedule associated with the object(s).

To blacklist an artifact, follow these steps:

Procedure

1. Select the *Data Profiling Blacklist* tile.
2. In the *Catalog Object Browser*, expand the nodes and select an artifact to blacklist. You can also select a schema to list all its children then select objects to blacklist from within it.

   Note that in the following browsers, you can search for an object within a node using a *Filter*: Publication Schedules (catalog only), Published Artifacts, Data Profiling Blacklist, Entity Grid Tags.
   1. Right-click the object and select *Filter*. If the *Filter* option does not display for the object, select *Refresh*.
   2. In the *Filter* dialog box, start typing the object name or part of the name. The list filters as you type.
   3. Select *OK* to save the filter on the node. To clear the filter later, right-click the object and select *Remove filter*.

      > **Note**
      >
      > Selecting *Refresh* on an object removes all filters from the object and its children.
3. To blacklist the artifact, select the *Blacklisted* check box and click *Save*. To blacklist all displayed artifacts, select the *Blacklisted* check box in the column heading and click *Save*.

   To re-enable data profiling for an artifact, clear the check box and click *Save*.

3.5.6.2 Information Available on the Data Profiling Blacklist

The Enterprise Semantic Services Data Profiling Blacklist lets you view and choose which artifacts to blacklist (remove data profiling values).

The search field above the table lets you search for published artifacts.

Name	Description	Filterable
Catalog Object	Fully qualified name of the selected object	
Blacklisted catalog objects	Number of blacklisted objects in the selected artifact	
Extracted values	The total number of extracted values for the selected object	
Schema Name	Name of the schema to which the artifact belongs	Yes Wildcards
Catalog Object	Catalog object name in the schema	Yes
Extracted Values	Number of extracted searchable values for the object	Yes
Blacklisting Date	Timestamp of when the object was blacklisted	No
Blacklisted	Select the check box to blacklist the object and click *Save*. Clear the check box to enable data profiling for the object and click *Save*.	No

3.5.7 Set Configuration Parameters

As an Enterprise Semantic Services administrator, you can set configuration parameter values such as maximum sizes of persistent queues, rolling policy of persistent queues, and so on.

To set configuration parameters, in the SAP HANA studio Administration Console, a system administrator sets values in the reserved table sap.hana.im.ess.eg.configuration::CONFIGURATION. To do so, specific database procedures and user privileges are required.

Example

To set the parameters for "'MAX_ESS_PROFILING_JOBS_SCHEDULE_TIME":

1. Re-enable _HANA_IM_ESS technical user: `ALTER USER _HANA_IM_ESS ENABLE PASSWORD LIFETIME;`
2. Connect with the _HANA_IM_ESS user with its password (the one used during installation).
3. Execute the procedure to increase configuration parameter MAX_ESS_PROFILING_JOBS_SCHEDULE_TIME:

   ```
   CALL
   "SAP_HANA_IM_ESS"."sap.hana.im.ess.eg.configuration::SET_CONFIGURATION_VALUE"('
   MAX_ESS_PROFILING_JOBS_SCHEDULE_TIME', value)
   ```

   where VALUE can be up to 1150000 (the default is 500000 ms).
4. When finished, disable the password: `ALTER USER _HANA_IM_ESS DISABLE PASSWORD LIFETIME;`

3.5.8 Troubleshooting Enterprise Semantic Services

Troubleshooting solutions, tips, and API error messages for Enterprise Semantic Services.

3.5.8.1 Troubleshoot Installation Issues

Troubleshoot Enterprise Semantic Services (ESS) installation issues.

Symptom	When importing the HANA_IM_ESS delivery unit (DU), an activation error occurs and displays in the SAP HANA studio job log view.
Solution	Check the job log details in SAP HANA studio. If the error message is not meaningful, then: • Access the diagnosis file for the index server. • From the bottom of the trace log, look for the first `Check results` message, which should indicate the root cause of the activation failure and suggest how to solve it.

Symptom	The ESS DU has been uninstalled using the uninstallation procedure. When you reimport the ESS DU, activation errors occur, showing that dependent objects are not found.
Cause	Activated objects may have dependent objects that do not yet exist and therefore cause an error.

Solution	Check the following:
	• Verify that all ESS DUs (including the DEMO DU) have been properly uninstalled through the SAP HANA Application Lifecycle Management console.
	• Verify that all related packages have been deleted (those with a naming of sap.hana.im.ess...); otherwise remove them as follows:
	  ○ Create a workspace in the *Repositories* tab of SAP HANA studio.
	  ○ Remove the packages from there.

3.5.8.2 Troubleshoot Publishing Issues

Troubleshoot Enterprise Semantic Services (ESS) publishing issues.

Symptom	The Publication Requests monitor displays a message that includes the phrase `If the failure repeats, contact SAP support.`
Cause	Transaction serialization failures in concurrency scenarios.
Solution	If the transaction that failed was a publish request, on the Publication Requests monitor for the artifact in question, select the *Retry* check box and the *Retry* button.

Symptom	Publishing request cannot be submitted to ESS. An internal server error message is returned to the application.
Cause	Investigate the error as follows: • In the SAP HANA Administration Console, set the trace level for xsa:sap.hana.im.ess to INFO. See Activate Error Trace for Enterprise Semantic Services [page 77]. • Inspect latest diagnosis file xsengine_alert_xxx.trc. • Check for the following error message: `Error: import: package access failed due to missing authorization (…)` This means that the user who is publishing has not been granted the activated ESS role sap.hana.im.ess.roles::Publisher or the application privilege sap.hana.im.ess::Publisher.
Solution	Grant the Publisher role to the user who is publishing. Activated roles are granted through the execution of the procedures GRANT_ACTIVATED_ROLE and GRANT_ APPLICATION_PRIVILEGE. Therefore, to be able to grant a role, a user must have the SQL object privilege EXECUTE on these procedures.

Symptom	Publishing requests appear as not processed in the SAP HANA ESS Administration tool's Publication Schedules view. Request Status remains REQUEST PENDING or REQUESTED.
Solution	Rerun the installation script install.html to verify the installation..

Symptom	A publishing request failed (the Request Status is FAILED in the SAP HANA ESS Administration tool Publication Schedules view).
Cause 1	SAP HANA view definition format is not supported.
Solution 1	The user can “upgrade” the format of the view by editing (make a small change such as adding a space) and saving it.
Cause 2	The SAP HANA view is not supported.
Solution 2	No user action.
Cause 3	API error
Solution 3	Invalid arguments have been passed to the API.

Related Information

API Error Messages [page 73]

3.5.8.3 Troubleshoot Search Issues

Troubleshoot Enterprise Semantic Services (ESS) search issues.

Symptom	The SAP HANA user 'User' cannot perform a search using the ESS API. An internal server error message is returned to the application.
Cause	Investigate the error as follows: • In the SAP HANA Administration Console, set the trace level for xsa:sap.hana.im.ess to INFO. See Activate Error Trace for Enterprise Semantic Services [page 77]. • Inspect latest diagnosis file xsengine_alert_xxx.trc. • Check for the following error message: `Error: import: package access failed due to missing authorization (…)` This means that the 'User' who is publishing has not been granted the privilege Role: sap.hana.im.ess.roles::User.
Solution	Grant the following privilege using SQL commands: `CALL "_SYS_REPO"."GRANT_ACTIVATED_ROLE"('sap.hana.im.ess.roles::User','user');`

Symptom	A search query does not return an expected catalog object that exists in the SAP HANA instance. OR Suggestions do not show an expected term, although that term is associated with a database object in the SAP HANA instance.
Cause	Insufficient privileges.
Solution	Verify the user who is posing the search query has sufficient privileges to access the searchable elements of the expected catalog object as in the following table.

Database object type	Searchable elements	Privileges needed
table, SQL view, virtual table, column view	metadata	Owner, object privilege, READ CATALOG, DATA ADMIN
table, SQL view, virtual table	Profiled data	Owner, SELECT object privilege
column view	Profiled data	Owner, SELECT object privilege, analytic privilege

Symptom	A search query does not return an expected database object that exists in the SAP HANA instance.
Cause	Assuming that the user has sufficient required authorizations, check the syntax of the search query.
Solution	See the following examples of common errors.

Search query	Unmatched searchable element	Correction
Customer name	Does not match “NAME1”	Name* Or follow suggested words
Sales ATT	Does not match value “AT&T”	“AT T” will match AT&T, AT-T, AT/T

Search query	Unexpected match	Correction
Sales_2006	Sales Sales_2007	“sales_2006”
Unit sales	Product unit	“unit sales”
Open insurance contract	Closed insurance contract	+open insurance contract “open insurance contract”

Symptom	Acronyms or abbreviations are not matched by a search query, which, as a result, does not return an expected database object that exists in the SAP HANA instance.
Cause	A configuration is missing in the term-mapping table: SAP_HANA_IM_ESS.”sap.hana.im.ess.services.search::Mapping”
Solution	To modify the entries in the term mapping table, see "Search Term Mapping" in the *Modeling Guide for SAP HANA Smart Data Integration and SAP HANA Smart Data Quality*.

Symptom	A search query does not return an expected database object that exists in the SAP HANA instance.
Cause	Assuming that the user has sufficient authorizations and the syntax of the search query is correct, then use the SAP HANA ESS Administration tool Publish and Unpublish Monitor view to verify whether the database object has been successfully published to ESS, that is, its Request Status is SUCCESSFUL.
Solution	If the Request Status is FAILED, take the appropriate actions according to the error code and message.

3.5.8.4 Troubleshoot Data Profiling Issues

Troubleshoot Enterprise Semantic Services (ESS) data profiling issues.

Symptom	Publishing requests have been processed and all ESS background jobs are active, but data profiling requests appear as not processed in the SAP HANA ESS Administration tool Data Profiling Monitor view. Profiling Status remains as REQUEST PENDING.
Cause	Investigate the error as follows: • Enable the trace level for xsa:sap.hana.im.ess to ERROR in the SAP HANA Administration Console • Inspect the latest diagnosis file `xsengine_alert_xxx.trc`. • Check for the following error message: `ESS815=Profiling Internal Error: Script server must be enabled to profile data` Verify whether the script server was created during installation. To do so, in the SAP HANA Administration Console, view the *Configuration* tab, and in daemon.ini, expand *scriptserver*.
Solution	See .

Symptom	A run-time object has the Request Status of FAILED in SAP HANA ESS Administration tool Data Profiling Monitor view. An error with message code `ESS805 "Insufficient privilege - user xxx must have SELECT with GRANT option on xxx"` is returned.

Cause	If the run-time object is not an activated object, then check that the SELECT right on the run-time object has been granted WITH GRANT OPTION to the technical user _HANA_IM_ESS.
Solution	Run the following SQL command in SAP HANA studio: `GRANT SELECT ON <catalog object> TO _HANA_IM_ESS WITH GRANT OPTION`

Symptom	A data profiling request failed (the Request Status is FAILED in the SAP HANA ESS Administration tool Data Profiling Monitor view).
Cause 1	Insufficient privileges to _HANA_IM_ESS.
Solution 1	Rerun the install.html script.
Cause 2	API error
Solution 2	Invalid arguments have been passed to the API. See API Error Messages [page 73].

3.5.8.5 API Error Messages

Troubleshoot Enterprise Semantic Services (ESS) API errors.

Error	API	Action
ESS100=Error occurred when extracting metadata from a publication artifact.	Publish	Not necessarily an API error. First check trace log file for details on the error.
ESS153=Metadata Extraction Internal Error: No view definition.	Publish	Check API artifact argument or SAP HANA view was concurrently deleted.
ESS154=Metadata Extraction Internal Error: No package name.	Publish	Check API artifact argument or package was concurrently deleted.
ESS158=Package name '{0}' does not exist.	Publish	Check API artifact argument or artifact was concurrently deleted.
ESS159=HANA view '{0}/{1}.{2}' does not exist.	Publish	Check API artifact argument or artifact was concurrently deleted.
ESS160=Publication artifact of type '{0}' is not supported.	Publish	Check API arguments and list of supported types of publication artifacts.
ESS161=Schema name '{0}' does not exist.	Publish	Check API artifact argument or schema was concurrently deleted.

Error	API	Action
ESS162=Catalog object XXX does not exist or is not supported.	Publish	Check API artifact argument or catalog object was concurrently deleted.
ESS163=Invalid publication artifact qualified name '{0}'.	Publish	Not necessarily an API error. Verify the artifact exists.
ESS164=Invalid container path '{0}'. Errors 180 – 186	Publish	Check API artifact argument or updates happened concurrently. Verify the path still exists.
ESS180=Expecting character '{0}' but encountered character '{1}' in the publication artifact.	Publish	Parsing error in artifact name. Check API artifact argument.
ESS181=Close quote without a matching open quote in string '{0}' of the publication artifact.	Publish	Parsing error in artifact name. Check API artifact argument.
ESS182=Invalid catalog publication artifact ['{0}'].	Publish	Parsing error in artifact name. Check API artifact argument.
ESS183=Invalid publication artifact '{0}' ['{1}'].	Publish	Parsing error in artifact name. Check API artifact argument.
ESS184=First identification element of the publication artifact is not a catalog or content '{0}'.	Publish	Parsing error in artifact name. Check API artifact argument.
ESS185=Unmatched quote for identification element {0} in publication artifact.	Publish	Parsing error in artifact name. Check API artifact argument.
ESS186=Identification element {0} should not be empty.	Publish	Parsing error in artifact name. Check API artifact argument.
Error containing string: "Search query syntax error" Errors 500 – 526	Search	Search syntax error. First check explanation given in error message. Then check search query in trace log with trace level DEBUG.
Error containing string: "Request error message" Errors 600 – 645	Search	Request message error. First check explanation given in error message. Then check request message in trace log with trace level DEBUG.
ESS705=Invalid scope name [{0}].	Publish	Only roman letters (both lowercase and uppercase), digits, and underscore characters are valid.

Error	API	Action
ESS715=Invalid type filter [{0}] for container [{1}]. Please use one of [table,virtualtable,view, columnview]	Publish	Invalid argument for typeFilter in API. Check explanation given in error message in trace log.
ESS720=External source [{0}] not supported.	Publish	Check API parameter “source”. Can only be LOCAL.
ESS725=Mandatory argument [{0}] is missing in function [{1}].	Publish CT on-demand Search	Mandatory API argument is missing. First check explanation given in error message. Then check search query in trace log with trace level DEBUG.
ESS730=Invalid value [{0}] for argument [{1}] in function [{2}], expecting: [{3}].	Publish CT on-demand Search	Invalid argument in API. First check explanation given in error message. Then check search query in trace log with trace level DEBUG.
ESS735=Invalid value [{0}] at position [{1}] for argument [{2}] in function [{3}], expecting: [{4}].	Publish CT on-demand Search	Invalid argument in API. First check explanation given in error message. Then check search query in trace log with trace level DEBUG.
ESS808=Profiling Error: Invalid Runtime Object Type {0}. Only the following input values are supported: {1}	CT on-demand	Request to profile an unknown type of run-time object. First check explanation given in error message. Then check search query in trace log with trace level DEBUG.
ESS809=Profiling Error: Column View “{0}”.”{1}” cannot be profiled	CT on-demand	Non-profilable column view due to current restrictions.
ESS810=Profiling Error: {0} “{1}”.”{2}” does not exist. Runtime object does not exist	CT on-demand	Check API argument or artifact was concurrently deleted.
ESS811=Profiling Error: At least one attribute to profile must be given	CT on-demand	Check API argument.
ESS812=Profiling Error: “{0}” is not attribute of {1} “{2}”.”{3}”	CT on-demand	Check API argument or artifact has been updated concurrently.

3.5.8.6 Troubleshooting Tips

Tips for troubleshooting and preparing disgnostic information for SAP support.

Related Information

Troubleshoot Repeated Errors [page 76]
Activate Error Trace for Enterprise Semantic Services [page 77]
Open the Error Log [page 78]
Collect Diagnostic Information [page 78]

3.5.8.6.1 Troubleshoot Repeated Errors

Procedure for error messages in HANA ESS Administration Monitor view that include `If the failure repeats`.

Prerequisites

The error trace for Enterprise Semantic Services must be activated before you retry the operation.

Context

To retry the failed operation:

Procedure

1. Go to the Enterprise Semantic Services Administration Monitor by entering the following URL in a web browser:

   `http://<<your_HANA_instance:port>>/sap/hana/im/ess/ui`
2. If the transaction that failed was a publish request, click the Publish and Unpublish Monitor tile, find your run-time object in the *Published Artifact* column, and click the *Retry* icon.

Next Steps

After you retry the operation and it still fails, collect diagnostic information.

3.5.8.6.2 Activate Error Trace for Enterprise Semantic Services

The error trace obtains detailed information about actions in Enterprise Semantic Services.

Prerequisites

To configure traces, you must have the system privilege TRACE ADMIN.

Context

To activate the error trace for Enterprise Semantic Services, follow these steps:

Procedure

1. Log on to SAP HANA studio with a user that has system privilege TRACE ADMIN.
2. In the Administration editor, choose the *Trace Configuration* tab.
3. Choose the *Edit Configuration* button for the trace that you want to configure.
4. Expand the *XS ENGINE* node.

   > **Note**
   >
   > Ensure you select the checkbox *Show All Components*.

5. Locate `xsa:sap.hana.im.ess` and check that *system trace level* is *INFO*, *ERROR*, or *DEBUG*, because it is usually set to *DEFAULT*.
6. Click *Finish*.

3.5.8.6.3 Open the Error Log

After the Enterprise Semantic Services error trace is activated, find the error message.

Procedure

1. Log on to SAP HANA studio with a user name that has system privilege CATALOG READ.
2. In the Administration editor, choose the *Trace Configuration* tab and go to *Diagnosis Files* tab.
3. Look for one of the two most recent files `xsengine_alert_xxx.trc` or `indexserver_alert_xxx.trc`..
4. Go to the end of the file to see the error.

3.5.8.6.4 Collect Diagnostic Information

Procedure to obtain diagnostic information to send to SAP support.

Procedure

1. Log on to SAP HANA studio with a user name that has the system privilege CATALOG READ.
2. In the Administration editor, choose the *Trace Configuration* tab and go to *Diagnosis Files* tab.
3. Choose *Diagnosis Information* > *Collect*.
4. Send the following information to SAP support:
   - Error message and error code
   - Collected diagnosis files
   - If the error appeared in the Publication Schedules monitor, send all other fields of the displayed row with the error message.

4 Maintaining Connected Systems

Avoid errors and minimize downtime by accounting for SAP HANA smart data integration components when planning maintenance tasks for connected systems such as source databases and the SAP HANA system.

4.1 Maintaining the SAP HANA System

Consider any effects to your data provisioning landscape before performing common SAP HANA maintenance tasks.

Most commonly, you may need to suspend and resume any data provisioning remote sources prior to performing SAP HANA maintenance operations.

4.1.1 Update the SAP HANA System

Suspend and resume remote sources when you need to system the SAP HANA system.

Prerequisites

Be sure to back up the SAP HANA system before starting the upgrade process.

Procedure

1. Suspend all remote sources.
2. Update the SAP HANA system.
3. If needed, update each Data Provisioning Agent in your landscape.
4. Resume all remote sources.

Related Information

Suspend and Resume Remote Sources [page 44]

4.1.2 Takeover/Failback with SAP HANA System Replication

Suspend and resume remote sources when you need to perform a takeover and failback operation with SAP HANA system replication.

Prerequisites

If the SAP HANA system is used as a data source, both the primary and secondary systems must be configured with a virtual IP.

Procedure

1. Suspend all remote sources.
2. Perform a takeover on the secondary SAP HANA system.
3. Perform a failback on the former primary SAP HANA system.
4. Resume all remote sources.

Next Steps

Following an unplanned failback operation, monitor remote subscription exceptions in the *Data Provisioning Remote Subscription Monitor*. Click *Retry Operation* to clear any exceptions.

4.1.3 Failover with SAP HANA Scale-Out

Suspend and resume remote sources when you need to perform a host auto-failover with SAP HANA scale-out.

Prerequisites

If the SAP HANA system is used as a data source, both the active and standby hosts must be configured with a virtual IP.

Procedure

1. Suspend all remote sources.
2. Stop the active SAP HANA host.
3. Wait for the standby host to take over operations from the primary host.
4. Resume all remote sources.

Next Steps

Following an unplanned failover operation, monitor remote subscription exceptions in the *Data Provisioning Remote Subscription Monitor*. Click *Retry Operation* to clear any exceptions.

Related Information

Suspend and Resume Remote Sources [page 44]

4.2 Maintaining Source Databases

Keep your databases performing at maximum capacity by learning about cleaning log files, recovery, and preparing for restarts.

Restart the Source Database [page 82]
When you need to restart a source database, stop any remote source capture and restart the capture after restarting the database.

Change the Source Database User Password [page 83]

When you change the password for the source database user, you must update any remote sources that access the database with that user.

Cleaning LogReader Archives Safely [page 83]
: Avoid disk space issues and ensure smooth real-time replication by regularly cleaning up the LogReader archives.

Recover from Missing LogReader Archives [page 86]
: When archive logs are missing, replication will fail. There are multiple solutions for recovering and restarting replication.

Change the Primary Archive Log Path During Replication [page 87]
: Replication is not impacted when the primary archive log path is changed during replication.

Maintain the Source Database without Propagating Changes to SAP HANA [page 87]
: Use the Maintenance User Filter to define a source database user that can perform maintenance tasks in a source database without having the changes propagated to the SAP HANA system through data provisioning adapters.

Recover with MS SQL Always On Failover [page 88]
: Re-execute a replication task when MS SQL fails over during the initial load.

Recover with SAP HANA System Replication Failover [page 89]
: Re-execute a replication task when SAP HANA fails over during the initial load.

4.2.1 Restart the Source Database

When you need to restart a source database, stop any remote source capture and restart the capture after restarting the database.

Procedure

1. Suspend capture on any remote sources that access the database.
2. Restart the source database.
3. Resume capture on any remote sources that access the database.

Related Information

Suspend and Resume Remote Sources [page 44]

4.2.2 Change the Source Database User Password

When you change the password for the source database user, you must update any remote sources that access the database with that user.

Procedure

1. Suspend capture on any remote sources that access the database.
2. In the SAP HANA Web-based Development Workbench, locate the remote source and change the password in the remote source credentials properties.
3. Resume capture on any remote sources that access the database.

Related Information

Suspend and Resume Remote Sources [page 44]
Suspend and Resume Remote Sources [page 44]

4.2.3 Cleaning LogReader Archives Safely

Avoid disk space issues and ensure smooth real-time replication by regularly cleaning up the LogReader archives.

The Oracle, DB2, and MS SQL LogReader adapters rely on the log reader archive log to retrieve change data from the source databases. Over time, the archive log can grow to consume large amounts of disk space if it is not cleaned.

Clean the Archive Log on Oracle [page 84]
Identify the Log Sequence Number (LSN) and use the RMAN utility to clean the log.

Clean the Archive Log on MS SQL [page 85]
Identify the Log Sequence Number (LSN) and truncate the log using a MS SQL command.

Clean the Archive Log on DB2 [page 85]
Identify the Log Sequence Number (LSN) and use the DB2 utility to clean the log.

4.2.3.1 Clean the Archive Log on Oracle

Identify the Log Sequence Number (LSN) and use the RMAN utility to clean the log.

Procedure

1. Identify the ending truncation point of the last commit transaction in the SAP HANA server.

   ```
   SELECT SRC.REMOTE_SOURCE_NAME, MIN(SUB_M.LAST_PROCESSED_BEGIN_SEQUENCE_ID)
   FROM SYS.M_REMOTE_SUBSCRIPTIONS SUB_M INNER JOIN SYS.REMOTE_SUBSCRIPTIONS_ SUB
   ON SUB_M.SUBSCRIPTION_NAME = SUB.SUBSCRIPTION_NAME
   INNER JOIN SYS.REMOTE_SOURCES_ SRC
   ON SUB.REMOTE_SOURCE_OID = SRC.REMOTE_SOURCE_OID
   WHERE LENGTH(LAST_PROCESSED_BEGIN_SEQUENCE_ID) > 0 and SRC.REMOTE_SOURCE_NAME =
   '<remote_source_name>' GROUP BY SRC.REMOTE_SOURCE_NAME
   ```

   The ending truncation point is returned. For example:

   ```
   00000000000000000000000000000000000000000000000000000000000000000000000000001298ee5000
   00001000003940000670f01e0000001298ee20000000000000000
   ```

2. Use the ending truncation point to identify the LSN of the log file in Oracle.

   ```
   SQL>SET SERVEROUTPUT ON FORMAT WRAPPED
   DECLARE LSN NUMBER(38);
   SCN NUMBER(38);
   SEQID VARCHAR2(200):='<truncation_point>';
   BEGIN
   SELECT TO_NUMBER(SUBSTR(SEQID,109,12),'XXXXXXXXXXXXXXX') INTO SCN FROM DUAL;
   SELECT SEQUENCE# INTO LSN FROM (SELECT SEQUENCE# FROM V$ARCHIVED_LOG WHERE
   ARCHIVED='YES' AND DELETED='NO' AND FIRST_CHANGE#<= SCN AND NEXT_CHANGE# > SCN)
   UNION (SELECT SEQUENCE# FROM V$ARCHIVED_LOG WHERE ARCHIVED='YES' AND
   DELETED='NO' AND FIRST_CHANGE#<= SCN AND NEXT_CHANGE# > SCN);
   DBMS_OUTPUT.PUT_LINE('THE LSN IS: ' || TO_CHAR(LSN));
   END;
   /
   ```

   The LSN is returned. For example: `916`

3. Identify the number of the Oracle redo log group.

   ```
   SQL> select count(GROUP#) from V$LOG;
   ```

   The number of the Oracle redo log group is returned. For example: `3`

4. Use the RMAN utility to clean the archive log.

   The log files that can safely be cleaned have a sequence number smaller than the LSN minus the number of the Oracle redo log group.

   For example, with an LSN of `916` and an Oracle redo log group of `3`, logs through sequence number `913` may be safely cleaned.

   ```
   rman target sys/<sys_password>@<SID>
   RMAN> delete archive log until sequence <sequence_number>;
   ```

4.2.3.2 Clean the Archive Log on MS SQL

Identify the Log Sequence Number (LSN) and truncate the log using a MS SQL command.

Procedure

1. Identify the end LSN of the last commit transaction in the SAP HANA server.

   ```
   SELECT SRC.REMOTE_SOURCE_NAME, MIN(SUB_M.LAST_PROCESSED_BEGIN_SEQUENCE_ID)
   FROM SYS.M_REMOTE_SUBSCRIPTIONS SUB_M INNER JOIN SYS.REMOTE_SUBSCRIPTIONS_ SUB
   ON SUB_M.SUBSCRIPTION_NAME = SUB.SUBSCRIPTION_NAME
   INNER JOIN SYS.REMOTE_SOURCES_ SRC
   ON SUB.REMOTE_SOURCE_OID = SRC.REMOTE_SOURCE_OID
   WHERE LENGTH(LAST_PROCESSED_BEGIN_SEQUENCE_ID) > 0 and SRC.REMOTE_SOURCE_NAME =
   '<remote_source_name>' GROUP BY SRC.REMOTE_SOURCE_NAME
   ```

   The ending truncation point is returned. For example:

   ```
   00000000000000000000000000000000000000000000000000000000000000000000000000000032b00000ef
   80004000004010000032b00000ef8000100000ef0000300000000
   ```

2. Use the truncation point to clean the archive log in MS SQL.

   ```
   declare @seq_id_str varchar(200) = '<truncation_point>';
   declare @xact_seqno_str varchar(200) = substring(@seq_id_str, len(@seq_id_str) -
   34*2+1, 10*2)
   declare @xdesid binary(10) = convert(binary(10),'00', 2)
   declare @xact_seqno binary(10) = convert(binary(10),@xact_seqno_str, 2)
   print 'Commit LSN: '
   print @xact_seqno
   exec sp_repldone @xdesid, @xact_seqno, 0,0,0
   exec sp_replflush
   ```

4.2.3.3 Clean the Archive Log on DB2

Identify the Log Sequence Number (LSN) and use the DB2 utility to clean the log.

Procedure

1. Identify the current LSN in the DB2 database.

   ```
   db2 => SELECT SUBSTR(RAVALUE,41,12) FROM
    <pds_username>.RA_XLOG_SYSTEM_  WHERE RAKEY = 'ltm_locator'
   ```

   > **Note**
   >
   > <pds_username> should be the same user specified in the remote source in SAP HANA.

The LSN is returned. For example: `0000004ff413`

2. Identify the database path in DB2.

```
db2 list active databases
```

The database path is returned.

3. Navigate to the DB2 database path, and identify the archive log file containing the current LSN.

```
db2flsn -q <log_sequence_number>
```

The filename of the log containing the current LSN is returned. For example: `S0000354.LOG`

Archive logs older than this file can safely be cleaned.

4. Identify the first archive log method in DB2.

```
db2 "get db cfg for <database_name>" | grep -E "First log archive method"
```

a. If the first archive log method is `LOGRETAIN`, use a DB2 command to delete the old log files.

```
db2=> prune logfile prior to <log_file_containing_LSN>
```

b. If the first archive log method is not `LOGRETAIN`, delete log files older than the identified LSN log manually from the archive log path on the DB2 host.

4.2.4 Recover from Missing LogReader Archives

When archive logs are missing, replication will fail. There are multiple solutions for recovering and restarting replication.

> **Tip**
>
> If a secondary archive log path is set prior to replication, replication with automatically switch to the secondary log path if an archive log is missing from the primary path.
>
> Process any exceptions in the *Data Provisioning Remote Subscription Monitor*.

Procedure

- Restore the missing archive logs from a backup location.

  An exception is raised when an archive log file cannot be accessed during replication.

  a. In the *Data Provisioning Remote Subscription Monitor*, click on the error in the *Remote Source Monitor* or *Remote Subscription Monitor* tables.
  For example:

```
LogReader is in ERROR state [ORA-00308: cannot open archived log '/
rqa16clnx3_work1/oraclearchive/o12lnxrdb/1_2581_896793021.dbf'
                                ORA-27037: unable to obtain file status
                                Linux-x86_64 Error: 2: No such file or
directory
```

```
                                    Additional information: 3
                                    ]. Check LogReader log for details. Context:
null
```

If the error contains the filename of the missing archive log, the log can be restored from a backup. If the error does not contain a missing archive log filename, clean the archive log safely.

b. Restore the missing archive log file from the backup location on the source machine.
For example:

```
cp /rqa16clnx3_work1/oraclearchive/backup/1_2581_896793021.dbf /
rqa16clnx3_work1/oraclearchive/o12lnxrdb
```

c. Suspend and resume remote source capture to recover the replication.
d. In the *Data Provisioning Remote Subscription Monitor*, click on the error in the CDC status column.
e. Select the error for the adapter, and click *Retry Operation*.
f. Select the error for the receiver, and click *Retry Operation*.
g. In the *Data Provisioning Remote Subscription Monitor*, verify that the CDC status no longer indicates an error.

- Recover when the missing archive logs cannot be restored.
  a. Drop all replication tasks with real-time enabled.
  b. Re-create the replication tasks with real-time enabled.

4.2.5 Change the Primary Archive Log Path During Replication

Replication is not impacted when the primary archive log path is changed during replication.

For example, if a new primary log path is set in Oracle:

```
SQL> SELECT DESTINATION FROM V$ARCHIVE_DEST WHERE DEST_NAME='LOG_ARCHIVE_DEST_1';
DESTINATION
--------------------------------------------------------------------------------
D:\oracle12\archive\o12ntpdb1
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=D:\oracle12\archive\o12ntpdb1
_2';
System altered.
SQL> SELECT DESTINATION FROM V$ARCHIVE_DEST WHERE DEST_NAME='LOG_ARCHIVE_DEST_1'
;
DESTINATION
--------------------------------------------------------------------------------
D:\oracle12\archive\o12ntpdb1_2
```

You can use the*Data Provisioning Remote Subscription Monitor* to verify that the replication for all tables has not been impacted.

4.2.6 Maintain the Source Database without Propagating Changes to SAP HANA

Use the Maintenance User Filter to define a source database user that can perform maintenance tasks in a source database without having the changes propagated to the SAP HANA system through data provisioning adapters.

The actions that can be filtered depend on the source database:

- For Oracle and MS SQL LogReader adapters, database transactions including INSERT, UPDATE, and DELETE, and DDL changes such as ALTER TABLE
- For the SAP HANA adapter, database transactions such as INSERT, UPDATE, and DELETE

Prerequisites

Determine the source database user that will perform the maintenance tasks.

Procedure

1. Suspend any remote sources for the associated source database.
2. In the SAP HANA Web-based Development Workbench, choose *Provisioning* > *Remote Sources* and right-click on the remote source.
3. In the *Maintenance User Filter* option, specify the username of the database maintenance user.
4. Re-enter the login credentials for the source database, and save the changes.
5. Resume the remote sources for the source database.

Related Information

Suspend and Resume Remote Sources [page 44]

4.2.7 Recover with MS SQL Always On Failover

Re-execute a replication task when MS SQL fails over during the initial load.

Context

If a MS SQL failover happens during the initial load of a replication task execution, the replication task fails.

For example:

```
Internal error: Remote execution error occurred when getting next row from Result
Set.
```

> **Note**
>
> If the failover occurs during real-time replication, no action is required, and the replication will continue automatically.

Procedure

1. In the *Data Provisioning Design Time Object Monitor* locate the replication task marked as `FAILED`.
2. After the MS SQL failover completes, re-execute the replication task.

4.2.8 Recover with SAP HANA System Replication Failover

Re-execute a replication task when SAP HANA fails over during the initial load.

Context

If an SAP HANA failover happens during the initial load of a replication task execution, the replication task fails.

For example:

```
Internal error: sql processing error.
```

> **Note**
>
> If the failover occurs during real-time replication, see Failover with SAP HANA Scale-Out [page 81].

Procedure

1. In the *Data Provisioning Design Time Object Monitor* locate the replication task marked as `FAILED`.
2. After the SAP HANA failover completes, re-execute the replication task.

5 Troubleshooting and Recovery Operations

This section describes common troubleshooting scenarios for your SAP HANA smart data integration landscape, as well as recovery steps to follow when errors occur.

5.1 Troubleshooting Real-Time Replication Initial Queue Failures

Diagnose and resolve common failure scenarios for initial queues in real-time replication tasks.

Log reader adapters require an instance port that must not be used by any other applications on the Data Provisioning Agent host.

Resolve Data Provisioning Server Timeouts [page 105]
When the default message timeout value for the Data Provisioning Server is too short, real-time replication tasks may fail during the initial load operation.

Load Clustered and Pooled Table Metadata into SAP HANA [page 106]
Real-time replication tasks with clustered or pooled tables may fail when metadata has not been correctly loaded into the SAP HANA database.

5.1.1 Resolve User Privilege Errors

Adapters used for real-time replication require the remote source database user to be configured with privileges specific to the source database type.

Context

Insufficient user privileges are indicated in the `<DPAgent_root>/log/framework.trc` trace log file.

For example:

```
Required privileges and/or roles not granted to the database user [ZMTEST]. Missing
privileges and/or roles are [SELECT ANY TRANSACTION]
```

Resolve the error by granting the missing user privileges.

Procedure

1. Connect to the source database with a DBA administrator user.
2. Grant the missing privileges required for the source database to the user specified in the remote source.
3. Re-create or edit the exisitng replication task.
4. Execute the replication task.

5.1.2 Resolve Remote Source Parameter Errors

A replication task may fail if remote source parameters are specified with invalid or out-of-range values, or if values for any mandatory dependent parameters are not specified.

Context

Remote source parameter errors are indicated in the `<DPAgent_root>/log/framework.trc` log file, and may vary based on the specific remote source parameter.

For example, the following scenarios may cause a remote source parameter error:

- Remote source parameter value is outside the valid range.
  For example, if the Oracle maximum operation queue size is set to a value outside of the valid range, such as 10.

```
[ERROR]
com.sap.hana.dp.oraclelogreaderadapter.OracleLogReaderAdapter.cdcOpen[669]    -
Adapter validation failed.
com.sap.hana.dp.cdcadaptercommons.validation.ValidationException: Failed to
validate properties. Error(s):
The value [10] of property [lr_max_op_queue_size] is not in the range [25,
2147483647].
```

- An Oracle remote source is configured to use TNSNAMES, but the TNSNAMES file and connection parameters are not specified.

```
[ERROR] WorkerThread.processRequest  - Adapter validation failed. Failed to
validate properties. Error(s):
Property [pds_tns_connection] is mandatory.
Property [pds_tns_filename] is mandatory. Context: null
com.sap.hana.dp.adapter.sdk.AdapterException: Adapter validation failed. Failed
to validate properties. Error(s):
Property [pds_tns_connection] is mandatory.
Property [pds_tns_filename] is mandatory.
```

- An Oracle remote source is configured as a Multitenant Database, but the container database service name, pluggable database service name, or Oracle multitenant credentials are not specified.

```
[ERROR] WorkerThread.processRequest  - Adapter validation failed. Failed to
validate properties. Error(s):
Property [cdb_password] is mandatory.
Property [cdb_service_name] is mandatory.
Property [cdb_username] is mandatory.
Property [pds_service_name] is mandatory. Context: null
com.sap.hana.dp.adapter.sdk.AdapterException: Adapter validation failed. Failed
to validate properties. Error(s):
Property [cdb_password] is mandatory.
Property [cdb_service_name] is mandatory.
Property [cdb_username] is mandatory.
Property [pds_service_name] is mandatory.
```

- A user name is specified incorrectly for a case-sensitive adapter.

```
[ERROR] WorkerThread.processRequest  - Adapter validation failed. User Name
[zmtest] does not exist in the source database. Note that 'User Name' option is
case-sensitive. Context: null
```

```
com.sap.hana.dp.adapter.sdk.AdapterException: Adapter validation failed. User
Name [zmtest] does not exist in the source database. Note that 'User Name'
option is case-sensitive.
```

Resolve the error by specifying complete remote source parameter values within the valid range.

Procedure

1. Alter the remote source parameter and specify a value within the valid range or any missing dependent parameter values.
2. Re-create the replication task or edit the existing replication task.
3. Execute the replication task.

Related Information

Alter Remote Source Parameters [page 46]

5.1.3 Resolve Improper Source Database Configuration

Real-time replication tasks may fail when the remote source database has not been properly configured to support real-time replication.

Enable the Oracle Archive Log [page 94]
Real-time replication tasks on Oracle remote sources may fail if the Oracle archive log has not been enabled.

Enable Supplemental Logging on Oracle [page 95]
Real-time replication tasks on Oracle remote sources may fail if supplemental logging on the Oracle source database has not been enabled or does not match the supplemental logging parameter in the remote source.

Enable the Secure File LOB Setting on Oracle [page 96]
Oracle LOB data may not replicate correctly when the DB_SECUREFILE setting is set to “ALWAYS” or “PREFERRED”.

Specify the Oracle Service Name [page 97]
If the Oracle service name is not specified correctly, the remote source can be created but remote tables cannot be browsed.

Configure the Microsoft SQL Server Transaction Log [page 98]
Replication tasks may fail if the Microsoft SQL Server transaction log cannot be read or is full.

Initialize the Microsoft SQL Server Database [page 98]
Replication tasks may fail when the data capture mode is set to “Native Mode” and the Microsoft SQL Server has not been initialized.

Enable the DB2 Archive Log [page 99]

Real-time replication tasks on DB2 remote sources may fail if the DB2 archive log has not been enabled.

Create the Temporary Tablespace on DB2 [page 99]
Real-time replication tasks on DB2 remote sources may fail if the user temporary tablespace has not been created.

Specify a Valid DB2 Port Number [page 100]
Remote sources for DB2 cannot be created if the DB2 port is outside the allowed range of 1-65335.

Verify the DB2 Native Connection Settings [page 100]
The LogReader may fail to initialize properly when the DB2 native connection fails.

Define the SAP ASE Adapter Interface [page 101]
Real-time replication tasks on the SAP ASE adapter may fail if an entry for the adapter has not been added to the interface file of the data server.

5.1.3.1 Enable the Oracle Archive Log

Real-time replication tasks on Oracle remote sources may fail if the Oracle archive log has not been enabled.

Context

A missing archive log may be indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[ERROR] com.sap.hana.dp.cdcadaptercommons.StatefulAdapterCDC
$CDCOpened.addSubscription[791]    - [oracleSrc] Failed to add the first
subscription. SubscriptionSpecification [header=remoteTableId=466,
remoteTriggerId=178688, sql=SELECT "T1"."INT_C1", "T1"."VARCHAR_C2" FROM
"""LR_USER"".""TESTTB1""" "T1" , subscription=, customId=, seqID=SequenceId
[value=[0, 0, 0, 0]], isLastSubscription=true, withSchemaChanges=false,
firstSubscriptionOnTable=true, lastSubscriptionOnTable=true]
com.sap.hana.dp.adapter.sdk.AdapterException: Failed to start Log Reader because of
failure to initialize LogReader. Error:Oracle LogMiner must be installed in order
to use this command. Please verify that LogMiner is installed.
```

An additional error may be indicated in the `<DPAgent_root>/log/repagent.log` log file.

For example:

```
ERROR  com.sybase.ds.oracle.logmnr.LogMiner  Failed to start Log Miner:
STARTSCN=2814017, ENDSCN=2814017
ERROR  com.sybase.ds.oracle.logmnr.LogMiner  Failed to start LogMiner when
verifying LogMiner installation. ORA-01325: archive log mode must be enabled to
build into the logstream
```

Resolve the error by enabling the archive log on the Oracle source database.

Procedure

1. On the Oracle source database, check whether archive logging is enabled.

   ```
   SQL> SELECT LOG_MODE FROM V$DATABASE;
   ```

   If the result for LOG_MODE is not ARCHIVELOG, archive logging is disabled.
2. Enable archive logging.

   ```
   SQL> SHUTDOWN IMMEDIATE;
   SQL> CONNECT SYS/PASSWORD AS SYSDBA;
   SQL> STARTUP MOUNT;
   SQL> ALTER DATABASE ARCHIVELOG;
   SQL> ALTER DATABASE OPEN;
   ```
3. Specify a local archive log directory.

   ```
   SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_1='LOCATION=<local\file\path>';
   ```
4. Verify the archive destination.

   ```
   SQL> SELECT DESTINATION FROM V$ARCHIVE_DEST WHERE DEST_NAME='LOG_ARCHIVE_DEST_1';
   DESTINATION
   ```

5.1.3.2 Enable Supplemental Logging on Oracle

Real-time replication tasks on Oracle remote sources may fail if supplemental logging on the Oracle source database has not been enabled or does not match the supplemental logging parameter in the remote source.

Context

An error may be indicated in the `<DPAgent_root>/log/framework.trc` log file in the following scenarios.

- Minimum database-level supplemental logging is disabled.
  For example:

  ```
  [ERROR]
  com.sap.hana.dp.adapter.framework.core.WorkerThread.processRequest[329]    -
  Adapter validation failed. Minimal supplemental logging not enabled.
  ```
- The remote source parameter is set to “database”, but PRIMARY KEY and UNIQUE KEY database-level supplemental logging is not enabled.
  For example:

  ```
  [ERROR]
  com.sap.hana.dp.adapter.framework.core.WorkerThread.processRequest[329]    -
  Adapter validation failed. Database PRIMARY KEY and/or UNIQUE supplemental
  logging is not enabled.
  ```
- The remote source parameter is set to “table”, but table-level supplemental logging is not enabled.

For example:

```
[ERROR]
com.sap.hana.dp.adapter.framework.core.WorkerThread.processRequest[329]    -
Adapter validation failed. Primary key and/or unique key supplemental logging is
not turned on for these Oracle system tables: [LOBFRAG$, TABPART$, TABSUBPART$,
COLTYPE$, INDSUBPART$, MLOG$, TYPE$, INDCOMPART$, NTAB$, TABCOMPART$, COLLECTION
$, LOB$, PROCEDUREINFO$, SNAP$, OPQTYPE$, DEFERRED_STG$, LOBCOMPPART$, ARGUMENT
$, RECYCLEBIN$, SEQ$, ATTRIBUTE$, INDPART$]
```

Resolve the error by enabling a supplemental logging level that matches the remote source configuration.

Procedure

- Enable minimum database-level supplemental logging.

  ```
  ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
  ```

- Enable PRIMARY KEY and UNIQUE KEY database-level supplemental logging.

  ```
  ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE) COLUMNS;
  ```

- Enable table-level supplemental logging on each table specified in the error message.

  ```
  ALTER TABLE <TABLE_NAME> ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
  ```

  For example:

  ```
  ALTER TABLE SYS.ARGUMENT$ ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
  ALTER TABLE SYS.ARGUMENT$ ADD SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS;
  ALTER TABLE SYS.ATTRIBUTE$ ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
  ALTER TABLE SYS.ATTRIBUTE$ ADD SUPPLEMENTAL LOG DATA (UNIQUE INDEX) COLUMNS;
  ...
  ```

5.1.3.3 Enable the Secure File LOB Setting on Oracle

Oracle LOB data may not replicate correctly when the DB_SECUREFILE setting is set to “ALWAYS” or “PREFERRED”.

Context

The Oracle Log Reader adapter supports Secure File LOB only when the primary database setting DB_SECUREFILE is set to “PERMITTED”.

Procedure

1. Set the Oracle parameter to “PERMITTED”.

   ```
   SQL> alter system set db_securefile = 'PERMITTED';
   ```

2. Verify the DB_SECUREFILE setting.

   ```
   SQL> show parameter db_securefile
   ```

5.1.3.4 Specify the Oracle Service Name

If the Oracle service name is not specified correctly, the remote source can be created but remote tables cannot be browsed.

Context

When the Oracle service name is set as a domain name, the “Database Name” remote source parameter must be specified as the Oracle service name.

For example, when checking the Oracle service name:

```
SQL> show parameter service_names;
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
service_names                        string      <qualified.domain.name>
```

Procedure

1. In the `listener.ora` file, enable the service listener.

   Set the USE_SID_AS_SERVICE_LISTENER parameter to “ON”.

   For example:

   ```
   USE_SID_AS_SERVICE_LISTENER=ON
   LISTENER =
   (DESCRIPTION_LIST =
       (DESCRIPTION =
       (ADDRESS = (PROTOCOL = TCP)(HOST = <agent_hostname>)(PORT = 1521))
       (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
       )
   )
   ```

2. Restart the Oracle listener.
3. Re-create the replication task.

5.1.3.5 Configure the Microsoft SQL Server Transaction Log

Replication tasks may fail if the Microsoft SQL Server transaction log cannot be read or is full.

Context

- An unreadable transaction log error may be indicated in the `<DPAgent_root>/log/framework.trc` log file. For example:

  ```
  ERROR  com.sybase.ds.mssql.log.device.LogDevice  The log file <C:\Program Files
  \Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\ms2014db3_log.ldf> is being
  locked by SQL Server process.
  ```

  Resolve the error by installing the sybfilter driver and making the log file readable.
- If the Microsoft SQL Server transaction log is full, the following error may be reported:

  ```
  com.microsoft.sqlserver.jdbc.SQLServerException: The transaction log for
  database 'BUDGET_LCL_DTM_HANA' is full due to 'ACTIVE_TRANSACTION'.
  ```

  Verify that the auto-extend option for the Microsoft SQL Server log file is enabled, and that the disk where the log file is located has available free space.

5.1.3.6 Initialize the Microsoft SQL Server Database

Replication tasks may fail when the data capture mode is set to “Native Mode” and the Microsoft SQL Server has not been initialized.

Context

The Microsoft SQL Server database must be initialized to ensure that the log reader adapter can open the supplemental log of each table marked for replication.

> **Note**
>
> Each Microsoft SQL Server needs to be initialized only one time.

When the database has not been initialized, an error may be indicated.

For example:

```
Error: (256, 'sql processing error: QUEUE: QATUSER_REPTEST_START: Failed to add
subscription for remote subscription QATUSER_REPTEST_START.Error: exception 151050:
CDC add subscription failed: Failed to add the first subscription. Error: Failed to
start Log Reader because of failure to initialize LogReader. Error: The server is
not initialized. Please run server_admin init first.\n\n: line 1 col 1 (at pos 0)')
```

Resolve the error by configuring the primary data server.

5.1.3.7 Enable the DB2 Archive Log

Real-time replication tasks on DB2 remote sources may fail if the DB2 archive log has not been enabled.

Context

A missing archive log may be indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[ERROR] com.sap.hana.dp.adapter.framework.core.WorkerThread.processRequest[329]
- Failed to add the first subscription. Error: Failed to start Log Reader because
of failure to initialize LogReader. Error: An error occured while getting DB
parameter <LOGARCHMETH1>. Exception: DB2 SQL Error: SQLCODE=-286, SQLSTATE=42727,
SQLERRMC=8192;QARUSER, DRIVER=4.18.60
```

Resolve the error by setting the primary DB2 UDB database transaction logging to archive logging.

5.1.3.8 Create the Temporary Tablespace on DB2

Real-time replication tasks on DB2 remote sources may fail if the user temporary tablespace has not been created.

Context

A temporary tablespace error may be indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[ERROR] com.sap.hana.dp.adapter.framework.core.WorkerThread.processRequest[329]
- Failed to add the first subscription. Error: Failed to start Log Reader because
of failure to initialize LogReader. Error: An error occured while getting DB
parameter <LOGARCHMETH1>. Exception: DB2 SQL Error: SQLCODE=-286, SQLSTATE=42727,
SQLERRMC=8192;QARUSER, DRIVER=4.18.60
```

Resolve the error by adding a temporary tablespace to the primary database.

5.1.3.9 Specify a Valid DB2 Port Number

Remote sources for DB2 cannot be created if the DB2 port is outside the allowed range of 1-65335.

Context

An invalid port error may be indicated during remote source creation.

For example:

```
(Catalog) Error reading Remote Object: InternalError:
dberror(CallableStatement.execute): 403 - internal error.: Cannot get remote source
objects: port out of range:70000
```

Resolve the error by configuring DB2 with a port in the valid range (1-65335).

Procedure

1. Change the DB2 port in the services file.
   - On Windows, `C:\Windows\System32\drivers\etc\services`
   - On UNIX or Linux, `/etc/services`
2. Update the database manager configuration.

   ```
   db2> update database manager configuration using svcename [<service_name> |
   <port_number>]
   ```
3. Restart the DB2 server.
4. Re-create the remote source.

5.1.3.10 Verify the DB2 Native Connection Settings

The LogReader may fail to initialize properly when the DB2 native connection fails.

Context

When the native connection fails, you may see the following error in the log:

```
[ERROR] com.sap.hana.dp.db2logreaderadapter.DB2RepAgentWrapper.initialize[858]    -
Failed to initialize LogReader.
Could not find Resource Bundle containing index: Could not get the log end locator
because: Native database connection failed with code <-1>.
```

Resolve the error by verifying that the connection details for your DB2 database are configured correctly:

- Host
- Port
- Database Name
- Database Source Name

5.1.3.11 Define the SAP ASE Adapter Interface

Real-time replication tasks on the SAP ASE adapter may fail if an entry for the adapter has not been added to the interface file of the data server.

Context

An error may be indicated when creating a remote source.

For example:

```
Error reading Remote Object: Internal Error: dberror(CallableStatement.execute):
403 - internal error: Cannot get remote source objects: Unknown streamType:
```

Resolve the error by adding an entry for the SAP ASE adapter to the interface file on the SAP ASE server.

Procedure

1. Add the entry to the interface file on the SAP ASE data server.

   ```
   <entry_name>
   master tcp ether <agent_hostname> <port>
   query tcp ether <agent_hostname> <port>
   ```

   > **Note**
   >
   > The entry name must match the adapter instance name specified in the remote source configuration. The port number must match the SAP ASE adapter server port configured in `<DPAgent_root>/Sybase/interfaces`.

2. Re-execute the replication task.

5.1.4 Resolve Improper Adapter Configurations on the Agent

Real-time replication tasks may fail when the associated adapters have not been properly configured on the Data Provisioning Agent.

Provide a Compatible JDBC Driver [page 102]
Replication tasks may fail if the JDBC driver is not provided or does not match the version of the source database.

Specify a Compatible Java Runtime Environment [page 103]
Replication errors may occur if a non-compatible Java Runtime Environment (JRE) is provided.

Configure the DB2 Environment Variables [page 103]
Remote subscriptions may fail to queue if the DB2 runtime environment variables are not set on the Data Provisioning Agent host.

5.1.4.1 Provide a Compatible JDBC Driver

Replication tasks may fail if the JDBC driver is not provided or does not match the version of the source database.

Context

When a compatible JDBC driver is not provided, a replication task failure error may be indicated.

For example:

```
Error reading Remote Object: InternalError: dberror(CallableStatement.execute): 403
- internal error: Cannot get remote source objects: Failed to install and start
Oracle JDBC driver bundle.
```

Resolve the error by providing the correct JDBC driver.

- When available, use the JDBC driver distributed with the source database installation.
  For example:
  - For Oracle, `$ORACLE_HOME/jdbc/lib/ojdbc7.jar`
  - For DB2, `$INSTHOME/sqllib/java/db2jcc4.jar`
- The JDBC driver version should not be lower than the source database version.
  You can use the `java` command to verify the JDBC driver version. For example:

```
java -jar ojdbc7.jar
Oracle 12.1.0.2.0 JDBC 4.1 compiled with JDK7 on Mon_Jun_30_11:30:34_PDT_2014
#Default Connection Properties Resource
#Thu Sep 29 05:37:00 PDT 2016
```

Procedure

1. Copy the JDBC driver to the `<DPAgent_root>/lib` directory.
2. Reopen the replication task and add objects from the remote source.

5.1.4.2 Specify a Compatible Java Runtime Environment

Replication errors may occur if a non-compatible Java Runtime Environment (JRE) is provided.

Context

We recommend that you use the SAP JVM bundled with the Data Provisioning Agent.

For complete information about supported Java Runtime Environment versions, see the *Product Availability Matrix (PAM)*.

5.1.4.3 Configure the DB2 Environment Variables

Remote subscriptions may fail to queue if the DB2 runtime environment variables are not set on the Data Provisioning Agent host.

Context

A replication failure may indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[INFO ] com.sap.hana.dp.db2logreaderadapter.DB2RepAgentWrapper.initialize[1186]
- Initializing LogReader ...
[ERROR] com.sap.hana.dp.adapter.framework.core.WorkerThread.processRequest[354]
- /rqac48lnx3_work1/dpfiles/dataprovagent/LogReader/lib/linux64/libsybrauni98.so:
libdb2.so.1: cannot open shared object file: No such file or directory Context:
java.lang.UnsatisfiedLinkError: /rqac48lnx3_work1/dpfiles/dataprovagent/
LogReader/lib/linux64/libsybrauni98.so: libdb2.so.1: cannot open shared object
file: No such file or directory
```

Resolve the error by setting the DB2 runtime environment variables prior to starting the Data Provisioning Agent.

Procedure

1. Source the DB2 profile in the `<DPAgent_root>/bin/DPAgent_env.sh` file.
   For example, add a line such as the following:

   ```
   source /home/db2inst1/sqllib/db2profile
   ```

   > **Tip**
   >
   > By sourcing the profile in the `DPAgent_env.sh` file, the environment variables will be set each time you use the Data Provisioning Agent Configuration tool.

2. Restart the agent with the Data Provisioning Agent Configuration tool.
3. Re-execute the replication task.

5.1.5 Resolve Uncommitted Source Database Transactions

Real-time replication tasks may fail when attempting to queue a remote subscription if the source database or table has uncommitted transactions.

Context

A replication task error may be indicated.

For example:

```
SQL ERROR--
Message: ORA-00054: resource busy and acquire with NOWAIT specified or timeout
expired
SQLState: 61000
Remote Code: 54
Message: SQL execution failed
```

Resolve the error by committing any transactions in the source database or table.

Procedure

1. Ensure that any transactions have been committed in the source database or table.
2. Re-execute the replication task.
   For trigger-based adapters such as SAP HANA or Teradata, reset the remote subscription.

5.1.6 Resolve Log Reader Instance Port Conflicts

Log reader adapters require an instance port that must not be used by any other applications on the Data Provisioning Agent host.

Context

Log reader adapter instances cannot be created when the specified instance port is already in use.

Procedure

1. Suspend any remote sources.
2. Edit the remote source configuration and specify an instance port that is not in use on the agent host.
3. Resume any remote sources.
4. In the *Data Provisioning Remote Subscription Monitor*, process any remote subscriptions by choosing *Retry Operations*.

Related Information

Suspend and Resume Remote Sources [page 44]
Monitoring Remote Subscriptions [page 12]

5.1.7 Resolve Data Provisioning Server Timeouts

When the default message timeout value for the Data Provisioning Server is too short, real-time replication tasks may fail during the initial load operation.

Context

The data provisioning adapter used by the replication task reports a timeout error.

For example:

```
Error: (256, 'sql processing error: QUEUE: SUB_T002: Failed to add subscription for
remote subscription SUB_T002[id = 165727] in remote source MSSQLECCAdapterSrc[id =
165373]. Error: exception 151050: CDC add subscription failed: Request timed out.\n
\n: line 1 col 1 (at pos 0)')
```

The error can be resolved by increasing the message timeout value or cleaning up the source database archive log.

Procedure

- Increase the timeout value by adjusting the `messageTimeout` value.

```
ALTER SYSTEM ALTER CONFIGURATION ('dpserver.ini','SYSTEM') SET ('framework',
'messageTimeout')= '9999'  WITH RECONFIGURE;
```

- Clean the source database archive log.

Related Information

Cleaning LogReader Archives Safely [page 83]

5.1.8 Load Clustered and Pooled Table Metadata into SAP HANA

Real-time replication tasks with clustered or pooled tables may fail when metadata has not been correctly loaded into the SAP HANA database.

Context

Missing metadata is indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[ERROR] com.sap.hana.dp.cdcadaptercommons.StatefulAdapterCDC
$CDCOpened.addSubscription[791]    - [OracleECCAdapterSrc] Failed to add the first
subscription. SubscriptionSpecification [header=remoteTableId=68,
remoteTriggerId=160778, sql=, subscription=, customId=, seqID=SequenceId [value=[0,
0, 0, 0]], isLastSubscription=true, withSchemaChanges=false,
firstSubscriptionOnTable=true, lastSubscriptionOnTable=true]
java.lang.NullPointerException: while trying to invoke the method
com.sap.hana.dp.adapter.sdk.parser.Query.getFromClause() of a null object loaded
from local variable 'query'
```

Resolve the error by replicating the metadata into SAP HANA.

Procedure

1. Edit the dictionary replication script with the name of the remote source and the SAP HANA schema name.
   By default, the dictionary replication script is located at `<DPAgent_root>/LogReader/scripts/replicate_dictionary.sql`.
2. Execute the dictionary replication script in the SAP HANA database.
3. Re-execute the failed replication task.

5.2 Recovering from Replication Failures

Replication tasks may fail and generate remote source or subscription exceptions for a number of reasons.

Generally, recovering replication involves processing any exceptions in addition to other tasks.

5.2.1 Check for Log Reader Errors

Replication may stop if there are log reader errors.

For more information about log reader errors, check the trace file located in the `log` folder of your Data Provisioning Agent instance. You may also choose to increase the trace log levels for the Data Provisioning Server.

Related Information

Activate Additional Trace Logging for the Data Provisioning Server [page 125]

5.2.2 Recover from a Source Table DDL Schema Change

Table replication may fail when there is a DDL schema change in the source table and the replication task is not configured to replicate with structure.

Context

When table replication fails, remote source and remote subscription exceptions may be generated.

Recover replication by processing any exceptions and re-creating the replication task.

Procedure

1. In the *Data Provisioning Remote Subscription Monitor*, verify any reported exceptions.
   Schema change exceptions on the failed replication task cannot be ignored or retried.
2. In the SAP HANA Web-based Development Workbench editor, drop the failed replication task.
3. Re-create the replication task with the “Initial + realtime with structure” replication behavior.
4. Activate and execute the new replication task.
5. In the *Data Provisioning Remote Subscription Monitor*, ignore the schema change exception.

5.2.3 Recover from a Truncated Source Table

A replication task may fail when a source table is truncated.

Context

Restore replication by truncating the target table and processing any remote subscription exceptions.

Procedure

1. In the *Data Provisioning Remote Subscription Monitor*, verify any remote subscription errors.
2. Truncate the target table in the SAP HANA system to restore data consistency between the source and target tables.
3. In the *Data Provisioning Remote Subscription Monitor*, ignore the TRUNCATE TABLE errors.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]

5.2.4 Recover from Source Table and Replication Task Recreation

Restore replication after a failure when a source table and replication task are both recreated in a short timeframe.

Context

A replication failure may occur under the following conditions:

- There are multiple active remote subscriptions to the same source database
- There is a significant amount of change data on the source table or other subscribed tables that has not yet been replicated to the SAP HANA target
- The remote subscription or replication task is dropped, and the source table and replication task are recreated in a short timeframe

Under these circumstances, Log Reader adapters may capture the DROP TABLE DDL and attempt to replicate it to the SAP HANA target, generating an exception.

Restore replication by processing any remote source and subscripton exceptions.

Procedure

1. In the *Data Provisioning Remote Subscription Monitor*, check for remote subscription errors.
2. Restore replication by selecting the exception and choosing *Ignore*.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]

5.2.5 Recover from a Source and Target Data Mismatch

Replication may fail when a new row is inserted in a source table that has a primary key and the target table already contains the row.

> **Note**
>
> Updating or deleting source table rows that do not exist in the target table does not cause a replication failure.

Context

Recover replication by processing any remote subscription exceptions and re-executing the replication task.

Procedure

1. In the *Data Provisioning Remote Subscription Monitor*, identify any errors.
2. Select the exception and choose *Ignore* to restore replication.
3. (Optional) Re-execute the replication task to avoid any data mismatch between the source and target tables.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]

5.2.6 Recover from an Agent Communication Issue

Process any remote subscription exceptions and restore replication when a network interruption or other communication issue occurs between the Data Provisioning Agent and the SAP HANA server.

Context

A communication issue may generate remote subscription exceptions such as the following:

- Connection to the agent has been lost
- Connection to all the agents in the cluster has been lost
- One or more subscriptions failed for the remote source

In some scenarios, no exceptions are generated, but changed source data is not replicated to the target. In these scenarios, an error message may be indicated in the Data Provisioning Server trace log.

For example:

```
Generic stream error: getsockopt, Event=EPOLLERR - , rc=111: Connection refused
$NetworkChannelBase$=
92 [0x00007f054cdb1798]  {refCnt=4, idx=4} 10.173.160.203/0_tcp-
>10.173.160.203/8780_tcp ConnectWait,[r--c]
exception throw location:
 1: 0x00007f23a75ec672 in Stream::NetworkChannelCompletionThread::run(void*&)+0x420
at
NetworkChannelCompletion.cpp:548 (libhdbbasis.so)
 2: 0x00007f23a752a047 in Execution::Thread::staticMainImp(void**)+0x743 at
Thread.cpp:463 (libhdbbasis.so)
 3: 0x00007f23a752b698 in Execution::Thread::staticMain(void*)+0x34 at
ThreadMain.cpp:26 (libhdbbasis.so)
[58457]{-1}[-1/-1] 2016-11-03 00:33:53.499182 e Stream
NetworkChannelCompletion.cpp(00622) : NetworkChannelCompletionThread #5 91
[0x00007f054cdb1958]  {refCnt=4, idx=5} 10.173.160.203/0_tcp-
>10.173.160.203/8780_tcp ConnectWait,[r--c]
: Error in asynchronous stream event: exception  1: no.2110001  (Basis/IO/Stream/
impl/NetworkChannelCompletion.cpp:548)
```

Procedure

1. Suspend all remote sources on the affected agent.
2. Restart the Data Provisioning Agent.
3. Check whether the status of the agent TCP port is “LISTEN” or “ESTABLISHED”.
   - On Windows, `netstat -na | findstr "5050"`
   - On Linux, `netstat -na|grep 5050`

   Additionally, verify that no operating system firewall rules are configured for the agent TCP port.
4. Restart the Data Provisioning Server on the SAP HANA system.
5. Resume all sources on the affected agent.
6. In the *Data Provisioning Remote Subscription Monitor*, retry or ignore any remaining remote subscription exceptions.

7. Verify the remote source and remote subscription statuses.
    - Remote source CDC status: OK
    - Remote subscription state: APPLY_CHANGE_DATA
8. Check for data consistency by comparing the source and target table row counts.

5.2.7 Resolve Stopped or Delayed Replication on Oracle

If replication from an Oracle source system is stopped or delayed, or you notice poor performance, check the instance log for reports of unsupported transactions.

Context

When the Oracle LogMiner starts scanning from the middle of a transaction and fails to translate the raw record, it reports an unsupported operation. This may occur most often on UPDATE operations involving wide tables.

The Oracle Log Reader adapter can manage these records by using a standby scanner, but frequent occurrence of unsupported operations can slow scan performance.

You have several options to reduce the number of unsupported operations.

Procedure

- Upgrade to a newer version of the Data Provisioning Agent, if possible.
- If parallel scanning is enabled, increase the value of the *Parallel scan SCN range* parameter.
- Disable parallel scanning.

Related Information

Data Provisioning Agent Log Files and Scripts [page 120]

5.2.8 Resolve Locked SAP HANA Source Tables

A replication task or flowgraph may fail if an SAP HANA source table is locked and the remote subscription request times out.

Context

A replication task or flowgraph that has timed out due to a locked table may fail with the following error:

```
SAP DBTech JDBC: [256]: sql processing error: QUEUE: RT_SFC100: Failed to add
subscription for remote subscription RT_SFC100.Error: exception 151050: CDC add
subscription failed: Request timed out.
```

Additionally, you can verify a locked table by checking the blocked transactions in the SAP HANA source.

To resolve a locked SAP HANA source table, refer to SAP Note 0001999998.

5.2.9 Reset the Remote Subscription

When log reader-based real-time replication has stopped, you can try resetting the remote subscription.

Procedure

1. From a SQL console, reset the remote subscription.

   ```
   ALTER REMOTE SUBSCRIPTION <subscription_name> RESET;
   ```

   > **Tip**
   >
   > If you have the required privileges, you can also reset remote subscriptions from the *Data Provisioning Remote Subscription Monitor*. For more information, see *Manage Remote Subscriptions*.

2. Check the Data Provisioning Agent for the adapter instance.

   In `<DPAgent_root>/LogReader`, look for a directory with the same name as your remote subscription instance. Back up and delete the folder, if it exists.

3. Execute the cleanup script for your database type.

   The cleanup scripts are located in `<DPAgent_root>/LogReader/scripts`.

   > **Note**
   >
   > Execute the script using the same user that is configured for replication.

4. Stop the SAP HANA Data Provisioning Agent service.

- On Windows, use the Services manager in Control Panel.
- On Linux, run `./dpagent_service.sh stop`.

5. (Optional) Enable additional logging on the Data Provisioning Agent.
   a. Open the Agent configuration file in a text editor.

      The configuration file is located at `<DPAgent_root>/dpagentconfig.ini`.
   b. Change the parameter `framework.log.level` from INFO to ALL.

   Increasing the log level generates additional information useful for further debugging, if necessary. You can safely revert the log level after resolving the issue.
6. Restart the SAP HANA Data Provisioning Agent service.
   - On Windows, use the Services manager in Control Panel.
   - On Linux, run `./dpagent_service.sh start`.
7. From a SQL console, queue and distribute the remote subscription.

```
ALTER REMOTE SUBSCRIPTION <subscription_name> QUEUE
```

```
ALTER REMOTE SUBSCRIPTION <subscription_name> DISTRIBUTE
```

> **Tip**
>
> If you have the required privileges, you can also queue and distribute remote subscriptions from the *Data Provisioning Remote Subscription Monitor*. For more information, see *Manage Remote Subscriptions*.

Next Steps

If you are unable to reset the remote subscription, you may need to clear any outstanding remote subscription exceptions first.

Related Information

Clear Remote Subscription Exceptions [page 114]
Manage Remote Subscriptions [page 46]

5.2.10 Clear Remote Subscription Exceptions

Replication may be stopped due to remote subscription exceptions. For example, a remote subscription exception may be generated when a primary key violation, constraint violation, or null primary key occurs.

Remote subscription exceptions are reported in multiple ways.

- Remote subscription exceptions appear in the task execution log.

- Email alerts may be sent to the administrator.
- Exceptions may be displayed in monitoring, or if you query the remote subscription exception public view.

After correcting the root cause of the remote subscription exception, you can clear the exceptions with a SQL statement:

```
PROCESS REMOTE SUBSCRIPTION EXCEPTION <subscription_id> IGNORE
```

5.3 Recovering from Crashes and Unplanned System Outages

Resume replication and ensure data consistency when you experience a crash or unplanned system outage.

5.3.1 Recover from an Index Server Crash

Verify your data consistency when you an experience an SAP HANA index server crash.

Procedure

1. After SAP HANA restarts, verify that all SAP HANA processes show an active GREEN status.
2. Compare source and target table row counts to verify data consistency.

5.3.2 Recover from a Data Provisioning Server Crash

Restart replication when the Data Provisioning Server crashes.

Context

When the Data Provisioning Server crashes, an error may be indicated in the `usr/sap/<sid>HDB<instance>/<hana_machine_name>/trace/dpserver_alert_<hana_machine_name>.trc` log file.

For example:

```
Instance LR1/00, OS Linux lsdub0007 3.0.101-80-default #1 SMP Fri Jul 15 14:30:41
UTC 2016 (eb2ba81) x86_64
----> Register Dump <----
  rax: 0x00007f638b068a90  rbx: 0x00007f5fa74a8010
  rcx: 0x0000000000000000  rdx: 0x0000000000000056
......
  xmm[14]: 0x00000000.3fffffff.3fffffff.00000000
  xmm[15]: 0x3fffffff.00000000.03020100.07060504
NOTE: full crash dump will be written to /usr/sap/LR1/HDB00/lsdub0007/trace/
dpserver_lsdub0007.30011.crashdump.20161102-143544.019227.trc
Call stack of crashing context:
 1: 0x0000000000000000 in <no symbol>+0x0 (<unknown>)
```

Recover from the crash by restarting replication and processing any remote subscription exceptions.

Procedure

1. Suspend all remote sources.
2. Restart the Data Provisioning Server.
3. Restart the SAP HANA system.
4. Resume all remote sources.
5. Process any remote subscription exceptions remaining after change data capture has resumed.

Next Steps

In a multiple container SAP HANA configuration, the Data Provisioning Server is managed individually in each tenant database. Repeat the recovery process for each tenant database.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]

Suspend and Resume Remote Sources [page 44]

5.3.3 Recover from a Data Provisioning Agent JVM Crash

The Data Provisioning Agent may crash if the Java virtual machine is configured with insufficient available memory.

Context

An error may be indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[ERROR] OracleRepAgentWrapper$2.run  - LogReader is not in REPLICATE state. State:
REPLICATION DOWN.
[ERROR] ReceiverImpl.sendError  - LogReader is in ERROR state [GC overhead limit
exceeded]. Check LogReader log for details.
[ERROR] ReceiverImpl.sendError  - Receiver is closed due to adapter error.Please
check REMOTE_SUBSCRIPTION_EXCEPTIONS table
```

Recover by adjusting the agent's maximum available memory and restarting the agent.

Procedure

1. Check the available memory on the Data Provisioning Agent host machine.

   For example:

   ```
   # free -m
                total       used       free     shared    buffers     cached
   Mem:        258313     196343      61970          0        589      53022
   -/+ buffers/cache:     142731     115582
   Swap:         2047          0       2047
   ```

2. Stop the Data Provisioning Agent service.
3. Adjust the agent maximum available memory in the agent runtime options.
4. Restart the Data Provisioning Agent service.

5.3.4 Recover from an Unplanned Source Database Outage

Replication may stop if there is an unplanned source database outage due to a network or hardware issue.

> **Note**
>
> On Oracle, the Oracle Log Reader adapter automatically reconnects to the source database when available and resumes replication without any additional actions.

Context

A source database outage may be indicated by multiple error types in the logs.

For example:

```
Could not connect to <jdbc:oracle:thin:@host:1675:DDCSTD>: The Network Adapter
could not establish the connection
```

```
java.lang.Exception: Log scanner <LogMinerScanner_1> stopped because of error: No
more data to read from socket
```

Recover by processing remote subscription exceptions and resuming replication on the remote source.

Procedure

1. Suspend the remote source until the outage is resolved.
2. Process any remote subscription exceptions.
3. Resume replication on the remote source.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]
Suspend and Resume Remote Sources [page 44]

5.3.5 Recover from an ASE Adapter Factory Crash

The dpadapterfactory process used in the SAP ASE and SAP ECC ASE adapters restarts automatically if it is stopped or crashes.

Context

An adapter factory crash may be indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[ERROR] com.sap.hana.dp.adapter.framework.cpp.CppAdapterManager.read[236]    -
Socket closed
com.sap.hana.dp.adapter.sdk.AdapterException: Socket closed
at
com.sap.hana.dp.adapter.framework.socket.SocketConnector.read(SocketConnector.java:
103)
```

```
at
com.sap.hana.dp.adapter.framework.cpp.CppAdapterManager.read(CppAdapterManager.java:
232)
```

Additional errors may be indicated in the <DPAgent_root>/log/dpsadapterfactory_<##########>.txt log file.

For example:

```
e Basis  FaultProtectionImpl.cpp(01610) + Instance (none)/(none), OS Linux rqahana3
3.0.101-0.46-default #1 SMP Wed Dec 17 11:04:10 UTC 2014 (8356111) x86_64
e Basis  FaultProtectionImpl.cpp(01610) + ----> Register Dump <----
e Basis  FaultProtectionImpl.cpp(01610) +   rax: 0xfffffffffffffffc  rbx:
0x00007f54e1adaa20
e Basis  FaultProtectionImpl.cpp(01610) +   rcx: 0xffffffffffffffff  rdx:
0x0000000000000000
……
Helper.cpp(00514) : Using 'x64_64 ABI unwind' for stack tracing
e Basis  FaultProtectionImpl.cpp(01610) + NOTE: full crash dump will be written to /
rqahana3_work2/songp/DPAgent_1.3.0_5030/log/dpsadapterfactory.crashdump.
20161020-233253.081342.trc
```

A full crash dump is written to <DPAgent_root>/log/dpsadapterfactory.crashdump.<timestamp>.trc

Resolve the errors by processing any remote exceptions.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]

5.4 Troubleshooting Data Provisioning Agent Issues

This section describes error situations related to the Data Provisioning Agent and their solutions.

Uninstalled Agent Reports Alerts or Exceptions [page 123]
After uninstalling the Data Provisioning Agent and installing a new version, alerts and exceptions may be reported for the previous installation.

Create an Agent System Dump [page 123]
Use the command-line agent configuration tool to generate system dumps when troubleshooting the Data Provisioning Agent.

Resolve Agent Parameters that Exceed JVM Capabilities [page 124]
Replication tasks may fail during the initial load if the fetch size exceeds the capabilities of the Data Provisioning Agent JVM.

5.4.1 Data Provisioning Agent Log Files and Scripts

Various log files are available to monitor and troubleshoot the Data Provisioning Agent.

The following log files are available:

Log File Name and Location	Description
`<DPAgent_root>/log/dpagent_service_eventlog_<date>.log`	Data Provisioning Agent event log
`<DPAgent_root>/log/framework_alert.trc`	Data Provisioning Agent adapter framework log. Use this file to monitor data provisioning agent statistics.
`<DPAgent_root>/log/framework.trc`	Data Provisioning Agent adapter framework trace log. Use this file to trace and debug data provisioning agent issues
`<DPAgent_root>/log/configtool.log`	Data Provisioning Agent Configuration Tool log
`<DPAgent_root>/LogReader/<instance>/log/<instance>.log`	Instance log for log reader-based adapters
`<DPAgent_root>/LogReader/admin_logs/admin<timestamp>.log`	Administrative log for log reader-based adapters

Additionally, scripts for performing source database initialization and cleanup operations can be found at `<DPAgent_root>/LogReader/scripts`.

5.4.2 Clean an Agent Started by the Root User

If the Data Provisioning Agent is accidentally started by the root user, you should clean up the agent installation. In a normal configuration, the Data Provisioning Agent should be started only by a user with the same rights and permissions as the installation owner defined during the agent installation.

Context

If the agent is started by the `root` user, additional files are created in the installation location. The agent cannot access these additional files because `root` is their owner, and they should be removed.

> **Note**
>
> This applies only if the agent was started by the user actually named `root`, and not other users that may belong to the root group or have similar permissions.

Procedure

1. Navigate to the `configuration` directory in the agent installation location.

   For example, `/usr/sap/dataprovagent/configuration`
2. Remove the following directories:
   - `com.sap.hana.dp.adapterframework`
   - `org.eclipse.core.runtime`
   - `org.eclipse.osgi`

   > **Caution**
   >
   > Do not remove the `config.ini` file or `org.eclipse.equinox.simpleconfigurator` directory.
3. Remove the `log` directory and `secure_storage` file.

   For example, `/usr/sap/dataprovagent/log` and `/usr/sap/dataprovagent/secure_storage`.

   > **Note**
   >
   > Depending on the permissions, you may require sudo access to remove the `secure_storage` file.
   >
   > For example, `sudo rm -rf secure_storage`.

5.4.3 Agent JVM Out of Memory

If the Java Virtual Machine on the Data Provisioning Agent is running out of memory, you may need to adjust configuration parameters on the Agent or your adapters.

Oracle log reader adapters

For Oracle log reader adapters, there are multiple possible solutions.

- Check the *Queue size of parallel scan tasks* parameter.
  The default value is 0, which allows for an unlimited queue size. To find the maximum queue size that results in acceptable memory use, set the queue size to 10,000 and increase in increments of 50,000 or 100,000.
- Check the *Parallel scan SCN range* parameter.
  Decrease the value as necessary.
- Check the *Number of parallel scanners* parameter.
  Decrease the value from 4 to 2.

General virtual memory issues

For general memory issues, you can increase the JVM memory usage.

In `dpagent.ini`, change the memory allocation parameter, and restart the Agent.

For example, to increase the memory to 8GB, `-Xmx8096m`.

5.4.4 Adapter Prefetch Times Out

When task execution fails due to an adapter timeout, you may need to adjust the adapter prefetch timeout parameter.

In the Index Server trace log, you may notice an error similar to the following:

```
DPAdapterAccess.cpp(01825) : DPAdapterAccess::Fetch: failed with error: Prefetch
timed out.
```

When this type of error appears, you can adjust the value of the adapter prefetch timeout parameter.

From a SQL console:

```
ALTER SYSTEM ALTER CONFIGURRATION ('dpserver.ini','SYSTEM')
SET ('framework','prefetchTimeout')='<seconds>' WITH RECONFIGURE;
```

For example, you might change the value to **600** to set the timeout to 10 minutes.

5.4.5 Agent Reports Errors when Stopping or Starting

If the Data Provisioning Agent reports error code 5 when stopping or starting on Windows, you may need to run the configuration tool as an administrator.

To run the configuration tool once as an administrator, right-click on `dpagentconfigtool.exe` and choose *Run as administrator*.

Alternatively, you can set the configuration tool to always run as an administrator. Right-click on `dpagentconfigtool.exe` and choose *Properties*. From the *Compatibility* tab, choose *Run this program as an administrator* and click *OK*.

5.4.6 Uninstalled Agent Reports Alerts or Exceptions

After uninstalling the Data Provisioning Agent and installing a new version, alerts and exceptions may be reported for the previous installation.

If the agent is uninstalled, but associated remote sources and the agent registration are not dropped, any replication tasks still using remote sources on the previous agent may report alerts and exceptions.

Solution

Drop outdated remote sources and the agent registration referring to the old installation.

To suspend and drop a remote source:

```
ALTER REMOTE SOURCE "<remote_source_name>" SUSPEND CAPTURE;
DROP REMOTE SOURCE "<remote_source_name>" CASCADE;
```

To drop an agent registration:

```
DROP AGENT "<agent_name>" CASCADE;
```

If you do not know the old agent name, you can find it using `SELECT * FROM sys.m_agents`.

5.4.7 Create an Agent System Dump

Use the command-line agent configuration tool to generate system dumps when troubleshooting the Data Provisioning Agent.

Context

By default, the configuration tool generates system dumps that include the following information:

- Log reader, framework, and OSGi logs and traces
- Information about running Java and Data Provisioning Agent processes
- Information about JVM, threads, OSGi, and adapters, if the agent connection is available

> **Note**
>
> To export a system dump, the Data Provisioning Agent must be started and running.

Procedure

1. At the command line, navigate to `<DPAgent_root>/bin`.
2. Execute the command using the `--createFullSystemDump` parameter.
   - On Windows, **`agentcli.bat --createFullSystemDump [<additional_parameters>]`**
   - On Linux, **`./agentcli.sh --createFullSystemDump [<additional_parameters>]`**

Table 31: Supported Parameters

Parameter	Description
`--snapshotName <name>`	Specifies the name for the exported dump. By default, the exported filename is `<hostname>_<time>_<name>`. If no snapshot name is specified, "dpagent" is used as the name.
`--runtimeOnly`	Restricts the exported dump to runtime information only. The exported dump will not include a copy of the logs or other folders.

5.4.8 Resolve Agent Parameters that Exceed JVM Capabilities

Replication tasks may fail during the initial load if the fetch size exceeds the capabilities of the Data Provisioning Agent JVM.

Context

An error may be indicated during the initial load.

For example:

```
Could not execute 'SELECT * FROM "SYSTEM"."V_ZMTEST_DPSIMPLE"' in 10:00.197
minutes .
SAP DBTech JDBC: [403]: internal error: Error opening the cursor for the remote
database Prefetch timed out. for query "SELECT "V_ZMTEST_DPSIMPLE"."COL1",
"V_ZMTEST_DPSIMPLE"."COL2", "V_ZMTEST_DPSIMPLE"."COL3", "V_ZMTEST_DPSIMPLE"."COL4",
"V_ZMTEST_DPSIMPLE"."COL5", "V_ZMTEST_DPSIMPLE"."COL6", "V_ZMTEST_DPSIMPLE"."COL7"
FROM "QARUSER.DPTEST01" "V_ZMTEST_DPSIMPLE" "
```

In addition, an error may be indicated in the `<DPAgent_root>/log/framework.trc` log file.

For example:

```
[FATAL] DPFramework | WorkerThread.processRequest  - Java heap space (failed to
allocate 40000016 bytes) (max heap: 256 MB)[FATAL] DPFramework |
WorkerThread.processRequest  - Java heap space (failed to allocate 40000016 bytes)
(max heap: 256 MB)
```

Resolve the errors by reducing the row fetch size and increasing the JVM maximum size.

Procedure

1. Drop all replication tasks.
2. Increase the row fetch size in the Agent Preferences.
3. Increase the maximum virtual machine size in the Agent Runtime Options.
4. Restart the Data Provisioning Agent.
5. Process any remote subscription exceptions.

Related Information

Processing Remote Source or Remote Subscription Exceptions [page 48]

5.5 Troubleshooting Other Issues

This section describes various issues unrelated to replication failures or the Data Provisioning Agent and their solutions.

Activate Additional Trace Logging for the Data Provisioning Server [page 125]
The trace logs contain detailed information about actions in the Data Provisioning server.

Resolve a Source and Target Data Mismatch [page 128]
If data in the source and target tables of a replication task is mismatched, fix the error by re-executing the replication task.

Configuring the Operation Cache [page 128]
You can improve performance by using an operation cache in some script servers.

Ensure Workload Management and Resource Consumption [page 130]
Ensure that in circumstances of limited memory or CPU resources that processes and system resources remain responsive.

5.5.1 Activate Additional Trace Logging for the Data Provisioning Server

The trace logs contain detailed information about actions in the Data Provisioning server.When you are troubleshooting issues, you can gain additional insight about the root cause by increasing the level of detail in the logs.

Prerequisites

To configure traces, you must have the system privilege TRACE ADMIN.

Procedure

1. Log on to SAP HANA studio with a user that has system privilege TRACE ADMIN.
2. In the Administration editor, choose the *Trace Configuration* tab.
3. Choose *Edit Configuration* under "Database Trace".
4. Filter the component list for `dp*`.
5. Set the *System Trace Level* to `DEBUG` or for any relevant components under the *DPSERVER* and *INDEXSERVER* nodes.

Scenario	Traces
Debugging Initial Load	○ *DPSERVER* ○ dpframework Trace information for message handling between the index server, Data Provisioning Server, and Data Provisioning Agent. ○ dpframeworkprefetch Prefetch-specific trace information. ○ dpframeworkmessagebody Trace information useful for diagnosing incorrect numbers of rows or suspected data corruption. **Note** This option can generate a large amount of data in the trace log. ○ *INDEXSERVER* ○ dpadaptermanager Trace information for initial smart data access requests. ○ *XSENGINE* ○ dpadaptermanager Trace information for initial smart data access requests from the SAP HANA Web-based Development Workbench.
Debugging Real-time Replication	○ *DPSERVER* ○ dpframework Trace information for message handling between the index server, Data Provisioning Server, and Data Provisioning Agent. ○ dpreceiver Trace information for message receiver, including the hand off of row data received by the framework. ○ dpdistributor Trace information for the message distributor. ○ dpapplier Trace information for the message applier. ○ dpserver

Scenario	Traces
	Trace information for communication between the Data Provisioning Server and the index server. ○ dpremotesubscriptionmanager Trace information for remote subsciption runtime details on the Data Provisioning Server side. ○ dpframeworkmessagebody Trace information useful for diagnosing incorrect numbers of rows or suspected data corruption. **Note** This option can generate a large amount of data in the trace log. ○ *INDEXSERVER* ○ dpdistributor Trace information for the message distributor. ○ dpapplier Trace information for the message applier. ○ dpremotesubscriptionmanager Trace information for remote subscription runtime details on the index server side.

Tip

To ensure that applicable trace information is captured at the time of the error and not overwritten by log wrapping, you can increase the maximum log size while capturing the traces. In the *GLOBAL* node, increase the value of the *maxfilesize* parameter.

6. Click *Finish*.

Results

Additional debug information for the modified components is added to the trace log.

Tip

You can also modify the trace log levels by issuing commands from a SQL console and manually specifying the component to adjust.

For example, to set the Data Provisioning Server *dpframework* trace to **DEBUG**:

```
ALTER SYSTEM ALTER CONFIGURATION ('dpserver.ini','SYSTEM') SET
('trace','dpframework') = 'DEBUG' WITH RECONFIGURE;
```

5.5.2 Resolve a Source and Target Data Mismatch

If data in the source and target tables of a replication task is mismatched, fix the error by re-executing the replication task.

5.5.3 Configuring the Operation Cache

You can improve performance by using an operation cache in some script servers.

The operation cache holds operation instances for Global Address Cleanse, Universal Data Cleanse, Geocode, and Type Identifier (TID), which are initialized and ready for use during task plan execution. This improves performance by avoiding the process of task plan operation initialization/create and deletion, and allows the re-use of the cached instances both inside a single plan and across plans.

Having more instances cached improves performance, but those additional cached instances consume more memory.

Operation cache instances are type specific and are set in the file `scriptserver.ini`.

You can use the following monitoring views to verify the cached operations, usage count, and so on.

- Select * from sys.m_caches;
- Select * from sys.m_cache_entries;

The operation cache can be configured in SAP HANA studio by editing the file `scriptserver.ini`.

Enabling operation cache

To start operation cache instances, you must set the *enable_adapter_operation_cache* parameter to “yes”. The following sample SQL statement sets the parameter to “yes”.

```
ALTER SYSTEM ALTER CONFIGURATION ('scriptserver.ini','SYSTEM') SET
('adapter_operation_cache', 'enable_adapter_operation_cache') = 'yes' WITH
RECONFIGURE;
```

Changing the operation cache default settings

You can turn off the cache for each node, or change the default number of instances. The following sample SQL statement changes the number of instances.

```
ALTER SYSTEM ALTER CONFIGURATION ('scriptserver.ini', 'SYSTEM') SET
('adapter_operation_cache', 'gac')='30';
```

By default, the operation cache is enabled for all the supported types of operations (which is recommended) and has the following number of instances.

Table 32: Operation cache default settings

Option	Number of instances default setting
Geocode	10
Global Address Cleanse (in the Cleanse node)	20
Universal Data Cleanse (in the Cleanse node)	60
Type Identification	20

Ensuring that real time jobs have priority

The operation cache is used for both batch jobs and real time jobs. Batch jobs can exhaust the operation cache, leaving insufficient resources to optimize the running of real time jobs. If you run real time jobs, use these settings to ensure that a dedicated number of operation cache instances are available for real time tasks. By default, the number of instances made available to real time tasks is half the total number of instances for each option.

Table 33: Real time default settings

Option	Number of instances default setting
Geocode	5
Global Address Cleanse (in the Cleanse node)	10
Universal Data Cleanse (in the Cleanse node)	30
Type Identification	0

Caching performance with the Global Address Cleanse

Within the Cleanse node, you can configure two options that are relevant to Global Address Cleanse. When caching is enabled, we recommend for better performance that these options are set as follows:

Table 34: Best performance Global Address Cleanse options

Option	Recommended setting
Country Identification Mode	assign
Default Country	NONE

5.5.4 Ensure Workload Management and Resource Consumption

Ensure that in circumstances of limited memory or CPU resources that processes and system resources remain responsive.

Information on the specifics and procedures of workload management is found in the *SAP HANA Administration Guide*.

SAP HANA smart data integration takes advantage of this framework and allows you to better handle circumstances of limited resources. Workload management in SAP HANA allows you to optimize for your system. This framework also works within the limited memory or CPU resources, as you define them in the workload class and mapping.

For example, if the workload class sets "STATEMENT THREAD LIMIT = 5", then SAP HANA creates up to five instances per node or operation in parallel during the task plan execution.

If the workload class sets "STATEMENT MEMORY LIMIT = 2GB", but any of the nodes or operation in the task plan require more than 2GB of memory, then the task would fail with an error "[MEMORY_LIMIT_VIOLATION] Information about current memory composite-limit violation".

You must consider these options and constraints to create the best possible performance.

6 SQL and System Views Reference

This section contains information about SQL syntax and system views that can be used in SAP HANA smart data integration and SAP HANA smart data quality.

For complete information about all SQL statements and system views for SAP HANA and other SAP HANA options, see the *SAP HANA SQL and System Views Reference*.

6.1 SQL Statements

SAP HANA smart data integration and SAP HANA smart data quality support many SQL statements to allow you to do such tasks as create agents and adapters, administer your system, and so on.

The CREATE AUDIT POLICY statement creates a new audit policy, which can then be enabled and cause the specified audit actions to occur.

Smart Data Integration: CREATE REMOTE SOURCE [page 147]
The CREATE REMOTE SOURCE statement defines an external data source connected to SAP HANA database.

Smart Data Integration: CREATE REMOTE SUBSCRIPTION [page 148]
The CREATE REMOTE SUBSCRIPTION statement creates a remote subscription in SAP HANA to capture changes specified on the entire virtual table or part of a virtual table using a subquery.

Smart Data Integration: DROP ADAPTER [page 152]
The DROP ADAPTER statement removes an adapter from all locations.

Smart Data Integration: DROP AGENT [page 153]
The DROP AGENT statement removes an agent.

Smart Data Integration: DROP AGENT GROUP [page 154]
The DROP AGENT GROUP statement removes an agent clustering group.

Smart Data Integration: DROP REMOTE SUBSCRIPTION [page 155]
The DROP REMOTE SUBSCRIPTION statement drops an existing remote subscription.

Smart Data Integration: GRANT [page 156]
GRANT is used to grant privileges and structured privileges to users and roles. GRANT is also used to grant roles to users and other roles.

Smart Data Integration: PROCESS REMOTE SUBSCRIPTION EXCEPTION [page 158]
The PROCESS REMOTE SUBSCRIPTION EXCEPTION statement allows the user to indicate how an exception should be processed.

Smart Data Integration: SESSION_CONTEXT [page 159]
Returns the value of session_variable assigned to the current user.

Smart Data Integration: START TASK [page 160]
Starts a task.

6.1.1 Smart Data Integration: ALTER ADAPTER

The ALTER ADAPTER statement alters an adapter. Refer to CREATE ADAPTER for a description of the AT LOCATION clause.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
ALTER ADAPTER <adapter_name> [PROPERTIES <properties>]
ALTER ADAPTER <adapter_name> {ADD | REMOVE} LOCATION {DPSERVER | AGENT <agent_name>}
```

```
ALTER ADAPTER <adapter_name> REFRESH AT LOCATION {DPSERVER | AGENT <agent_name>}
```

Syntax Elements

```
<adapter_name> ::= <identifier>
<agent_name> ::= <identifier>
```

The name of the adapter to be altered and the agent name if the adapter is set up on the agent.

```
<properties> ::= <string_literal>
```

This specifies optional properties of the adapter such as display_name. When display_name is not specified, then adapter_name appears in the user interface.

Description

The ALTER ADAPTER statement alters an adapter. Refer to CREATE ADAPTER for a description of the AT LOCATION clause.

Examples

Add or remove an existing adapter at agent or Data Provisioning Server

Create two agents and an adapter at the first agent:

```
CREATE AGENT TEST_AGENT_1 PROTOCOL 'TCP' HOST 'test_host1' PORT 5050;
CREATE AGENT TEST_AGENT_2 PROTOCOL 'HTTP';
CREATE ADAPTER TEST_ADAPTER AT LOCATION AGENT TEST_AGENT_1;
```

Add an existing adapter TEST_ADAPTER to agent TEST_AGENT_2:

```
ALTER ADAPTER TEST_ADAPTER ADD LOCATION AGENT TEST_AGENT_2;
```

Remove an existing adapter TEST_ADAPTER from agent TEST_AGENT_2:

```
ALTER ADAPTER TEST_ADAPTER REMOVE LOCATION AGENT TEST_AGENT_2;
```

Add an existing adapter TEST_ADAPTER at the Data Provisioning Server:

```
ALTER ADAPTER TEST_ADAPTER ADD LOCATION DPSERVER;
```

Remove an existing adapter TEST_ADAPTER at Data Provisioning Server:

```
ALTER ADAPTER TEST_ADAPTER REMOVE LOCATION DPSERVER;
```

Refresh configuration and query optimization capabilities of an adapter

Read configuration and query optimization capabilities of an adapter from the adapter setup at the agent or Data Provisioning Server:

```
ALTER ADAPTER TEST_ADAPTER REFRESH AT LOCATION DPSERVER;
ALTER ADAPTER TEST_ADAPTER REFRESH AT LOCATION AGENT TEST_AGENT_2;
```

Update display name property of an adapter

Change display name for an adapter to 'My Custom Adapter':

```
ALTER ADAPTER TEST_ADAPTER PROPERTIES 'display_name=My Custom Adapter';
```

6.1.2 Smart Data Integration: ALTER AGENT

The ALTER AGENT statement changes an agent's host name and/or port and SSL property if it uses the TCP protocol. It can also assign an agent to an agent group.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
ALTER AGENT <agent_name> HOST <agent_hostname> [ PORT <agent_port_number> ]
            [ {ENABLE | DISABLE} SSL ]
ALTER AGENT <agent_name> PORT <agent_port_number> [ {ENABLE | DISABLE} SSL ]
ALTER AGENT <agent_name> {ENABLE | DISABLE} SSL
ALTER AGENT <agent_name> SET AGENT GROUP <agent_group_name>
ALTER AGENT <agent_name> UNSET AGENT GROUP <agent_group_name>
```

Syntax Elements

```
<agent_name> ::= <identifier>
```

The name of the agent to be altered.

```
<agent_hostname> ::= <string_literal>
<agent_port_number> ::= <integer_literal>  {ENABLE | DISABLE} SSL
```

Specifies if agent's TCP listener on the specified port uses SSL.

```
<agent_group_name> ::= <identifier>
```

The name of the agent clustering group to which the agent should be attached.

Description

The ALTER AGENT statement changes an agent's host name and/or port if it uses the TCP protocol. It can also assign an agent to an agent group.

Examples

Alter TEST_AGENT's hostname test_host and port to 5051, if it uses 'TCP' protocol

```
ALTER AGENT TEST_AGENT HOST 'test_host' PORT 5051;
```

Alter TEST_AGENT's hostname test_host, if it uses 'TCP' protocol

```
ALTER AGENT TEST_AGENT HOST 'test_host';
```

Alter TEST_AGENT's port to 5051, if it uses 'TCP' protocol

```
ALTER AGENT TEST_AGENT PORT 5051;
```

Assign TEST_AGENT to agent group TEST_GROUP

```
ALTER AGENT TEST_AGENT SET AGENT GROUP TEST_GROUP;
```

Remove TEST_AGENT from agent group TEST_GROUP

```
ALTER AGENT TEST_AGENT UNSET AGENT GROUP TEST_GROUP;
```

6.1.3 Smart Data Integration: ALTER REMOTE SOURCE

The ALTER REMOTE SOURCE statement modifies the configuration of an external data source connected to the SAP HANA database.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

The ALTER REMOTE SOURCE SQL statement is available for use in other areas of SAP HANA, not only SAP HANA smart data integration. Refer to the ALTER REMOTE SOURCE topic for complete information. The information below is specific to smart data integration functionality.

Syntax

```
ALTER REMOTE SOURCE <remote_source_name> <adapter_clause> [<credential_clause>]
ALTER REMOTE SOURCE <remote_source_name> { SUSPEND | RESUME } { CAPTURE |
DISTRIBUTION }
ALTER REMOTE SOURCE <remote_source_name> {CLEAR OBJECTS | REFRESH OBJECTS | CANCEL
REFRESH OBJECTS}
ALTER REMOTE SOURCE <remote_source_name> START LATENCY MONITORING
<latency_ticket_name> [INTERVAL <interval_in_seconds>]
ALTER REMOTE SOURCE <remote_source_name> STOP LATENCY MONITORING
<latency_ticket_name>
ALTER REMOTE SOURCE <remote_source_name> CLEAR LATENCY HISTORY
[<latency_ticket_name>]
```

Syntax Elements

Syntax elements specific to smart data integration are described as follows. For information about syntax elements that are not specific to smart data integration, refer to the ALTER REMOTE SOURCE topic.

```
<adapter_clause> ::= [ADAPTER <adapter_name> [AT LOCATION { DPSERVER | AGENT
<agent_name> | AGENT GROUP <agent_group_name>}] <configuration_clause>]
```

Adapter configuration.

```
<agent_name> ::= <identifier>
<agent_group_name> ::= <identifier>
```

Refer to CREATE ADAPTER for a description of the AT LOCATION clause.

```
<configuration_clause> ::= CONFIGURATION '<configuration_xml_string>'
```

XML-formatted configuration string for the remote source.

```
ALTER REMOTE SOURCE SUSPEND CAPTURE
```

Suspends the adapter and agent from reading any more changes from source system. This is helpful when the source system or SAP HANA is preparing for planned maintenance or an upgrade.

```
ALTER REMOTE SOURCE RESUME CAPTURE
```

Resumes the suspended adapter to read changed data from source system.

```
ALTER REMOTE SOURCE SUSPEND DISTRIBUTION
```

Suspends the application of real-time changes in SAP HANA tables but collects changed data from the source system.

```
ALTER REMOTE SOURCE RESUME DISTRIBUTION
```

Resumes applying real-time changes in SAP HANA tables.

```
CLEAR OBJECTS
```

Clears all the data received from the adapter for this remote source from HANA tables.

```
REFRESH OBJECTS
```

Starts building HANA dictionary tables that contain remote source objects.

```
CANCEL REFRESH OBJECTS
```

Cancels the long-running REFRESH background operation. This stops fetching records from the adapter but keeps the data received so far form the remote source on HANA tables.

```
START LATENCY MONITORING
```

Starts the collection of latency statistics one time or at regular intervals. The user specifies a target latency ticket in the monitoring view.

```
STOP LATENCY MONITORING
```

Stops the collection of latency statistics into the given latency ticket.

```
CLEAR LATENCY HISTORY
```

Clears the latency statistics (for either one latency ticket, or for the whole remote source, from the monitoring view.

Description

The ALTER REMOTE SOURCE statement modifies the configuration of an external data source connected to the SAP HANA database. Only database users with the object privilege ALTER for remote sources or the system privilege DATA ADMIN may alter remote sources.

Examples

The configuration clause must be a structured XML string that defines the settings for the remote source. For example, the following string configures a remote source for an Oracle database.

```
CONFIGURATION '<?xml version="1.0" encoding="UTF-8"?>
    <ConnectionProperties name="configurations">
        <PropertyGroup name="generic">
            <PropertyEntry name="instance_name">ora_inst</PropertyEntry>
            <PropertyEntry name="admin_port">12345</PropertyEntry>
```

```
            <PropertyEntry name="map_char_types_to_unicode">false</PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="database">
            <PropertyEntry name="cdb_enabled">false</PropertyEntry>
            <PropertyEntry name="pds_use_tnsnames">false</PropertyEntry>
            <PropertyEntry name="pds_host_name"><db_hostname></PropertyEntry>
            <PropertyEntry name="pds_port_number">1521</PropertyEntry>
            <PropertyEntry name="pds_database_name">ORCL</PropertyEntry>
            <PropertyEntry name="cdb_service_name"></PropertyEntry>
            <PropertyEntry name="pds_service_name"></PropertyEntry>
            <PropertyEntry name="pds_tns_filename"></PropertyEntry>
            <PropertyEntry name="pds_tns_connection"></PropertyEntry>
            <PropertyEntry name="cdb_tns_connection"></PropertyEntry>
            <PropertyEntry name="_pds_tns_connection_with_cdb_enabled"></
PropertyEntry>
            <PropertyEntry name="pds_byte_order"></PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="schema_alias_replacements">
            <PropertyEntry name="schema_alias"></PropertyEntry>
            <PropertyEntry name="schema_alias_replacement"></PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="security">
            <PropertyEntry name="pds_use_ssl">false</PropertyEntry>
            <PropertyEntry name="pds_ssl_sc_dn"></PropertyEntry>
            <PropertyEntry name="_enable_ssl_client_auth">false</PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="jdbc_flags">
            <PropertyEntry name="remarksReporting">false</PropertyEntry>
        </PropertyGroup>
        <PropertyGroup name="cdc">
            <PropertyGroup name="databaseconf">
                <PropertyEntry name="pdb_timezone_file"><timezone_file></
PropertyEntry>
                <PropertyEntry name="pdb_archive_path"></PropertyEntry>
                <PropertyEntry name="pdb_supplemental_logging_level">table</
PropertyEntry>
            </PropertyGroup>
            <PropertyGroup name="parallelscan">
                <PropertyEntry name="lr_parallel_scan">false</PropertyEntry>
                <PropertyEntry name="lr_parallel_scanner_count"></PropertyEntry>
                <PropertyEntry name="lr_parallel_scan_queue_size"></PropertyEntry>
                <PropertyEntry name="lr_parallel_scan_range"></PropertyEntry>
            </PropertyGroup>
            <PropertyGroup name="logreader">
                <PropertyEntry name="skip_lr_errors">false</PropertyEntry>
                <PropertyEntry name="lr_max_op_queue_size">1000</PropertyEntry>
                <PropertyEntry name="lr_max_scan_queue_size">1000</PropertyEntry>
                <PropertyEntry name="lr_max_session_cache_size">1000</PropertyEntry>
                <PropertyEntry name="scan_fetch_size">10</PropertyEntry>
                <PropertyEntry name="pdb_dflt_column_repl">true</PropertyEntry>
                <PropertyEntry name="pdb_ignore_unsupported_anydata">false</
PropertyEntry>
                <PropertyEntry name="pds_sql_connection_pool_size">15</
PropertyEntry>
                <PropertyEntry name="pds_retry_count">5</PropertyEntry>
                <PropertyEntry name="pds_retry_timeout">10</PropertyEntry>
            </PropertyGroup>
        </PropertyGroup>
</ConnectionProperties>'
```

6.1.4 Smart Data Integration: ALTER REMOTE SUBSCRIPTION

The ALTER REMOTE SUBSCRIPTION statement allows the QUEUE command to initiate real-time data processing, and the DISTRIBUTE command applies the changes.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
ALTER REMOTE SUBSCRIPTION [<schema_name>.]<subscription_name>
< QUEUE | DISTRIBUTE | RESET >
```

Syntax Elements

```
<subscription_name> ::= <identifier>
```

The name of the remote subscription.

Description

The ALTER REMOTE SUBSCRIPTION statement allows the QUEUE command to initiate real-time data processing, and the DISTRIBUTE command applies the changes. Typically, the initial load of data is preceded by QUEUE command. The DISTRIBUTE command is used when initial load completes. The RESET command can be used to reset the real-time process to start from the initial load again.

Example

Capture changes from a virtual table to an SAP HANA table.

```
CREATE AGENT TEST_AGENT PROTOCOL 'TCP' HOST 'test_host1' PORT 5050;
CREATE ADAPTER 'DB2ECCAdapter' AT LOCATION AGENT TEST_AGENT;
CREATE REMOTE SOURCE MYECC ADAPTER 'DB2ECCAdapter' CONFIGURATION
'<configuration_xml>' AT LOCATION AGENT TEST_AGENT;
CREATE VIRTUAL TABLE MARA_VT AT MYECC."<NULL>"."<NULL>".MARA;
CREATE COLUMN TABLE TGT_MARA LIKE MARA_VT;
CREATE REMOTE SUBSCRIPTION TEST_SUB ON MARA_VT TARGET TABLE TGT_MARA;
ALTER REMOTE SUBSCRIPTION TEST_SUB QUEUE;
```

Perform initial load of data using INSERT-SELECT or a TASK.

```
INSERT INTO TGT_MARA SELECT * FROM MARA_VT;
ALTER REMOTE SUBSCRIPTION TEST_SUB DISTRIBUTE;
```

Now insert or update a material record in ECC system and see it updated to TGT_MARA table in SAP HANA. Reset the real-time process and restart the load.

```
ALTER REMOTE SUBSCRIPTION TEST_SUB RESET;
ALTER REMOTE SUBSCRIPTION TEST_SUB QUEUE;
```

Perform initial load of data using INSERT-SELECT or a TASK.

```
INSERT INTO TGT_MARA SELECT * FROM MARA_VT;
ALTER REMOTE SUBSCRIPTION TEST_SUB DISTRIBUTE;
```

6.1.5 Smart Data Integration: CANCEL TASK

Cancels a task that was started with START TASK.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
CANCEL TASK <task_execution_id> [WAIT <wait_time_in_seconds>]
```

Syntax Elements

```
<task_execution_id> ::= <unsigned_integer>
```

Specifies the task execution ID to cancel. See the START TASK topic for more information about TASK_EXECUTION_ID.

```
 <wait_time_in_seconds> ::= <identifier>
```

Number of seconds to wait for the task to cancel before returning from the command.

Description

Cancels a task that was started with START TASK.

The default behavior is for the CANCEL TASK command to return after sending the cancel request. Optionally, a WAIT value can be specified where the command will wait for the task to actually cancel before returning. If the command has waited the specified amount of time, then the CANCEL TASK will error out with the error code 526 (request to cancel task was sent but task did not cancel before timeout was reached).

> **Note**
>
> If the WAIT value is 0, the command returns immediately after sending the cancel request, as it would if no WAIT value were entered.

Examples

Assuming that a TASK performTranslation was already started using START TASK and has a task execution ID of 255, it would be cancelled using the following commands. The behavior is the same for the following two cases:

```
CANCEL TASK 255;
```

```
CANCEL TASK 255 WAIT 0;
```

Assuming that a TASK performTranslation was already started using START TASK and has a task execution id of 256 and the user wants to wait up to 5 seconds for the command to cancel, it would be cancelled using the following command:

```
CANCEL TASK 256 WAIT 5;
```

If the task was able to cancel within 5 seconds, the CANCEL TASK will return as a success. If it didn't cancel within 5 seconds, then the return will be the error code 526.

SQL Script

You can call CANCEL TASK within the SQL Script CREATE PROCEDURE. Refer to the SAP HANA SQL Script Reference for complete details about CREATE PROCEDURE.

```
CREATE PROCEDURE "CANCEL_TASK"."CANCEL_MY_TASK"(in exec_id INT)
LANGUAGE SQLSCRIPT AS
BEGIN
     CANCEL TASK :exec_id;
END;
```

CANCEL TASK is not supported in:

- Table UDF
- Scalar UDF

- Trigger
- Read-only procedures

Related Information

Smart Data Integration: START TASK [page 160]

6.1.6 Smart Data Integration: CREATE ADAPTER

The CREATE ADAPTER statement creates an adapter that is deployed at the specified location.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
CREATE ADAPTER <adapter_name> [PROPERTIES <properties>] AT LOCATION
{DPSERVER | AGENT <agent_name>}
```

Syntax Elements

```
<adapter_name> ::= <identifier> <agent_name> ::= <identifier>
```

The name of the adapter to be created and the agent name if the adapter is set up on the agent.

```
<properties> ::= <string_literal>
```

Specifies optional properties of the adapter such as display_name. When display_name is not specified, then adapter_name displays in the user interface.

```
AT LOCATION DPSERVER
```

The adapter runs inside the Data Provisioning Server process in SAP HANA.

```
AT LOCATION AGENT<agent_name>
```

The adapter runs inside the agent that is set up outside of SAP HANA.

Description

The CREATE ADAPTER statement creates an adapter that is deployed at the specified location. The adapter must be set up on the location prior to running this statement. When the statement is executed, the Data Provisioning Server contacts the adapter to retrieve its configuration details such as connection properties and query optimization capabilities.

Examples

Create adapter at Data Provisioning Server

Create an adapter TEST_ADAPTER running in the Data Provisioning Server.

```
CREATE ADAPTER TEST_ADAPTER AT LOCATION DPSERVER;
```

Create adapter at agent

Create an agent with name TEST_AGENT.

```
CREATE AGENT TEST_AGENT PROTOCOL 'TCP' HOST 'test_host' PORT 5050;
```

Create an adapter TEST_ADAPTER on agent TEST_AGENT.

```
CREATE ADAPTER TEST_ADAPTER AT LOCATION AGENT TEST_AGENT;
```

6.1.7 Smart Data Integration: CREATE AGENT

The CREATE AGENT statement registers connection properties of an agent that is installed on another host.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
CREATE AGENT <agent_name> PROTOCOL { {'HTTP'} |
{'TCP' HOST <agent_hostname> PORT <agent_port_number>} [AGENT GROUP
<agent_group_name>] [{ENABLE | DISABLE} SSL] }
```

Syntax Elements

```
<agent_name> ::= <identifier>
```

The name of the agent to be created and its protocol.

```
PROTOCOL 'HTTP'
```

Agent uses HTTP protocol for communication with DP server. Use this protocol when the SAP HANA database is on the cloud.

```
PROTOCOL 'TCP' HOST <agent_hostname> PORT <agent_port_number>
```

Agent uses TCP protocol and listens on the specified port to receive requests from DP server. Use this protocol when the SAP HANA database can connect to agent's TCP port.

```
{ENABLE | DISABLE} SSL
```

Specifies if agent's TCP listener on the specified port uses SSL.

```
<agent_hostname> ::= <string_literal>
<agent_port_number> ::= <integer_literal>
```

DP server connects to the agent listening on the specified hostname and port. Use this protocol when the SAP HANA database is on-premise.

```
<agent_group_name> ::= <identifier>
```

The name of the agent clustering group to which the agent should belong.

Description

The CREATE AGENT statement registers connection properties of an agent that is installed on another host. The DP server and agent use these connection properties when establishing communication channel.

Examples

Create agent with TCP protocol

Create an agent TEST_AGENT running on test_host and port 5050.

```
CREATE AGENT TEST_AGENT PROTOCOL 'TCP' HOST 'test_host' PORT 5050;
```

Create agent with HTTP protocol
Create an agent TEST_AGENT that uses HTTP. `CREATE AGENT TEST_AGENT PROTOCOL 'HTTP';`

Create agent with HTTP protocol in an agent group
Create an agent TEST_AGENT that uses HTTP and belongs to agent clustering group TEST_GROUP. `CREATE AGENT TEST_AGENT PROTOCOL 'HTTP' AGENT GROUP TEST_GROUP;`

6.1.8 Smart Data Integration: CREATE AGENT GROUP

The CREATE AGENT GROUP statement creates an agent clustering group to which individual agents can be assigned.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
CREATE AGENT GROUP <agent_group_name>
```

Syntax Elements

```
<agent_group_name> ::= <identifier>
```

The name of the agent group to create.

Description

The CREATE AGENT GROUP statement creates an agent clustering group to which individual agents can be assigned. An agent group can be used instead of a single agent to provide fail-over capabilities.

Examples

Create agent group named TEST_GROUP

```
CREATE AGENT GROUP TEST_GROUP;
```

Related Information

Smart Data Integration: ALTER AGENT [page 134]
Smart Data Integration: CREATE AGENT [page 143]

6.1.9 Smart Data Integration: CREATE AUDIT POLICY

The CREATE AUDIT POLICY statement creates a new audit policy, which can then be enabled and cause the specified audit actions to occur.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

The CREATE AUDIT POLICY SQL statement is available for use in other areas of SAP HANA, not only SAP HANA smart data integration. Refer to the CREATE AUDIT POLICY topic for complete information. The information below is specific to smart data integration functionality.

Syntax

Refer to the *SAP HANA SQL and System Views Reference* for complete information about CREATE AUDIT POLICY syntax.

Syntax Elements

Syntax elements specific to smart data integration are described below. For information about syntax elements that are not specific to smart data integration, refer to the *SAP HANA SQL and System Views Reference*.

```
<audit_action_name> ::= CREATE AGENT                |
                        ALTER AGENT                 |
                        DROP AGENT                  |
                        CREATE ADAPTER              |
                        ALTER ADAPTER               |
```

```
                        DROP ADAPTER                  |
                        CREATE REMOTE SUBSCRIPTION    |
                        ALTER REMOTE SUBSCRIPTION     |
                        DROP REMOTE SUBSCRIPTION      |
                        PROCESS REMOTE SUBSCRIPTION EXCEPTION
```

Audit Action Name	Group Number	Audit Operation
CREATE AGENT	17	Registering a Data Provisioning Agent
ALTER AGENT	17	Altering a Data Provisioning Agent registration
DROP AGENT	17	Dropping a Data Provisioning Agent registration
CREATE ADAPTER	17	Registering a Data Provisioning Adapter
ALTER ADAPTER	17	Altering the registration of a Data Provisioning Adapter
DROP ADAPTER	17	Dropping the registration of a Data Provisioning Adapter
CREATE REMOTE SUBSCRIPTION	17	Creating a subscription to a remote source
ALTER REMOTE SUBSCRIPTION	17	Altering a subscription to a remote source
DROP REMOTE SUBSCRIPTION	17	Dropping a subscription to a remote source
PROCESS REMOTE SUBSCRIPTION EXCEPTION	17	Processing exceptions raised by a subscribed remote source

Description

The CREATE AUDIT POLICY statement creates a new audit policy. This audit policy can then be enabled and cause the auditing of the specified audit actions to occur.

6.1.10 Smart Data Integration: CREATE REMOTE SOURCE

The CREATE REMOTE SOURCE statement defines an external data source connected to SAP HANA database.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

The CREATE REMOTE SOURCE SQL statement is available for use in other areas of SAP HANA, not only SAP HANA smart data integration. Refer to the CREATE REMOTE SOURCE topic for complete information. The information below is specific to smart data integration functionality.

Syntax

Refer to the *SAP HANA SQL and System Views Reference* for complete information about CREATE REMOTE SOURCE syntax.

Syntax Elements

Syntax elements specific to smart data integration are described below. For information about syntax elements that are not specific to smart data integration, refer to the *SAP HANA SQL and System Views Reference*.

```
<adapter_clause> ::= ADAPTER <adapter_name>
     [AT LOCATION {DPSERVER | AGENT <agent_name> | AGENT GROUP <agent_group_name>} ]
     CONFIGURATION <connection_info_string>
```

Adapter configuration.

```
<agent_name> ::= <identifier>
<agent_group_name> ::= <identifier>
```

Refer to CREATE ADAPTER for description on AT LOCATION.

```
<object_privilege> ::=  PROCESS REMOTE SUBSCRIPTION EXCEPTION
```

Description

The CREATE REMOTE SOURCE statement defines an external data source connected to SAP HANA database. Only database users having the system privilege CREATE SOURCE or DATA ADMIN are allowed to add a new remote source.

6.1.11 Smart Data Integration: CREATE REMOTE SUBSCRIPTION

The CREATE REMOTE SUBSCRIPTION statement creates a remote subscription in SAP HANA to capture changes specified on the entire virtual table or part of a virtual table using a subquery.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
CREATE REMOTE SUBSCRIPTION [<schema_name>.]<subscription_name> { {ON
[<schema_name>.]<virtual_table_name>} | {AS (<subquery>)} } { TARGET {TABLE
[<schema_name>].<table_name> } | {TASK
[<schema_name>].<task_name>[(<var_list>)] [PROCEDURE
PARAMETERS(<param_list>)} | {PROCEDURE
[<schema_name>].<proc_name>[(<param_list>)]}
change type column <column_name>
change time column <column_name>
change seq column <column_name>
<load_behavior>
}
```

Syntax Elements

```
<subscription_name> ::= <identifier>
```

The name of the remote subscription.

```
<var_list> ::= <start_task_var>[{, <start_task_var>}...]
```

Specifies one or more start task variables.

```
<start_task_var> ::= <var_name> => <var_value>
```

Specifies the name and value for a start task variable. A task plan can contain variables that allow for dynamic replacement of task plan variables. Variable values provided in this section will be used at runtime (for example, when executing the task using START TASK).

```
<var_name> ::= <identifier>
```

Name of variable that was defined within the task plan.

```
<var_value> ::= <string_literal>
```

Value that should be used in place of the variable name specified when executing the task.

```
<param_list> ::= <start_task_param>[{, <start_task_param>}...]
```

Specifies one or more start task parameters.

```
<start_task_param> ::= <identifier>
```

Task parameters.

If the task uses table types for input and/or output, then the task expects actual table, virtual table, or view names at runtime. These actual tables, virtual tables, or view names are specified as task parameters. Depending on the type of remote subscription being created, the task parameters may or may not need actual table, virtual table, or view names for specific parameters (see below for more details).

```
change type column <column_name>
```

```
change time column <column_name>
change seq column <column_name>
<load_behavior>
```

For a target table that logs the loading history, these parameters specify the target column names that will show the change type and corresponding timestamp for each operation. The `change type column <column_name>` displays I, U, or D for insert, upsert, or delete. In the case when multiple operations of the same type occur on the same source row with the same timestamp (because the operations are in the same transaction), use the `change seq column <column_name>`, which adds an incremental digit to distinguish the operations.

The load behavior options are:

`upsert`: INSERT and UPDATE apply as is, DELETE converts to UPDATE

`insert`: INSERT applies as is, UPDATE and DELETE convert to INSERT

The following example is for UPSERT for a remote subscription called `user.subscription` on a source table called `SQLServer_dbo.table`. The target table`user.table` includes a column called CHANGE_TYPE (with a data type of VARCHAR or NVARCHAR) and a column CHANGE_TIME (with a data type of TIMESTAMP).

```
create remote subscription user.subscription
      on "user"."SQLServer_dbo.table"
      target table user.table
      change type column "CHANGE_TYPE"
      change time column "CHANGE_TIME"
      upsert;
```

The following example for INSERT is for the same remote subscription and includes the CHANGE_TIME column.

```
create remote subscription user.subscription
      on "user"."SQLServer_dbo.table"
      target table user.table
      change type column "CHANGE_TYPE"
      change time column "CHANGE_TIME"
      change seq column "CHANGE_SEQUENCE"
      insert;
```

Description

The CREATE REMOTE SUBSCRIPTION statement creates a remote subscription in SAP HANA to capture changes specified on the entire virtual table or part of a virtual table using a subquery. The changed data can be applied to a SAP HANA target table or passed to a TASK or PROCEDURE if the changes require transformation. The owner of the remote subscription must have the following privileges:

- SELECT privilege on tables specified in the ON or AS `<subquery>` clauses
- INSERT, UPDATE, DELETE privileges on the target table
- EXECUTE privilege on the stored procedure
- START TASK privilege on the task

> **Note**
>
> If you create a remote subscription using the CREATE REMOTE SUBSCRIPTION SQL statement, use **`technical user`** for the *Credentials Mode* parameter when creating a remote source.

Remote subscription for TARGET TASK using ON Clause

```
CREATE REMOTE SUBSCRIPTION [<schema_name>.]<subscription_name>
ON [<schema_name>.]<virtual_table_name>
TARGET TASK [<schema_name>].<task_name>[(<var_list>)] [PROCEDURE
PARAMETERS(<param_list>)]
```

<param_list> must contain one of the parameters as [<schema_name>.]<virtual_table_name>. This parameter must be the same schema and virtual table name as specified in the ON clause. Only one parameter in <param_list> can be a virtual table.

Each parameter in <param_list> is used in comparing its columns with columns for the corresponding table type defined in the task plan. Hence, the order of parameters in <param_list> must match the order of table types defined in the task plan for input and output sources.

The task plan table type corresponding to the procedure parameter [<schema_name>.]<virtual_table_name> must have the same columns (excluding _OP_CODE and _COMMIT_TIMESTAMP). This table type must have _OP_CODE as the last but one column and _COMMIT_TIMESTAMP as the last column.

Remote subscription for TARGET TASK using AS Clause

```
CREATE REMOTE SUBSCRIPTION [<schema_name>.]<subscription_name>
AS <subquery>
TARGET TASK [<schema_name>].<task_name>[(<var_list>)] [PROCEDURE
PARAMETERS(<param_list>)]
```

<param_list> must contain one of the parameters as table type and this table type (schema and name) must be the same as the one defined in the task plan. This table type must also have the same columns as being output by the subquery (excluding _OP_CODE and _COMMIT_TIMESTAMP). This table type must have _OP_CODE as the last but one column and _COMMIT_TIMESTAMP as the last column. Only one parameter in <param_list> can be a table type.

Each parameter in <param_list> is used in comparing its columns with columns for the corresponding table type defined in the task plan. Hence the order of parameters in <param_list> must match the order of table types defined in task plan for input and output sources.

Example

Create remote subscription from virtual table to real-time task

Create a remote subscription on a virtual table and apply changes using real-time task.

```
create schema "IM_SERVICES";
drop REMOTE SOURCE "OracleAdapter" cascade;
CREATE REMOTE SOURCE "OracleAdapter" ADAPTER "OracleAdapter" AT LOCATION dpserver
CONFIGURATION '' WITH CREDENTIAL TYPE 'PASSWORD' USING '';
DROP TABLE "SYSTEM"."VT_EMPLOYEE_PK_TABLE";
CREATE VIRTUAL TABLE "SYSTEM"."VT_EMPLOYEE_PK_TABLE" AT
"OracleAdapter"."<NULL>"."<NULL>"."employee_pk_table";
DROP TYPE "IM_SERVICES"."TT_PARAM_IN";
DROP TYPE "IM_SERVICES"."TT_PARAM_OUT";
CREATE TYPE "IM_SERVICES"."TT_PARAM_IN"   AS TABLE ("empno" integer, "deptid"
integer, "empname" VARCHAR(200), "salary" decimal(28,7), "bonus" double,
"_OP_CODE" VARCHAR(1),"_COMMIT_TIMESTAMP" SECONDDATE);
CREATE TYPE "IM_SERVICES"."TT_PARAM_OUT"  AS TABLE ("empno" integer, "deptid"
integer, "empname" VARCHAR(200), "salary" decimal(28,7), "bonus" double);
DROP TABLE "IM_SERVICES"."T_OUT";
CREATE COLUMN TABLE "IM_SERVICES"."T_OUT" LIKE "IM_SERVICES"."TT_PARAM_OUT" ;
DROP TASK "IM_SERVICES"."TSKM_RT_VAR";
DROP REMOTE SUBSCRIPTION "IM_SERVICES"."RSUB_VAR";
CREATE REMOTE SUBSCRIPTION "IM_SERVICES"."RSUB_VAR"
AS (select "empno","deptid","empname","salary","bonus"  from
"SYSTEM"."VT_EMPLOYEE_PK_TABLE")
TARGET TASK "IM_SERVICES"."TSKM_RT_VAR" ("expr_var01_in1" => '100',
"expr_var02_in2" => 'upper(''walkerIN'')')
PROCEDURE PARAMETERS ( "IM_SERVICES"."TT_PARAM_IN", "IM_SERVICES"."T_OUT");
DROP REMOTE SUBSCRIPTION "IM_SERVICES"."RSUB_VAR";
CREATE REMOTE SUBSCRIPTION "IM_SERVICES"."RSUB_VAR"
ON "SYSTEM"."VT_EMPLOYEE_PK_TABLE"
TARGET TASK "IM_SERVICES"."TSKM_RT_VAR" ("expr_var01_in1" => '100',
"expr_var02_in2" => 'upper(''walkerIN'')')
PROCEDURE PARAMETERS ( "SYSTEM"."VT_EMPLOYEE_PK_TABLE", "IM_SERVICES"."T_OUT");
SELECT * FROM "SYS"."REMOTE_SUBSCRIPTIONS_";
truncate table "IM_SERVICES"."T_OUT";
alter remote subscription "IM_SERVICES"."RSUB_VAR" queue;
alter remote subscription "IM_SERVICES"."RSUB_VAR" DISTRIBUTE;
```

6.1.12 Smart Data Integration: DROP ADAPTER

The DROP ADAPTER statement removes an adapter from all locations.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
DROP ADAPTER <adapter_name> [<drop_option>]
```

Syntax Elements

```
<adapter_name> ::= <identifier>
```

The name of the adapter to be dropped.

```
<drop_option> ::= CASCADE | RESTRICT
```

When <drop_option> is not specified, a restrict drop will be performed.

```
CASCADE
```

Drops the adapter and dependent objects.

Description

The DROP ADAPTER statement removes an adapter from all locations.

Example

Create two agents and an adapter at both the agents.

```
CREATE AGENT TEST_AGENT_1 PROTOCOL 'TCP' HOST 'test_host1' PORT 5050;
CREATE AGENT TEST_AGENT_2 PROTOCOL 'HTTP';
CREATE ADAPTER TEST_ADAPTER AT LOCATION AGENT TEST_AGENT_1;
ALTER ADATER TEST_ADAPTER ADD LOCATION AGENT TEST_AGENT_2;
--Drop adapter TEST_ADAPTER.
DROP ADAPTER TEST_ADAPTER;
```

6.1.13 Smart Data Integration: DROP AGENT

The DROP AGENT statement removes an agent.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
DROP AGENT <agent_name> [<drop_option>]
```

Syntax Elements

```
<agent_name> ::= <identifier>
```

The name of the agent to be dropped.

```
<drop_option> ::= CASCADE | RESTRICT
```

When <drop_option> is not specified, a restrict drop is performed.

```
CASCADE
```

Drops the agent and its dependent objects.

```
RESTRICT
```

Drops the agent only if it does not have any dependent objects.

Description

The DROP AGENT statement removes an agent.

Example

Create an agent TEST_AGENT and adapter CUSTOM_ADAPTER on the agent. Make sure that the custom adapter is setup on the agent.

```
CREATE AGENT TEST_AGENT PROTOCOL 'TCP' HOST 'test_host' PORT 5050;
CREATE ADAPTER CUSTOM_ADAPTER AT LOCATION AGENT TEST_AGENT;
```

Drop the agent called TEST_AGENT.

```
DROP AGENT TEST_AGENT;
```

6.1.14 Smart Data Integration: DROP AGENT GROUP

The DROP AGENT GROUP statement removes an agent clustering group.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
DROP AGENT GROUP <agent_group_name>
```

Syntax Elements

```
<agent_group_name> ::= <identifier>
```

The name of the agent group to be dropped.

Description

The DROP AGENT GROUP statement removes an agent clustering group. All dependent objects must be removed before an agent clustering group can be dropped.

Example

Create an agent group TEST_GROUP.

```
CREATE AGENT GROUP TEST_GROUP;
```

Drop the agent called TEST_GROUP.

```
DROP AGENT GROUP TEST_GROUP;
```

6.1.15 Smart Data Integration: DROP REMOTE SUBSCRIPTION

The DROP REMOTE SUBSCRIPTION statement drops an existing remote subscription.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
DROP SUBSCRIPTION [<schema_name>.]<subscription_name>
```

Syntax Elements

```
<subscription_name> ::= <identifier>
```

The name of the remote subscription.

Description

The DROP REMOTE SUBSCRIPTION statement drops an existing remote subscription. If the remote subscription is actively receiving changes from source table, then a RESET command is automatically called before dropping it.

Example

Drop the remote subscription test_sub.

```
DROP REMOTE SUBSCRIPTION TEST_SUB;
```

6.1.16 Smart Data Integration: GRANT

GRANT is used to grant privileges and structured privileges to users and roles. GRANT is also used to grant roles to users and other roles.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

The GRANT SQL statement is available for use in other areas of SAP HANA, not only SAP HANA smart data integration. Refer to the GRANT topic for complete information. The information below is specific to smart data integration functionality.

Syntax

Refer to the GRANT topic for complete information about GRANT syntax.

Syntax Elements

Syntax elements specific to smart data integration are described below. For information about syntax elements that are not specific to smart data integration, refer to the GRANT topioc.

```
<system_privilege> ::= ADAPTER ADMIN | AGENT ADMIN
```

System privileges are used to restrict administrative tasks. The table below describes the supported system privileges.

System Privilege	Privilege Purpose
ADAPTER ADMIN	Controls the execution of the following adapter-related commands: CREATE ADAPTER, DROP ADAPTER and ALTER ADAPTER. Also allows access to ADAPTERS and ADAPTER_LOCATIONS system views.
AGENT ADMIN	Controls the execution of the following agent-related commands: CREATE AGENT, DROP AGENT and ALTER AGENT. Also allows access to AGENTS and ADAPTER_LOCATIONS system views.

```
<source_privilege> ::= CREATE REMOTE SUBSCRIPTION            |
                       PROCESS REMOTE SUBSCRIPTION EXCEPTION |
```

Source privileges are used to restrict the access and modifications of a source entry.

Source Privilege	Privilege Purpose
CREATE REMOTE SUBSCRIPTION	This privilege allows the creation of remote subscriptions executed on this source entry. Remote subscriptions are created in a schema and point to a virtual table or SQL on tables to capture changed data.
PROCESS REMOTE SUBSCRIPTION EXCEPTION	This privilege allows processing exceptions on this source entry. Exceptions that are relevant for all remote subscriptions are created for a remote source entry.

```
<object_privilege> ::=  AGENT MESSAGING          |
                  PROCESS REMOTE SUBSCRIPTION EXCEPTION
```

Object privileges are used to restrict the access and modifications on database objects. Database objects are tables, views, sequences, procedures, and so on. The table below describes the supported object privileges.

Object Privilege	Privilege Purpose	Command Types
AGENT MESSAGING	Authorizes the user with which the agent communicates with the data provisioning server using HTTP protocol. For example: `GRANT AGENT MESSAGING ON AGENT "CloudAgent" TO HANA_USER;`	DDL

Object Privilege	Privilege Purpose	Command Types
PROCESS REMOTE SUBSCRIPTION EXCEPTION	Authorizes processing exceptions of a remote subscription.	DDL

Not all object privileges are applicable to all kinds of database objects. To learn which object types allow which privilege to be used, see the table below.

Privilege	Schema	Table	View	Sequence	Function / Procedure	Remote Subscription	Agent
AGENT MESSAGING	--	--	--	--	--	--	YES
PROCESS REMOTE SUBSCRIPTION EXCEPTION	--	--	--	--	--	YES	--

6.1.17 Smart Data Integration: PROCESS REMOTE SUBSCRIPTION EXCEPTION

The PROCESS REMOTE SUBSCRIPTION EXCEPTION statement allows the user to indicate how an exception should be processed.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
PROCESS REMOTE SUBSCRIPTION EXCEPTION <exception_id> {RETRY|IGNORE}
```

Syntax Elements

```
<exception_id> ::= <integer_literal>
```

The exception ID for remote subscription or remote source.

RETRY Indicates to retry the current failed operation. If the failure is due to opening a connection to a remote source, then the connection is established. If the failure happens when applying changed data to a target table, then the RETRY operation retries the transaction again on the target table.

IGNORE Indicates to ignore the current failure. If the failure happens when applying changed data to a target table, then the IGNORE operation skips the current transaction and proceeds with the next transaction. The exception is cleared.

Description

The PROCESS REMOTE SUBSCRIPTION EXCEPTION statement allows the user to indicate how an exception should be processed.

Example

Ignore exception 101.

```
PROCESS REMOTE SUBSCRIPTION EXCEPTION 101 IGNORE;
```

6.1.18 Smart Data Integration: SESSION_CONTEXT

Returns the value of session_variable assigned to the current user.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

SESSION_CONTEXT is available for use in other areas of SAP HANA, not only SAP HANA smart data integration. Refer to the SESSION_CONTEXT topic for complete information. The information below is specific to smart data integration functionality.

Syntax

```
SESSION_CONTEXT(session_variable)
```

Description

A predefined session variables that is set by the server and is read-only (cannot be SET or UNSET) is 'TASK_EXECUTION_ID'.

6.1.19 Smart Data Integration: START TASK

Starts a task.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Syntax

```
START TASK <task_name> [ASYNC] [(<var_list>)] [PROCEDURE PARAMETERS(<param_list>)]
```

Syntax Elements

```
<task_name> ::= [<schema_name>.]<identifier>
<schema_name> ::= <identifier>
```

The identifier of the task to be called, with optional schema name.

```
<var_list> ::= <start_task_var>[{, <start_task_var>}...]
```

Specifies one or more start task variables. Variables passed to a task are scalar constants. Scalar parameters are assumed to be NOT NULL.

```
<start_task_var> ::= <var_name> => <var_value>
```

Specifies the name and value for a start task variable. A task can contain variables that allow for dynamic replacement of task plan parameters. This section is where, at run time during START TASK, the values that should be used for those variables can be provided.

```
<var_name> ::= <identifier>
```

Name of variable that was defined.

```
<var_value> ::= <string_literal>
```

Value that should be used in place of the variable name specified when executing the task.

```
<param_list> ::= <start_task_param>[{, <start_task_param>}...]
```

Specifies one or more start task parameters.

```
 <start_task_param> ::= <identifier>
```

Task parameters. If the task uses table types for input and/or output, then those need to be specified within this section. For more information about these data types, see BNF Lowest Terms Representations and Data Types in the Notation topic.

Parameters are implicitly defined as either IN or OUT, as inferred from the task plan. Arguments for IN parameters could be anything that satisfies the schema of the input table type (for example, a table variable internal to the procedure, or a temporary table). The actual value passed for tabular OUT parameters can be, for example, '?', a physical table name, or a table variable defined inside the procedure.

Description

Starts a task.

START TASK when executed by the client the syntax behaves in a way consistent with the SQL standard semantics, e.g. Java clients can call a procedure using a JDBC CallableStatement. Scalar output variables are a scalar value that can be retrieved from the callable statement directly.

> **Note**
>
> Unquoted identifiers are implicitly treated as uppercase. Quoting identifiers will respect capitalization and allow for using white spaces which are normally not allowed in SQL identifiers.

Examples

The TASK performTranslation was already created, and the task plan has two table type input parameters and a single table type output parameter. You call the performTranslation task passing in the table types to use for execution.

```
START TASK performTranslation PROCEDURE PARAMETERS (in1, in2, out1) ;
```

The TASK performCleansing was already created, and the task plan used had a variable defined as CASING.

You call the performCleansing task using a variable that will set the CASING variable to have the value of 'mixed'.

```
START TASK performCleansing (CASING => 'mixed');
```

The TASK performCleansing was already created, and the task plan used had a variable defined as CASING. You call the performCleansing task using a variable that will set the CASING variable to have the value of 'mixed'.

By adding the ASYNC text, execution of the task will also be asynchronous.

```
START TASK performCleansing ASYNC (CASING => 'mixed');
```

SQL Script

You can call START TASK within the SQL Script CREATE PROCEDURE. Refer to the SAP HANA SQL Script Reference for complete details about CREATE PROCEDURE.

<proc_sql> now includes <start_task>:

```
<proc_sql> ::= <subquery>          |
               <select_into_stmt>  |
               <insert_stmt>       |
               <delete_stmt>       |
               <update_stmt>       |
               <replace_stmt>      |
               <call_stmt>         |
               <create_table>      |
               <drop_table>        |
               <start_task>
```

START TASK is not supported in:

- Table UDF
- Scalar UDF
- Trigger
- Read-only procedures

TASK_EXECUTION_ID session variable

The TASK_EXECUTION_ID session variable provides a unique task execution ID. Knowing the proper task execution ID is critical for various pieces of task functionality including querying for side-effect information and task processing status, and canceling a task.

TASK_EXECUTION_ID is a read-only session variable. Only the internal start task code updates the value.

The value of TASK_EXECUTION_ID will be set during the START TASK command execution. In the case of asynchronous execution (START TASK ASYNC), the value is updated before the command returns so it is available before the actual task has finished asynchronously running. If the execution of START TASK was successful, then the value is updated to the unique execution ID for that START TASK execution. If the execution of START TASK was unsuccessful, then the TASK_EXECUTION_ID variable will be set back to the state as if no START TASK was run.

The users can obtain the value of TASK_EXECUTION_ID by using either of the following:

- The already existing SESSION_CONTEXT() function. If this function is used and if no tasks have been run or a task was run and it was unsuccessful, then a NULL value will be returned.
- The M_SESSION_CONTEXT monitoring view. This would need to be queried using a KEY value of "TASK_EXECUTION_ID". If no row exists with that key, then that means that the session variable hasn't been set (no tasks run or last task execution was unsuccessful).

> **Note**
>
> Session variables are string values. The user needs to cast appropriately based on how they want to use the value.

Table 35: Examples

Action	SQL
Obtain the last task execution ID	`SELECT SESSION_CONTEXT('TASK_EXECUTION_ID') FROM dummy;`
See monitoring information for the last task that was executed (with type casting)	`SELECT * FROM M_TASKS WHERE TASK_EXECUTION_ID = CAST (SESSION_CONTEXT('TASK_EXECUTION_ID') AS BIGINT);`
Cancel the last task that was executed (with type casting)	`CANCEL TASK CAST(SESSION_CONTEXT('TASK_EXECUTION_ID') AS BIGINT);`

6.2 System Views

System views allow you to query for various information about the system state using SQL commands. The results appear as tables.

System views are located in the SYS schema. In a system with tenant databases, every database has a SYS schema with system views that contain information about that database only. In addition, the system database has a further schema, SYS_DATABASES, which contains views for monitoring the system as a whole. The views in the SYS_DATABASES schema provide aggregated information from a subset of the views available in the SYS schema of all tenant databases in the system. These union views have the additional column DATABASE_NAME to allow you to identify to which database the information refers. To be able to view information in these views, you need the system privilege CATALOG READ or DATABASE ADMIN.

SAP HANA system views are separated into two categories: **metadata** views and **runtime** views. Metadata views provide metadata about objects in the database, including options or settings that were set using a DDL statement. Runtime views provide actual HANA runtime data, including statistics and status information related to the execution of DML statements. Runtime views start with M_ for monitoring.

Stores dictionary status information, remote source owner information, and the status of data collection.

Smart Data Integration: M_REMOTE_SUBSCRIPTION_COMPONENTS [page 170]
Provides the status of a remote subscription for each internal component.

Smart Data Integration: M_REMOTE_SUBSCRIPTION_STATISTICS [page 171]
Provides details of current processing details of a remote subscription (e.g. number of messages or transactions received, applied since the start of the SAP HANA database).

Smart Data Integration: M_REMOTE_SUBSCRIPTIONS [page 172]
Provides the status and run-time information of a remote subscription.

Smart Data Integration: M_SESSION_CONTEXT [page 173]
Session variables for each connection

Smart Data Integration: REMOTE_SOURCE_OBJECT_COLUMNS [page 174]
If the adapter can provide column-level information for each table in the remote system, once this dictionary is built you can search for relationships between tables. This table is useful for analyzing relationships between tables in the remote source.

Smart Data Integration: REMOTE_SOURCE_ OBJECT_DESCRIPTIONS [page 174]
Stores description of browsable node in different languages.

Smart Data Integration: REMOTE_SOURCE_OBJECTS [page 175]
Stores browsable nodes as well as importable objects (virtual tables).

Smart Data Integration: REMOTE_SOURCES [page 176]
Remote sources

Smart Data Integration: REMOTE_SUBSCRIPTION_EXCEPTIONS [page 176]
Provides details about an exception that occurred during the execution of a remote subscription. The exceptions can be processed using the PROCESS REMOTE SUBSCRIPTION EXCEPTION SQL statement.

Smart Data Integration: REMOTE_SUBSCRIPTIONS [page 177]
Lists all the remote subscriptions created for a remote source.

Smart Data Integration: TASK_CLIENT_MAPPING [page 178]
Provides the client mapping when a task is created by the ABAP API.

Smart Data Integration: TASK_COLUMN_DEFINITIONS [page 178]
Defines the columns present in a particular table.

Smart Data Integration: TASK_EXECUTIONS [page 179]
Task-level run-time statistics generated when START TASK is run.

Smart Data Integration: TASK_LOCALIZATION [page 180]
Contains localized values for the task framework tables.

Smart Data Integration: TASK_OPERATIONS [page 181]
Contains all operations that exist for a given task, as well as details about those operations.

Smart Data Integration: TASK_OPERATIONS_EXECUTIONS [page 181]
Operations-level task statistics generated when START TASK is run.

Smart Data Integration: TASK_PARAMETERS [page 182]
Details about the task parameters view

Smart Data Integration: TASK_TABLE_DEFINITIONS [page 183]
Contains all of the tables used by the various side-effect producing operation.

Smart Data Integration: TASK_TABLE_RELATIONSHIPS [page 184]

Defines the relationships, if any, between tables within an operation.

Smart Data Integration: TASKS [page 184]
Details about tasks.

Smart Data Integration: VIRTUAL_COLUMN_PROPERTIES [page 186]
Lists the properties of the columns in a virtual table sent by the adapter via CREATE VIRTUAL TABLE SQL statement.

Smart Data Integration: VIRTUAL_TABLE_PROPERTIES [page 186]
Lists the properties of a virtual table sent by the adapter via the CREATE VIRTUAL TABLE SQL statement.

Smart Data Quality: BEST_RECORD_GROUP_MASTER_STATISTICS [page 187]
Contains a summary of Best Record group master statistics.

Smart Data Quality: BEST_RECORD_RESULTS [page 188]
Contains governance information for every column in every record that is updated in the best record process.

Smart Data Quality: BEST_RECORD_STRATEGIES [page 189]
Contains information on which strategies are used in each strategy group and in which order.

Smart Data Quality: CLEANSE_ADDRESS_RECORD_INFO [page 190]
Describes how well an address was assigned as well as the type of address.

Smart Data Quality: CLEANSE_CHANGE_INFO [page 191]
Describes the changes made during the cleansing process.

Smart Data Quality: CLEANSE_COMPONENT_INFO [page 192]
Identifies the location of parsed data elements in the input and output.

Smart Data Quality: CLEANSE_INFO_CODES [page 193]
Contains one row per info code generated by the cleansing process.

Smart Data Quality: CLEANSE_STATISTICS [page 194]
Contains a summary of Cleanse statistics.

Smart Data Quality: GEOCODE_INFO_CODES [page 195]
Contains one row per info code generated by the geocode transformation process.

Smart Data Quality: GEOCODE_STATISTICS [page 196]
Contains a summary of Geocode statistics.

Smart Data Quality: MATCH_GROUP_INFO [page 196]
Contains one row for each match group.

Smart Data Quality: MATCH_RECORD_INFO [page 197]
Contains one row for each matching record per level.

Smart Data Quality: MATCH_SOURCE_STATISTICS [page 198]
Contains counts of matches within and between data sources.

Smart Data Quality: MATCH_STATISTICS [page 199]
Contains statistics regarding the run of the transformation operation.

Smart Data Quality: MATCH_TRACING [page 200]
Contains one row for each match decision made during the matching process.

6.2.1 Smart Data Integration: ADAPTER_CAPABILITIES

Specifies the SQL capabilities of the adapters stored in the system.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
ADAPTER_NAME	NVARCHAR(64)	Adapter name
SOURCE_VERSION	NVARCHAR(64)	Source versions supported by the adapter

6.2.2 Smart Data Integration: ADAPTER_LOCATIONS

Specifies the location of adapters.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
ADAPTER_NAME	NVARCHAR(64)	Adapter name
LOCATION	VARCHAR(11)	Location of the adapter: 'indexserver', 'dpserver', 'agent'
AGENT_NAME	NVARCHAR(256)	Agent name

6.2.3 Smart Data Integration: ADAPTERS

Stores adapters available in the SAP HANA system.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
ADAPTER_NAME	NVARCHAR(64)	Adapter name
PROPERTIES	NVARCHAR(1000)	Optional properties of the adapter such as display_name and description
CONFIGURATION	NCLOB	UI properties that must be displayed when configuring remote data source
IS_SYSTEM_ADAPTER	VARCHAR(5)	Specifies whether the adapter is a system adapter: 'TRUE'/'FALSE'
IS_ESS_DEFINITION_SUPPORTED	VARCHAR(5)	Specifies if the procedure GET_REMOTE_SOURCE_TABLE_ESS_DEFINITIONS is enabled for remote sources created using this adapter: 'TRUE'/'FALSE'

6.2.4 Smart Data Integration: AGENT_CONFIGURATION

Agent configuration

Structure

Column name	Data type	Description
AGENT_NAME	NVARCHAR(256)	Agent name
KEY	VARCHAR(128)	Agent property key
VALUE	NCLOB	Agent property value

6.2.5 Smart Data Integration: AGENT_GROUPS

Lists active data provisioning agent groups in the system.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
AGENT_GROUP_NAME	NVARCHAR(256)	Name of the agent group.

6.2.6 Smart Data Integration: AGENTS

Lists active data provisioning agents in the system.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
AGENT_NAME	NVARCHAR(256)	Agent name
PROTOCOL	VARCHAR(4)	Protocol for communication with SAP HANA database: 'TCP', 'HTTP'
AGENT_HOST	NVARCHAR (64)	Agent host specified when using TCP
AGENT_PORT	INTEGER	Agent port specified when using TCP
IS_SSL_ENABLED	VARCHAR(5)	Specifies whether the agent listening on TCP port uses SSL
AGENT_GROUP_NAME	NVARCHAR(256)	Agent clustering group to which the agent belongs.

6.2.7 Smart Data Integration: M_AGENTS

Provides the status of all agents registered in the SAP HANA database.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
AGENT_NAME	NVARCHAR(256)	Agent name
FREE_PHYSICAL_MEMORY	BIGINT	Free physical memory on the host
FREE_SWAP_SPACE	BIGINT	Free swap memory on the host
LAST_CONNECT_TIME	TIMESTAMP	The last time the session cookie was used for successful re-connection
SYS_TIMESTAMP	TIMESTAMP	Host timestamp in local time zone
USED_PHYSICAL_MEMORY	BIGINT	Used physical memory on the host
USED_SWAP_SPACE	BIGINT	Used swap memory on the host
UTC_TIMESTAMP	TIMESTAMP	Host timestamp in UTC
AGENT_VERSION	VARCHAR(32)	Agent version
AGENT_STATUS	VARCHAR(16)	Agent status

6.2.8 Smart Data Integration: M_REMOTE_SOURCES

Stores dictionary status information, remote source owner information, and the status of data collection.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
USER_NAME	NVARCHAR(256)	User name
REMOTE_SOURCE_NAME	NVARCHAR(256)	Remote source name
LAST_REFRESH_TIME	TIMESTAMP	The successful completion timestamp of the refresh operation
REFRESH_START_TIME	TIMESTAMP	The timestamp of when the refresh operation was executed
REFRESH_STATUS	VARCHAR(32)	Refresh operation status: • STARTED • COMPLETED • RUNNING (GET OBJECTS) • RUNNING (GET OBJECT DETAILS) • FAILED • CANCELLED • CLEARED
REFRESH_ERROR_MESSAGE	NVARCHAR(2000)	Exception message that occurred during refresh operation

6.2.9 Smart Data Integration: M_REMOTE_SUBSCRIPTION_COMPONENTS

Provides the status of a remote subscription for each internal component.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Remote subscription schema name
SUBSCRIPTION_NAME	NVARCHAR(256)	Remote subscription name
COMPONENT	VARCHAR(10)	• DPSERVER • ADAPTER • RECEIVER • APPLIER

Column	Data type	Description
STATUS	VARCHAR	Component status
MESSAGE	VARCHAR	Additional information

6.2.10 Smart Data Integration: M_REMOTE_SUBSCRIPTION_STATISTICS

Provides details of current processing details of a remote subscription (e.g. number of messages or transactions received, applied since the start of the SAP HANA database).

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Remote subscription schema name
SUBSCRIPTION_NAME	NVARCHAR(256)	Remote subscription name
RECEIVED_MESSAGE_COUNT	BIGINT	Total message/transaction count received by the current connection
RECEIVED_MESSAGE_SIZE	BIGINT	Total size of messages/transactions received by the current connection
APPLIED_MESSAGE_COUNT	BIGINT	Total number of messages/transactions applied
APPLIED_MESSAGE_SIZE	BIGINT	Total size of messages/records applied
REJECTED_MESSAGE_COUNT	BIGINT	Total number of messages/records rejected
LAST_MESSAGE_RECEIVED	TIMESTAMP	Time at which the last message/transaction is received
LAST_MESSAGE_APPLIED	TIMESTAMP	Time at which the last message/transaction is applied
RECEIVER_LATENCY	BIGINT	Receiver latency in microseconds
APPLIER_LATENCY	BIGINT	Applier latency in microseconds

6.2.11 Smart Data Integration: M_REMOTE_SUBSCRIPTIONS

Provides the status and run-time information of a remote subscription.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Remote subscription schema name
SUBSCRIPTION_NAME	NVARCHAR(256)	Remote subscription name
STATE	VARCHAR(256)	State of event
OPTIMIZED_QUERY_STRING	NCLOB	This is generated and saved so that if there are multiple subscriptions interested in same query result, and the same internal_distribution_id, both the subscriptions can use the same result.
OPTIMIZED_QUERY_HASH	VARCHAR(128)	Hash is used to query the match for optimized query string
INTERNAL_DISTRIBUTION_ID	BIGINT	Generated integer to identify if multiple target tables are interested in the changes from same source SQL or virtual table
OPTIMIZED_QUERY_RESULTSET_TYPE	TINYINT	0 - REGULAR 1 - CLUSTER 2 - POOL
REMOTE_SUBSCRIPTION	NVARCHAR(256)	An optional subscription name registered by the adapter in the remote source system
VOLUME_ID	INTEGER	Persistence Volume ID
BEGIN_MARKER	VARCHAR(64)	Generated begin marker in the format `B<remote_source_oid>_<remote_subscription_oid>_<YYYYMMDDHH24MMSSFF7>` when QUEUE command is called.
END_MARKER	VARCHAR(64)	Generated end marker in the format `E<remote_source_oid>_<remote_subscription_oid>_<YYYYMMDDHH24MMSSFF7>` when DISTRIBUTE command is called.
BEGIN_MARKER_TIME	TIMESTAMP	Timestamp when QUEUE request is received.

Column	Data type	Description
END_MARKER_TIME	TIMESTAMP	Timestamp when DISTRIBUTE command is called.
LAST_PROCESSED_TRANSACTION_ID	VARBINARY(128)	Transaction ID of the last processed transaction.
LAST_PROCESSED_TRANSACTION_TIME	TIMESTAMP	Time when the last transaction was applied.
LAST_PROCESSED_BEGIN_SEQUENCE_ID	VARBINARY(68)	Last processed transaction's begin record sequence ID
LAST_PROCESSED_COMMIT_SEQUENCE_ID	VARBINARY(68)	Last processed transaction's commit record sequence ID
LAST_RECEIVED_SEQUENCE_ID	VARBINARY(68)	Last received sequence ID
LAST_RECEIVED_CUSTOM_ID	NVARCHAR(64)	Last received custom ID. Custom IDs may be used by adapters with every changed-data row of a transaction.
LAST_PROCESSED_CUSTOM_ID	NVARCHAR(64)	Last processed custom ID. Custom IDs may be used by adapters with every changed-data row of a transaction.

6.2.12 Smart Data Integration: M_SESSION_CONTEXT

Session variables for each connection

> **Note**
>
> The M_SESSION_CONTEXT view is available for use in other areas of SAP HANA, not only SAP HANA smart data integration. Refer to the M_SESSION_CONTEXT topic for complete information. The information below is specific to smart data integration functionality.

This view shows session variables of all open connections.

Each variable is categorized in SECTION column to USER (user defined variable using SET command or client API call) or SYSTEM (predefined variable or system property).

Table 36: Predefined variables

Variable Name (M_SESSION_CONTEXT.KEY)	Value Constraint	Set by Client or Server	Shown in M_SESSION_CONTEXT	Server Usage	Description
TASK_EXECUTION_ID	bigint	server	yes	START TASK	Shows unique task execution ID

6.2.13 Smart Data Integration: REMOTE_SOURCE_OBJECT_COLUMNS

If the adapter can provide column-level information for each table in the remote system, once this dictionary is built you can search for relationships between tables. This table is useful for analyzing relationships between tables in the remote source.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
USER_NAME	NVARCHAR(256)	For secondary credentials, need to know the owner name
REMOTE_SOURCE_NAME	NVARCHAR(256)	To uniquely identify a remote source
OBJECT_NAME	NVARCHAR(5000)	Unique name to identify remote source object
COLUMN_NAME	NVARCHAR(256)	Column name
DATA_TYPE_NAME	VARCHAR(16)	SAP HANA data type
REMOTE_DATA_TYPE_NAME	VARCHAR(32)	Remote source data type
REMOTE_CONTENT_TYPE	NVARCHAR(256)	Examples include address, unit of measure, user-defined types, ZIP code, and so on
LENGTH	INTEGER	Length/precision of the column
SCALE	INTEGER	Scale of the column
IS_NULLABLE IS_AUTOINCREMENT	VARCHAR(5)	Various column properties

6.2.14 Smart Data Integration: REMOTE_SOURCE_OBJECT_DESCRIPTIONS

Stores description of browsable node in different languages.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
USER_NAME	NVARCHAR(256)	User name
REMOTE_SOURCE_NAME	NVARCHAR(256)	Remote source name
OBJECT_NAME	NVARCHAR(5000)	Unique name to identify remote source object
LANGUAGE_CODE	VARCHAR(2)	Language code
DESCRIPTION	NVARCHAR(5000)	Description of this object

6.2.15 Smart Data Integration: REMOTE_SOURCE_OBJECTS

Stores browsable nodes as well as importable objects (virtual tables).

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
USER_NAME	NVARCHAR(256)	User name
REMOTE_SOURCE_NAME	NVARCHAR(256)	Remote source name
OBJECT_NAME	NVARCHAR(5000)	Unique name to identify remote source object
DISPLAY_NAME	NVARCHAR(256)	Display name for this object
IS_IMPORTABLE	VARCHAR(5)	If the object is importable as a virtual table: 'TRUE'/'FALSE'
IS_EXPANDABLE	VARCHAR(5)	If the object can be expanded or browsed to get inner objects: 'TRUE/'FALSE'
PARENT_OBJECT_NAME	NVARCHAR(5000)	The parent object name for this object
DEFINITION_TYPE	VARCHAR(32)	Object definition type
DEFINITION	NCLOB	Object definition

6.2.16 Smart Data Integration: REMOTE_SOURCES

Remote sources

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column name	Data type	Description
REMOTE_SOURCE_NAME	NVARCHAR(256)	Remote source name
ADAPTER_NAME	NVARCHAR(256)	Adapter name
CONNECTION_INFO	NVARCHAR(256)	Connection information
AGENT_GROUP_NAME	NVARCHAR(256)	Name of the agent group name

6.2.17 Smart Data Integration: REMOTE_SUBSCRIPTION_EXCEPTIONS

Provides details about an exception that occurred during the execution of a remote subscription. The exceptions can be processed using the PROCESS REMOTE SUBSCRIPTION EXCEPTION SQL statement.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
EXCEPTION_OID	BIGINT	Exception ID
OBJECT_TYPE	VARCHAR(19)	'REMOTE SOURCE', 'REMOTE SUBSCRIPTION'
OBJECT_SCHEMA_NAME	NVARCHAR(256)	Schema name of remote source or remote subscription based on OBJECT_TYPE

Column	Data type	Description
OBJECT_NAME	NVARCHAR(256)	Object name of remote source or remote subscription based on OBJECT_TYPE
EXCEPTION_TIME	TIMESTAMP	Time at which the exception was raised
ERROR_NUMBER	INTEGER	Error number
ERROR_MESSAGE	NVARCHAR(2000)	Error message
COMPONENT	VARCHAR(8)	Component that raised the exception

6.2.18 Smart Data Integration: REMOTE_SUBSCRIPTIONS

Lists all the remote subscriptions created for a remote source.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Remote subscription schema name
SUBSCRIPTION_NAME	NVARCHAR(256)	Remote subscription name
OWNER_NAME	NVARCHAR(256)	Owner name
REMOTE_SOURCE_NAME	NVARCHAR(256)	Remote source name
IS_VALID	VARCHAR(5)	Specifies whether the remote subscription is valid or not. This becomes FALSE when its source or target objects are changed or dropped.
SUBSCRIPTION_TYPE	VARCHAR(13)	Remote subscription type
VIRTUAL_TABLE_SCHEMA_NAME	NVARCHAR(256)	Virtual table schema name
VIRTUAL_TABLE_NAME	NVARCHAR(256)	Virtual table name
SUBSCRIPTION_QUERY_STRING	NCLOB	Select statement specified in the subscription when subscription type is SQL
TARGET_OBJECT_TYPE	VARCHAR(9)	Remote subscription target object type: 'TABLE', 'PROCEDURE', 'TASK'
TARGET_OBJECT_SCHEMA_NAME	NVARCHAR(256)	Target object schema name
TARGET_OBJECT_NAME	NVARCHAR(256)	Target object name

Column	Data type	Description
TARGET_OTHER_PARAM_STRING	NVARCHAR(4000)	Constant parameter string to pass at execution when target object type is PROCEDURE or TASK
TASK_PROCEDURE_PARAMETERS	NVARCHAR(5000)	A comma-separated list of task parameters.

6.2.19 Smart Data Integration: TASK_CLIENT_MAPPING

Provides the client mapping when a task is created by the ABAP API.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
CLIENT	NVARCHAR(128)	Name of the client that created the task with the ABAP API

6.2.20 Smart Data Integration: TASK_COLUMN_DEFINITIONS

Defines the columns present in a particular table.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation
COLUMN_NAME	NVARCHAR(128)	Name of the column used in the task plan within a table
MAPPED_NAME	NVARCHAR(128)	Mapped name of the column used in a task plan within a table

6.2.21 Smart Data Integration: TASK_EXECUTIONS

Task-level run-time statistics generated when START TASK is run.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

TASK_EXECUTIONS shows one record per task plan execution.

Data in this view is updated while the task is in progress. For example, STATUS, PROCESSED_RECORDS, and TOTAL_PROGRESS_PERCENT are continuously updated until the task is complete.

Users may view information only for tasks that they ran themselves or were granted permissions to view.

Structure

Column	Data type	Description
HOST	VARCHAR(64)	Host name
PORT	INTEGER	Internal port
SCHEMA_NAME	NVARCHAR(256)	Schema name used in the task
TASK_NAME	NVARCHAR(256)	Name of the task
CONNECTION_ID	INTEGER	Connection identifier
TRANSACTION_ID	INTEGER	Transaction identifier used for the task execution
TASK_EXECUTION_ID	BIGINT	Task execution unique identifier

Column	Data type	Description
PARENT_TASK_EXECUTION_ID	BIGINT	Parent task identifier
IS_ASYNC	VARCHAR(5)	TRUE if the task is asynchronous, else FALSE
PARAMETERS	NVARCHAR(5000)	Input parameters for the task
PROCEDURE_PARAMETERS	NVARCHAR(5000)	Displays the input <param-list> values that were specified in the START TASK SQL command
START_TIME	TIMESTAMP	Start time of the task
END_TIME	TIMESTAMP	End time of the task
DURATION	BIGINT	Execution time of the task (microseconds)
STATUS	VARCHAR(16)	Status of the task: STARTING, RUNNING, FAILED, COMPLETED, CANCELLING, or CANCELLED
CURRENT_OPERATION	NVARCHAR(128)	Current operation of the task
PROCESSED_RECORDS	BIGINT	Total number of records processed
TOTAL_PROGRESS_PERCENT	BIGINT	Total task progress (percent)
USER_NAME	NVARCHAR(256)	User name
APPLICATION_USER_NAME	NVARCHAR(256)	Application user name
HAS SIDE EFFECTS	VARCHAR(5)	'TRUE' if the task produces side effect data, else 'FALSE'

6.2.22 Smart Data Integration: TASK_LOCALIZATION

Contains localized values for the task framework tables.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
LOC_TYPE_ID	INTEGER	Identifier of the type of the entity being localized
LOC_ID	NVARCHER(64)	Identifier of the entity being localized
LANGUAGE	NVARCHAR(1)	One-character code of the localized language
DESCRIPTION	NVARCHAR(1024)	Localized description

6.2.23 Smart Data Integration: TASK_OPERATIONS

Contains all operations that exist for a given task, as well as details about those operations.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
COMMENTS	NVARCHAR(512)	Comments made on the operation
HAS_SIDE_EFFECTS	TINYINT	Specifies whether the operation has side-effect data
OPERATION_TYPE	NVARCHAR(128)	Type of operation in the task plan

6.2.24 Smart Data Integration: TASK_OPERATIONS_EXECUTIONS

Operations-level task statistics generated when START TASK is run.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

TASK_OPERATIONS_EXECUTIONS shows one record per operation.

Data in this view is updated while the task is in progress. For example, STATUS, PROCESSED_RECORDS, and OPERATIONS_PROGRESS_PERCENT are continuously updated until the task is complete.

Users may view information only for tasks that they ran themselves or were granted permissions to view.

Structure

Column	Data type	Description
HOST	VARCHAR(64)	Host name
PORT	INTEGER	Internal port
TASK_EXECUTION_ID	BIGINT	Task identifier
CONNECTION_ID	INTEGER	Connection identifier
TRANSACTION_ID	INTEGER	Transaction identifier used for the task execution
CURRENT_OPERATION	NVARCHAR	Name of operation
OPERATION_TYPE	NVARCHAR(128)	Type of operation
OPERATION_NAME	NVARCHAR(128)	Internal name of operation
START_TIME	TIMESTAMP	Start time of the task
END_TIME	TIMESTAMP	End time of the task
DURATION	BIGINT	Execution time of the task (microseconds)
STATUS	VARCHAR(16)	Status of the task: • STARTING • RUNNING • FAILED • COMPLETED • CANCELLING • CANCELLED
PROCESSED_RECORDS	BIGINT	Total number of records processed
OPERATION_PROGRESS_PERCENT	DOUBLE	Operation progress (percent)
HAS_SIDE_EFFECTS	VARCHAR(5)	'TRUE' if the task produces side effect data, else 'FALSE'

6.2.25 Smart Data Integration: TASK_PARAMETERS

Details about the task parameters view

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Schema in which the task was created
TASK_NAME	NVARCHAR(256)	Name of task
PARAMETER_NAME	NVARCHAR(256)	Name of parameter
POSITION	INTEGER	Position of parameter
TABLE_TYPE_SCHEMA	NVARCHAR(256)	Schema in which the TableType was created
TABLE_TYPE_NAME	NVARCHAR(256)	Name of TableType
PARAMETER_TYPE	VARCHAR(7)	Parameter type: IN or OUT

6.2.26 Smart Data Integration: TASK_TABLE_DEFINITIONS

Contains all of the tables used by the various side-effect producing operation.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_ID	INTEGER	Unique identifier for the table
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for an operation
SIDE_EFFECT_SCHEMA	NVARCHAR(128)	Schema where the generated side-effect table is located
SIDE_EFFECT_NAME	NVARCHAR(128)	Name of the generated side-effect table
IS_PRIMARY_TABLE	TINYINT	Specifies whether this table is the primary table in a relationship

Column	Data type	Description
OPERATION_TABLE_TYPE	NVARCHAR(20)	Type of operation that the table is used within

6.2.27 Smart Data Integration: TASK_TABLE_RELATIONSHIPS

Defines the relationships, if any, between tables within an operation.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for an operation
RELATED_TABLE_NAME	NVARCHAR(128)	Name of the table to which the table specified in TABLE_NAME is related
FROM_ATTRIBUTE	NVARCHAR(128)	Name of the column in the TABLE_NAME table that relates to the TO_ATTRIBUTE
TO_ATTRIBUTE	NVARCHAR(128)	Name of the column in the RELATED_TABLE_NAME table that relates to the FROM_ATTRIBUTE

6.2.28 Smart Data Integration: TASKS

Details about tasks.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
TASK_OID	BIGINT	Unique identifier for a task
TASK_NAME	NVARCHAR(256)	Name of task
SCHEMA_NAME	NVARCHAR(256)	Schema the task was created in
OWNER_NAME	NVARCHAR(256)	Owner of the task
CREATE_TIME	TIMESTAMP	Creation time
MEMORY_SIZE	BIGINT	Memory size of loaded task
TASK_TYPE	NVARCHAR(64)	Type of task ('PLAN' or 'PROCEDURE'), based on how the task was created
PLAN_VERSION	NVARCHAR(32)	Version of the task plan
PLAN	NCLOB	Task plan used to define the task, or task plan generated to call the procedure
COMMENTS	NVARCHAR(256)	Description of the task, from the task plan
HAS_TABLE_TYPE_INPUT	VARCHAR(5)	'TRUE' if the task is modeled with a table type as input, meaning data would need to be passed at execution time
HAS SDQ	VARCHAR(5)	'TRUE' if the task contains SDQ (smart data quality) functionality
IS_REALTIME_TASK	VARCHAR(5)	'TRUE' if the task is a realtime task, else 'FALSE'
IS_VALID	VARCHAR(5)	'TRUE' if the task is in a valid state; 'FALSE if it has been invalidated by a dependency
IS_READ_ONLY	VARCHAR(5)	'TRUE' if the task is read only (has only table type outputs), 'FALSE' if it writes to non-table-type outputs
PROCEDURE_SCHEMA	NVARCHAR(256)	If the task was created with a procedure instead of a plan, this attribute will contain the schema name of the stored procedure
PROCEDURE_NAME	NVARCHAR(256)	If the task was created with a procedure instead of a plan, this attribute will contain the name of the name of the stored procedure
INPUT_PARAMETER_COUNT	SMALLINT	Number of input (tableType) parameters
OUTPUT_PARAMETER	SMALLINT	Number of output (tableType) parameters
SQL_SECURITY	VARCHAR(7)	Security model for the task, either 'DEFINER' or 'INVOKER'

6.2.29 Smart Data Integration: VIRTUAL_COLUMN_PROPERTIES

Lists the properties of the columns in a virtual table sent by the adapter via CREATE VIRTUAL TABLE SQL statement.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Schema name of virtual table
TABLE_NAME	NVARCHAR(256)	Virtual table name
COLUMN_NAME	NVARCHAR(256)	Virtual table column name
PROPERTY	NVARCHAR(256)	Property name
VALUE	NVARCHAR(512)	Property value

6.2.30 Smart Data Integration: VIRTUAL_TABLE_PROPERTIES

Lists the properties of a virtual table sent by the adapter via the CREATE VIRTUAL TABLE SQL statement.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Schema name of virtual table
TABLE_NAME	NVARCHAR(256)	Virtual table name
PROPERTY	NVARCHAR(256)	Property name

Column	Data type	Description
VALUE	NCLOB	Property value. For example: • Large XSD of size 1M

6.2.31 Smart Data Quality: BEST_RECORD_GROUP_MASTER_STATISTICS

Contains a summary of Best Record group master statistics.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
NUM_RECORDS	BIGINT	Total number of records processed
NUM_GROUP_MASTERS	BIGINT	Number of group master records processed
NUM_DUPLICATES	BIGINT	Number of duplicate records processed
NUM_SURVIVORS	BIGINT	Number of surviving records processed
NUM_NON_MATCH_RECORDS	BIGINT	Number of non-matching records processed

6.2.32 Smart Data Quality: BEST_RECORD_RESULTS

Contains governance information for every column in every record that is updated in the best record process.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
DST_TABLE_NAME	NVARCHAR(128)	Name of the destination table for the operation
DST_ROW_ID	BIGINT	Unique identifier for the destination row
DST_COLUMN_NAME	NVARCHAR(128)	Name of the destination column in the destination table
DST_ROW_TYPE	NVARCHAR(1)	Identifies how the record was updated or if it was newly created
SRC_TABLE_NAME	NVARCHAR(128)	Name of the source table for the operation
SRC_ROW_ID	BIGINT	Unique identifier for the source row
SRC_COLUMN_NAME	NVARCHAR(128)	Name of the source column in the source table
STRATEGY_GROUP_ID	INTEGER	Identification number that identifies the best record strategy group
STRATEGY_ID	INTEGER	Identification number that identifies each strategy listed in the strategy group
BEST_RECORD_RULE	NVARCHAR(256)	Name of the rule that updates one or more columns as it is defined in the best record configuration
ACTION_NAME	NVARCHAR(256)	Name of the action that updates a column as it is defined in the best record configuration

Column	Data type	Description
UPDATE_NUM	INTEGER	Number of times the column was updated in the best record process
OPERATION_TYPE	NVARCHAR(1)	Identifies how the record was updated in the best record process

6.2.33 Smart Data Quality: BEST_RECORD_STRATEGIES

Contains information on which strategies are used in each strategy group and in which order.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
STRATEGY_GROUP_NAME	NVARCHAR(256)	Name of the strategy group as defined in the best record configuration
STRATEGY_ID	INTEGER	Identification number that identifies each strategy listed in the strategy group
STRATEGY_ORDER	INTEGER)	Order of the strategy as defined in the list of strategies
STRATEGY_NAME	NVARCHAR(256)	Name of the strategy as defined in the best record configuration

6.2.34 Smart Data Quality: CLEANSE_ADDRESS_RECORD_INFO

Describes how well an address was assigned as well as the type of address.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
ENTITY_INSTANCE	INTEGER	Identifier to differentiate between multiple entities processed in a row
ENTITY_INSTANCE_OCCURRENCE	INTEGER	Unique identifier to identify the occurrence of an entity
DATA_SOURCE	NVARCHAR(256)	Source where the data was produced
ISO_COUNTRY_2CHAR	NVARCHAR(4)	Two-character country code
ASSIGNMENT_TYPE	NVARCHAR(4)	Code that represents the type of an address
ASSIGNMENT_INFORMATION	NVARCHAR(4)	Code that specifsies the validity of an address
ASSIGNMENT_LEVEL	NVARCHAR(4)	Code that represents the level to which the address matched data in the address reference data

6.2.35 Smart Data Quality: CLEANSE_CHANGE_INFO

Describes the changes made during the cleansing process.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
ENTITY_ID	NVARCHAR(12)	Identifier describing the type of record that was processed
ENTITY_INSTANCE	INTEGER	Identifier to differentiate between multiple entities processed in a row
ENTITY_INSTANCE_OCCURRENCE	INTEGER	Unique identifier to identify the occurrence of an entity
COMPONENT_ID	NVARCHAR(12)	Identification number that refers to data components
COMPONENT_ELEMENT_ID	NVARCHAR(12)	Identification number that refers to more granular elements within a component
DATA_SOURCE	NVARCHAR(256)	Source where the data was produced
CHANGE_SIGNIFICANCE_ID	NVARCHAR(12)	Identification number that refers to the significance of the change

6.2.36 Smart Data Quality: CLEANSE_COMPONENT_INFO

Identifies the location of parsed data elements in the input and output.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
ENTITY_ID	NVARCHAR(12)	Identifier describing a data attribute such as a person name, organization name, address and so on.
ENTITY_INSTANCE	INTEGER	Identifier to differentiate between multiple entities processed in a row
ENTITY_INSTANCE_OCCURRENCE	INTEGER	Unique identifier to identify the occurrence of an entity
DATA_SOURCE	NVARCHAR(256)	Source where the data originated
COMPONENT_ID	NVARCHAR(12)	Identification number that refers to data components
COMPONENT_ELEMENT_ID	NVARCHAR(12)	Identification number that refers to more granular elements within a component
TABLE_NAME	NVARCHAR(128)	Name of the input table where the component element was found
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
COLUMN_NAME	NVARCHAR(128)	Name of the column in the input table where the component element was found

Column	Data type	Description
COLUMN_START_POSITION	INTEGER	Starting character of the component element in the input column
COLUMN_DATA_LENGTH	INTEGER	Number of characters of the component element in the input column
OUTPUT_TABLE_NAME	NVARCHAR(128)	Name of the output table where the component element was written
OUTPUT_COLUMN_NAME	NVARCHAR(128)	Name of the column in the output table where the component element was written
OUTPUT_COLUMN_START_POSITION	INTEGER	Starting character of the component element in the output column
OUTPUT_COLUMN_DATA_LENGTH	INTEGER	Number of characters of the component element in the output column

6.2.37 Smart Data Quality: CLEANSE_INFO_CODES

Contains one row per info code generated by the cleansing process.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation

Column	Data type	Description
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
ENTITY_ID	NVARCHAR(12)	Identifier describing the type of record that was processed
ENTITY_INSTANCE	INTEGER	Identifier to differentiate between multiple entities processed in a row
ENTITY_INSTANCE_OCCURRENCE	INTEGER	Unique identifier to identify the occurrence of an entity
DATA_SOURCE	NVARCHAR(256)	Source where the data was produced
INFO_CODE	NVARCHAR(10)	Information code that gives information about the processing of the record

6.2.38 Smart Data Quality: CLEANSE_STATISTICS

Contains a summary of Cleanse statistics.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
ENTITY_ID	NVARCHAR(12)	Identifier describing the type of record that was processed
ENTITY_INSTANCE	INTEGER	Identifier to differentiate between multiple entities processed in a row

Column	Data type	Description
NUM_RECORDS	BIGINT	Total number of records processed for the entity instance
NUM_VALIDS	BIGINT	Number of valid records processed for the entity instance
NUM_SUSPECTS	BIGINT	Number of suspect records processed for the entity instance
NUM_BLANKS	BIGINT	Number of blank records processed for the entity instance
NUM_HIGH_SIGNIFICANT_CHANGES	BIGINT	Number of records with high significance changes for the entity instance

6.2.39 Smart Data Quality: GEOCODE_INFO_CODES

Contains one row per info code generated by the geocode transformation process.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
DATA_SOURCE	NVARCHAR(256)	Source where the data was produced
INFO_CODE	NVARCHAR(10)	Information code generated by the geocode transformation operation

6.2.40 Smart Data Quality: GEOCODE_STATISTICS

Contains a summary of Geocode statistics.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
NUM_RECORDS	BIGINT	Total number of records processed
NUM_ASSIGNED	BIGINT	Number of assigned records processed
NUM_UNASSIGNED	BIGINT	Number of unassigned records processed

6.2.41 Smart Data Quality: MATCH_GROUP_INFO

Contains one row for each match group.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
GROUP_ID	INTEGER	Group identification number
GROUP_COUNT	INTEGER	Number of records in the match group
SOURCE_COUNT	INTEGER	Number of sources represented in the match group
REVIEW_GROUP	NVARCHAR(1)	Indicates whether the group is flagged for review
CONFLICT_GROUP	NVARCHAR(1)	Indicates whether the group is flagged for conflict

6.2.42 Smart Data Quality: MATCH_RECORD_INFO

Contains one row for each matching record per level.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan

Column	Data type	Description
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
GROUP_ID	INTEGER	Group identification number

6.2.43 Smart Data Quality: MATCH_SOURCE_STATISTICS

Contains counts of matches within and between data sources.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
SOURCE_NAME	NVARCHAR(256)	Data source name
RELATED_SOURCE_NAME	NVARCHAR(256)	Related data source name
NUM_MATCH_DECISIONS	INTEGER	Number of comparisons resulting in a match decision between records in each SOURCE_ID/RELATED_SOURCE_ID pair

6.2.44 Smart Data Quality: MATCH_STATISTICS

Contains statistics regarding the run of the transformation operation.

> **Caution**
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
NUM_RECORDS	BIGINT	Total number of records processed by the transformation operation
NUM_MATCH_RECORDS	BIGINT	Number of records that reside in match groups
NUM_NON_MATCH_RECORDS	BIGINT	Number of non-matching records that do not reside in match groups
NUM_MATCH_GROUPS	BIGINT	Number of match groups identified
NUM_REVIEW_GROUPS	BIGINT	Number of match groups flagged for review
NUM_NON_REVIEW_GROUPS	BIGINT	Number of match groups not flagged for review
NUM_CONFLICT_GROUPS	BIGINT	Number of match groups flaggged with conflicts
NUM_COMPARISONS_PERFORMED	BIGINT	Number of comparisons performed by the transformation operation
NUM_MATCH_DECISIONS	BIGINT	Number of comparisons resulting in a match decision

6.2.45 Smart Data Quality: MATCH_TRACING

Contains one row for each match decision made during the matching process.

> ⚠ Caution
>
> This information applies only for the SAP HANA option cited in the title of this topic. You must have purchased the license for it in order to use the information contained in this topic.

Structure

Column	Data type	Description
SCHEMA_NAME	NVARCHAR(256)	Name of the schema where the task is located
TASK_NAME	NVARCHAR(256)	Name of the task
TASK_EXECUTION_ID	BIGINT	Unique identifier for a particular run of a task plan created when START TASK is called
OPERATION_NAME	NVARCHAR(128)	Name of the operation in the task plan
TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for the operation
ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
RELATED_TABLE_NAME	NVARCHAR(128)	Name of the table defined in the task plan for an operation
RELATED_ROW_ID	BIGINT	Unique identifier of the row processed for this execution of the task plan
POLICY_NAME	NVARCHAR(256)	Name of the match policy that processed the related rows
RULE_NAME	NVARCHAR(256)	Name of the match rule that processed the related rows
SCORE	INTEGER	Similarity score of the related rows

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 a 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.

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.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

go.sap.com/registration/contact.html