SAP®

Configuration and Users Guide

Replication Agent™ for SAP HANA® Database 15.7.1 SP200

Microsoft Windows, UNIX, and Linux

Contents

Contents

Conventions

These style and syntax conventions are used in SAP® documentation.

Style conventions

Key	Definition
`monospaced (fixed-width)`	• SQL and program code • Commands to be entered exactly as shown • File names • Directory names
`italic monospaced`	In SQL or program code snippets, placeholders for user-specified values (see example below).
italic	• File and variable names • Cross-references to other topics or documents • In text, placeholders for user-specified values (see example below) • Glossary terms in text
bold sans serif	• Command, function, stored procedure, utility, class, and method names • Glossary entries (in the Glossary) • Menu option paths • In numbered task or procedure steps, user-interface (UI) elements that you click, such as buttons, check boxes, icons, and so on

If necessary, an explanation for a placeholder (system- or setup-specific values) follows in text. For example:

Run:

```
installation directory\start.bat
```

where *installation directory* is where the application is installed.

Syntax conventions

Key	Definition
{ }	Curly braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you enter the command.
[ ]	Brackets mean that choosing one or more of the enclosed options is optional. Do not type the brackets when you enter the command.
( )	Parentheses are to be typed as part of the command.
|	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas that you type as part of the command.
...	An ellipsis (three dots) means you may repeat the last unit as many times as you need. Do not include ellipses in the command.

Case-sensitivity

- All command syntax and command examples are shown in lowercase. However, replication command names are not case-sensitive. For example, **RA_CONFIG**, **Ra_Config**, and **ra_config** are equivalent.
- Names of configuration parameters are case-sensitive. For example, **Scan_Sleep_Max** is not the same as **scan_sleep_max**, and the former would be interpreted as an invalid parameter name.
- Database object names are not case-sensitive in replication commands. However, to use a mixed-case object name in a replication command (to match a mixed-case object name in the primary database), delimit the object name with double quote characters. For example: **pdb_get_tables** "*TableName*"

About Replication Agent for SAP HANA

Replication Agent™ for SAP HANA® is a lightweight server that replicates data from a primary to a replicate SAP HANA database.

Replication Agent for SAP HANA performs initial data extraction and load, and also replicates data changes on the pimary SAP HANA server to the replicate SAP HANA server.

System Architecture

Replication Agent for SAP HANA is an application that runs inside a Replication Management Agent (RMA) container.

The RMA container is a server that provides many services, such as a management application programming interface (API), a logging service, and a repository for metadata. The container may run applications, such as Replication Agent for SAP HANA, that consume these services, or may simply be used as an agent for communication with other containers. You can connect multiple containers to form a peer-to-peer grid. When you begin to replicate data from SAP HANA database, you must first start a container that contains the Replication Agent for SAP HANA components.
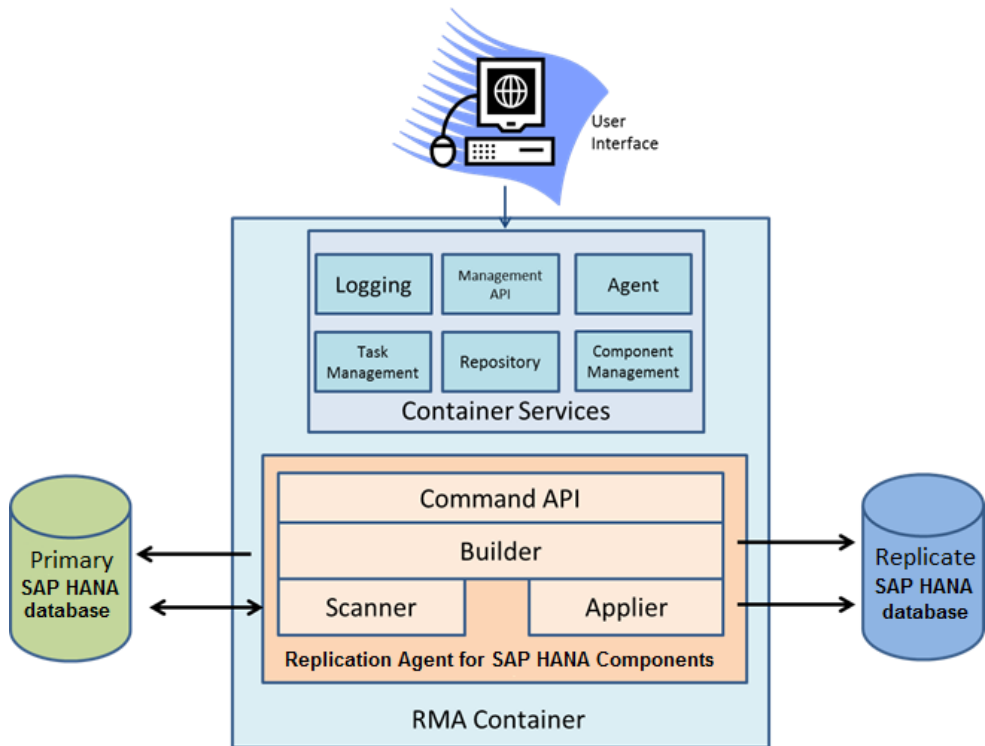
These RMA components are used by Replication Agent for SAP HANA:

- Command API – is an API that executes commands for the Replication Agent for SAP HANA. For example, **Start Replication** returns results that depend on the command executed.
- Builder – builds shadow tables and other control tables. For example, the trigger queue replicates data at the primary database, builds user tables, and sends change data capture (CDC) tables to the replicate database.
- Scanner – reads data from the shadow tables on the source database.
- Applier – writes data to the CDC tables using one of the CDC modes and the replicate database user-defined tables when a non-CDC mode is enabled.

You can use the Replication Agent in the SAP HANA Control Center graphical user interface (GUI) or from the command line interface (CLI) to issue replication commands. The CLI commands are processed by the Tabular Data Stream™ (TDS) communication protocol in the container.

Support for Multitenancy
Multitenancy allows system administrators to deploy a single instance of the software that runs on a server, serving multiple client organizations known as tenants.

Replication Agent for SAP HANA allows you to deploy mutiple replication models to a single container as long as each model has a distinct schema. You can configure, manage, deploy, and undeploy each replication model independently.

Replication Agent for SAP HANA does not support deployment of mutliple replication models that have the same schema. For example, you cannot deploy replication models where both read from or write to have the same database schema.

Replication Strategies

Replication Agent for SAP HANA creates shadow tables and triggers at the primary database; the triggers move changes from user table to shadow tables, then a replication scanning process moves the changed data from shadow tables to the replicate database.

Replication Agent for SAP HANA offers several replication strategies.

Eventual Consistency

When you use this strategy, tables are scanned separately without any commit order. Only primary key information and operation type is stored in the shadow tables; the data in the other database is queried from the primary table when it is applied to the replicate database.

Use this strategy when performance and load at the database server is more important than immediate data consistency.

Change Data Capture

This strategy is a variant of the eventual consistency strategy. The only difference is that staging tables, rather than user tables, are created and populated in the replicate database. Replication Agent for SAP HANA also manages the removal of data from the CDC tables. This strategy shares eventual consistency's performance and load characteristics.

Use this strategy when you want to stage the data before writing it to the final destination, and when replication performance and primary database server load considerations are more important than immediate data consistency.

Transactional Consistency

When you use this strategy, updates are applied at the replicate database in the same order as they were applied at the primary database. To achieve consistency, transaction identifiers are recorded with each database operation, and all columns are captured with every change.

Use this strategy when data consistency is more important than performance and database server load. Data consistency between the primary and replicate database server varies according to the amount of latency in the replication path.

This strategy places the heaviest load on the primary database and does not perform as well as the eventual consistency strategy.

Transactional Change Data Capture

This strategy is a variant of the transactional consistency strategy. The only difference is that staging tables, rather than user tables, are created and populated in the replicate database. Replication Agent for SAP HANA also manages the data removal from the CDC tables. This strategy shares transactional consistency's performance and load characteristics.

Use this strategy when you want to stage the data before writing it to the final destination. You can also use it when immediate data consistency is more important than performance and primary database server load.

Planning Your Installation

Before installing the Replication Agent for SAP HANA, verify the system requirements, download the installation software from the SAP Service Marketplace (SMP), and identify the installation directories.

Before You Begin

Before creating a replication model, identify the schemas on the primary and replicate SAP HANA databases as well as the users for those schemas on the primary and replicate databases.

To learn more about creating SAP HANA databases users, and schemas see the *SAP HANA SQL and System Views Reference*.

Installation Checklist

Complete the installation checklist before you set up the Replication Agent for SAP HANA.

Table 1. Installation Checklist

	Action	Topic
1	Meet the minimum operating system requirements.	*System Requirements* on page 8
2	Download the Replication Agent for SAP HANA software from the SMP Web site.	*Downloading Products from SMP* on page 8
3	Before installing Replication Agent for SAP HANA, choose a SySAM license model, determine license server information, and generate a license.	*Generating a License Key from SMP* on page 9
4	Determine the directory in which to install the Replication Agent for SAP HANA.	*Identify the Replication Agent for SAP HANA Installation Directory* on page 9

System Requirements

The minimum platform and operating system requirements to install the Replication Agent for SAP HANA.

Platform	Operating System Version
Linux	Supported on x64-bit server platforms: SuSE Linux Enterprise Server 11.1 (SP1), 11.2 (SP2)
Windows	GUI platforms supported: • Windows Server 2008 R2 • Windows Server 2012 • Windows XP Professional • Windows 7 (64-bit) • Windows Vista

Downloading Products from SMP

Download Replication Agent for SAP HANA software from the SMP Web site.

Prerequisites
An SMP login ID and password.

Task

1. Go to *https://websmp204.sap-ag.de/patches*, and enter your login ID and password.
2. Select **SAP Software Download Center > Installations and Upgrades > A-Z Index** and click **R** for SAP® Replication Server® products.
3. From the product listing, select **Sybase Replication Server, HANA Edition 15.7**.
4. Select the software you need by version and platform.
5. Generate a license RAH_Server license key based on the served or unserved SySAM license model.

   See *Generating a License Key from SMP* on page 8.
6. Copy the license key file in this directory after installation:
   • `$SYBASE/SYSAM-2_0/licenses` (UNIX or Linux)
   • `%SYBASE%\SYSAM-2_0\licenses` (Windows)

where *$SYBASE* or *%SYBASE%* is the installation directory of your product.

7. Download your installation files, and uncompress and extract all the installation images to your local drive.

Generating a License Key from SMP

Before installing Replication Agent for SAP HANA, choose a SySAM license model, determine license server information, and generate a license.

When you purchase SySAM 2–enabled SAP® products, you must generate, download, and deploy SySAM product licenses.

These steps summarize the actions you must take to install a SySAM license. See the *SySAM Users Guide* for details.

1. Choose a SySAM license model.

License Model	Description
Unserved license model	Licenses are obtained directly from a license file. If you are using an unserved license, save the license file to the machine on which you install the product.
Served license model	A license server manages the allocation of licenses among multiple machines.

2. For the served license model, decide whether to use an existing or new license server.

   The license server and your product installation do not need to share the same machine, operating system, or architecture.

3. If you chose the served license model, either:
   - Install the new license server on a machine that does not have an existing license server, or
   - To install on a machine that is running a SySAM 2.3 license server, follow migration instructions in the *SySAM Users Guide* to migrate to the newer SySAM version.

4. Generate a RAH_SERVER license key from the SAP SMP Web site at *http:// service.sap.com/licensekeys*.

Identifying the Replication Agent for SAP HANA Installation Directory

Determine the directory in which to install the Replication Agent for SAP HANA.

The default installation directory is $SYBASE/HANARAX.

Installing Replication Agent for SAP HANA in GUI Mode

Use the SAP Replication Server media to install Replication Agent for SAP HANA.

Prerequisites

- Review the system requirements and complete the preinstallation tasks.
- Close all nonessential applications, and minimize any open windows.

Task

1. Log in to the Replication Agent host machine using an operating system user account with authority to start, stop, and administer the Replication Agent instance.

   For example, log in using the "sybase" user account.

2. Insert the SAP Replication Server 15.7.1 SP200 distribution media (CD or DVD) in the appropriate drive.

3. Mount the CD or DVD.

   On Windows:

   The operating system automatically mounts the CD or DVD. Go to step 4.

   On UNIX:

   Set the DISPLAY environment variable to point to a host with a graphical user interface such as X-Windows, otherwise, the installation starts in console mode.

   See the *SAP Replication Server Installation Guide* for setting the environment variable to point to a GUI mode on other platforms.

4. Start the installation program.

   The Replication Agent for SAP HANA installation program is in the `hrax_suite` subdirectory.

   On Windows, either:

   a) Select **Start > Run**, and enter `x:\hrax_suite\setup.exe`, where `x:` is your CD drive, or,

   b) In the Windows Explorer, double-click **setup.exe**.

   On UNIX, enter `cd /cdrom./hrax_suite/setup.bin`.

5. Click **Next**.

   You can click **Cancel** to stop the installation at any point.

6. On the End-User License Agreement window, select the geographic location.

7. Select, read, and agree to the appropriate license agreement.

   If you agree to the terms of the Replication Agent for SAP HANA license, select **I agree to the terms of the license for the install location specified**.

   You must agree to the license agreement to proceed with the installation.

   Click **Next**.

8. On the Choose Install Folder window, select the installation directory and click **Next**.

   You can use the default directory `Sybase` or choose a different directory.

9. On the Choose Install Set window, select the default choice **Full**, and click **Next**.

10. If the installer does not detect an appropriate license, you see the SySAM window. Select one of:

    • **Specify License Keys** – click **Browse** to select the license file, or enter the license key.

    • **Use Previously Deployed License Server** – if you have a SySAM license server already installed, enter the host name and the port number of the license server.

      > **Note:** Enter the host name of the machine where the license server is running and the port number, if you are not using the default port.

    • **Continue Installation Without a License Key** – if you do not have licenses for the Replication Agent for SAP HANA.

      You can install and use Replication Agent for SAP HANA without a license for a grace period of 30 days.

      To continue using Replication Agent for SAP HANA after the end of the grace period, obtain a valid license from the SAP Service Marketplace at *https://websmp204.sap-ag.de/*, and install it.

      Click **Next**.

11. On the SySAM Notification window, enter the information for e-mail notifications about license management events that require your attention.

    a) Select **Yes** and either accept the default values, or enter:

       • SMTP server host name
       • SMTP server port number
       • E-mail return address
       • Recipients of the notification
       • Message severity level of an event that triggers e-mail notification:
          • Informational
          • Warning
          • Error

    b) Select **No** if you choose not to log e-mail alerts or severity messages.

See the *SySAM Users Guide* for more information about e-mail notifications that concern licensing issues.

Click **Next**.

12. On the Pre-Installation Summary window, review the product features or components that you have selected, and click **Install**.

   • The installation program installs the components in the installation directory you specified, and shows an installation progress indicator.
   • If you encounter errors messages during the installation, exit the installation program wizard to correct the cause of the error, then restart the installation program.
   • If the software is installed successfully, you see a window confirming the successful installation.

13. Once the Replication Agent for SAP HANA product components are installed, click **Done** to close the installation program.

Installing Replication Agent for SAP HANA in Console Mode

Install the Replication Agent for SAP HANA in an interactive console mode.

Prerequisites

- Review the system requirements and complete the preinstallation tasks.
- Close all nonessential applications, and minimize any open windows.

Task

Installing Replication Agent for SAP HANA in console mode uses the same steps as the GUI wizard, except you:

- Start the installation program wizard at the command line prompt, and
- Use only the keyboard to select all of the installation options.

Note: If there is not enough disk space in your default temporary directory, set the IATEMPDIR (Linux or UNIX platform) or TMP (Windows) environment variable to a directory that has enough space. Include the full path to this directory.

1. Log in to the Replication Agent host machine using an operating system user account with authority to start, stop, and administer the Replication Agent instance.

   For example, log in using the "sybase" user account.

2. Insert the SAP Replication Server 15.7.1 SP200 distribution media (CD or DVD) in the appropriate drive.

   Note: On a Microsoft Windows platform, when the installation program starts automatically, click **Cancel**.

3. Open an operating system command window, and set the media drive as the current drive.

   Note: The Replication Agent for SAP HANA installation program is in the hrax_suite subdirectory.

   On Windows:

   ```
   setupConsole.exe -i console
   ```

   On UNIX:

   ```
   setup.bin -i console
   ```

   You see the Welcome message.

4. Follow the remaining prompts to install the Replication Agent for SAP HANA software.

The installation program installs the components in the installation directory you specified, and displays an installation progress indicator.

If you encounter errors messages during the installation, exit the installation program console to correct the cause of the error, then restart the installation program.

If the software is installed successfully, you see a window confirming the successful installation.

Starting the SAP HANA Agent Control Center GUI

Start the Replication Agent for SAP HANA Control Center GUI instance on your platform.

Prerequisites
Make sure the JAVA_HOME environment variable is set to the `SAP_JVM` directory.

Task

1. To start the Replication Agent for SAP HANA Control Center GUI, execute the start-up script for your platform.

Linux	• Navigate to the `$Replication Agent for SAP HANA/bin` directory.
	• Run `HANARAXCC.sh`.
Windows	• Navigate to the `%Replication Agent for SAP HANA%\bin` directory.
	• Run `HANARAXCC.cmd`.

   The Start Agent? dialog is displayed.

2. If a local container has not yet started communicating with the Replication Agent for SAP HANA, click **Yes** in the Start Agent? dialog to start one.

   The SAP HANA Agent Control Center GUI opens the Agent Servers window.

   The Agent Servers window shows all the registered containers running. The Known Agents node contains a list of all registered hosts on the subnet. For example, the Known Agents list contains a host named OAKN00531400A, which has these instances:
   - AgentContainer – is the management agent instance of the container.
   - RAXContainer – is the runtime agent container instance.
     The RAXContainer is used to execute a replication model.

3. To start the RAXContainer, right-click it and select the **Start Server RAXContainer** item from the drop-down menu.

   You see a message indicating that the RAXContainer server has started successfully.

4. Select the **RAXContainer** instance and click the **Properties** tab to inspect the server properties.

   If the server properties list does not appear, select the host and click the **RAXContainer** instance again.

You see a list of server instance properties. Accept the default values for these properties:

Table 2. Server Properties

Property Name	Description
Server Name	The name of the Replication Agent Container instance.
Host Name	The name of the host on which the instance is running.
Service URL	The URL that accesses the instance.
Distinguished Name	The fully qualified name of an instance.
RMI Port	The remote method invocation (RMI) port on which the instance is listening. The Replication Agent for SAP HANA uses the RMI port to communicate with the instance.
TDS Port	The TDS port on which the instance is listening. The CLI uses the TDS port to communicate with the instance.
Process ID	The operating system process ID for the instance.
Status	The status of the instance. Options include *RUNNING* and *STOPPED*.
Server Session Credentials	The default credentials are: • Login – sa • Password – Sybase123
Server Properties Overrides	These properties are in the `boot-strap.prop` file located in the `\%IN-STANCE_NAME%\config` directory. where *Instance_Name* is the name of the Replication Agent for SAP HANA.

The RAXContainer Server window has these tabs:

• Server Log tab – provides a view of the instance log. The file log is in the `%INSTANCE_NAME%\logs` directory.
• Performance tab – shows statistics on a Java Virtual Machine (JVM) for the instance, including memory usage and garbage collection.

Deploying a Replication Model

Once you have a Replication Agent for SAP HANA instance running, you can deploy a replication model.

A replication model specifies all the configuration information that a Replication Agent for SAP HANA instance requires to execute a model, and the replication strategy. For example, you can define the HANA2HANA replication model for replicating data from a primary to a replicate SAP HANA database using the transactional CDC strategy.

1. Right-click **RAXContainer** and select **Deploy A Model** menu option.
2. In the Deployment Wizard window, select the **Primary** tab and enter the configuration information for the primary SAP HANA database:
    - **Primary Database User Name** – is the user name for accessing the primary database schema.
    - **Primary Database Password** – is the password for authenticating the primary user.
    - **Primary HANA Host Name** – is the host name for the primary database.
    - **Primary HANA Port** – is the port number for the primary database.
    - **Primary HANA Schema Name** – the schema name from which you will be replicating data.
    - **Concurrent Materialization Tasks** – the number of tasks the Replication Agent for SAP HANA reads from the primary schema in parallel. Begin with the default value of 10, then adjust the value based on initial results.
3. Select the **Replicate** tab and enter the configuration information for the replicate SAP HANA database:
    - **Replicate Database User Name** – is the user name for accessing the primary schema.
    - **Replicate Database Password** – is the password for authenticating the replicate user.
    - **Replicate HANA Host Name** – is the host name for the replicate database.
    - **Replicate HANA Port** – is the port number for the replicate HANA database.
    - **Replicate HANA Schema Name** – the schema name to which you will be replicating data.
    - **Commit Batch Size** – the number of rows of data to commit per batch.

    Begin with the default value of 100, then adjust based on initial results. For testing purposes, you can use two different schemas on the same SAP HANA database.
4. Click the **Replication Strategy** field and select **TRANS_CDC** to use the transactional CDC replication strategy.
5. (Optional) If you want Replication Agent for SAP HANA to execute data definition language (DDL) statements at the replicate database, select the **Capture DDL** check box.
6. Set the retention value units based on your requirement for managing the CDC tables.

The **Retention** field is applicable only when you are using a CDC or TRANS_CDC strategy. The retention value is measured in minutes, hours, or days, specifying the length of time that long data is retained in the CDC tables before Replication Agent for SAP HANA performs a cleanup task.

7. (Optional) Click **Load** to overwrite the configuration information for the primary and replicate SAP HANA database using a `properties` file.

8. (Optional) Click **Save** to save the current primary and replicate SAP HANA database configuration information to a `properties` file.

9. Enter the model name. The default value is HANA2HANA

   Note: If you are running in a multitenancy domain, you must enter the replication model names as the source and target database schemas. For example, **SourceDBSchemaA2** and **TargetDBSchemaA**.

10. Right-click **RAXContainer** and select **Deploy A Model** menu option to deploy the replication model.

    Once the deployment is successful, the replication model name appears under the **RAXContainer** instance and a confirmation dialog is displayed.

11. Click **OK**.

Updating a Deployed Replication Model

Use the Model Properties window to update a deployed replication model.

Some properties of a deployed replication model can be updated without undeploying and redeploying the model. The properties that you can update appear in editable white background fields on the Model Properties window.

1. To edit a deployed model, select the model under the **RAXContainer**.

   For example, select the **HANA2HANA** model.

2. On the Model Properties window, edit the fields that you want to update.

   You can update the properties with the editable white background only.

   For property descriptions, see *Deploying a Replication Model* on page 19.

3. Click **Update**.

Identifying Tables in the Primary Schema

To view which tables are included in the replication model, select the host and expand the RAXContainer from the **Agent Servers > Known Agents** list.

1. To view the list of operations you can perform, right-click the **HANA2HANA** replication model and select the **Primary Schema Tables** menu option.
2. On the Primary Tables Schema window, enter the search filter criteria:
   - **Table Include Filter** – limits the table list to the tables that match with the search criteria.
   - **Table Exclude Filter** – excludes the tables that match with the filter search criteria from the table list.
   - **Add All Tables in the Schema** – adds all tables in the schema to the load script.
   - **Selected Tables Only** – includes only selected tables for replication.
   - **Non Selected Tables Only** – includes tables that are not selected for replication.

   Use the wildcard (*) to symbolize any number of any characters. For example, *ORDER matches with both NEW_ORDER and OORDER table names.

   A table that matches the search filter criteria, but is not included in the replication model, is included in the search result list with an Add check box next to it; a table that is already included has a Remove check box next to it.
3. Select or unselect tables from the Tables in Primary Schema pane for replication.
4. Select the **Add** or the **Remove** check box to include or exclude a table from the replication model.
5. Select any of these **Replication Options** check boxes:
   - **Create** – creates the table at the replicate database, if the table does not already exist. This refers to the replicate table not the CDC table. The CDC table is always created if it does not exist when you use either the *CDC* or the *Transactional CDC* strategy.
   - **Replace** – replaces the table at the replicate database, if it exists.
   - **Materialize** – populates the replicate table from the primary table before beginning trigger-based replication.
     For example, if the primary table, `PRIMARY.T1` has one million rows in it before replication starts, Replication Agent for SAP HANA selects one million rows from the `PRIMARY.T1` and applies them to the `REPLICATE.T1` before moving the data to the shadow tables with the triggers. Data from the shadow tables is not applied until all one million initial rows are replicated.

   Note: In the CDC strategy, the Materialize option populates the replicated CDC tables.

- **Defer** – instructs Replication Agent for SAP HANA not to create the table at the replicate database until the tables contain data that is to be replicated.

6. Click **Execute** to save the changes to the replication model.

   These changes are not applied to the primary or replicate database until replication begins.

7. To requery the database for the table name list, click **Refresh**.

   On the Status Messages panel, you can view the progress of:
   - Total Rows
   - Number of Remaining Rows
   - Average Rows/Seconds
   - Total Bytes
   - Bytes Remaining
   - Average Bytes
   - Estimated Time Remaining

   On the Table Materialization Status panel, you can view:
   - Primary Table Name
   - Replicate Table Name
   - Materialization Status
   - Progress in percentage
   - Row count
   - Rows Remaining
   - Rows/Seconds
   - Table Size
   - Other Statistical Information

   On the Status Messages panel, you can view the status messages for the **Replication Options** that you selected for each table from the primary to the replicate database.

8. (Optional) Right-click the model and select **Save Deployment Script for <model name>** to save a script of the Replication Agent for SAP HANA commands that are executed to deploy the model.

   This `Deploy_HANA2HANA_For_RAXContainer.sql` script might help you understand the Replication Agent for SAP HANA command reference APIs. You can execute the commands in the script in the command line interface (CLI) using a TDS client such as SAP **isql** utility.

Starting Replication

Start replication using the deployed model under the RAXContainer.

1. Right-click the model and select **Start Replication** menu item.

   You see a dialog indicating that replication has started. The model begins to:
   - Create the triggers, shadow tables, and trigger queues at the primary database.
   - Create CDC tables at the replicate database, if you are using either the *CDC* or the *Transactional CDC* strategy.
   - Create or replace tables at the replicate database, deferring the creation of those tables that are empty if you have selected the Defer option
   - Materialize the data from the primary to the replicate database.
   - Apply the trigger data from the primary to the replicate database.

2. Return to the Materialization Status and Replication Options window and click **Refresh** to see the overall materialization progress.

   Once the materialization completes, replication of the trigger data begins.

   Note: For the transactional CDC strategy, application of the trigger data to the replicate database cannot begin until materialization completes for all tables.

3. To monitor replication progress, right-click the model and select **Replication Status** menu item.

   On the Replication Status window, you can view the primary database activity, including transaction information that is written to the primary database and from the primary database to the shadow tables.

   The Table Replication Status panel provides information about tables being replicated with only an active transaction. You can view:
   - Tasks – the number of tasks (equivalent to process threads) dedicated to scanning the shadow tables.
   - Transactions – the number of transactions that have been executed for the table.
   - Inserts, Updates, and Deletes – the number of data manipulation language (DML) commands executed the tables.
   - Errors – the number of errors encountered while scanning the tables.
   - Queue Depth – the number of commands in the shadow table for the table.

   The Apply Statistics panel provides detailed statistics for the selected table. You can view:
   - Task Statistics – displays the activity of the task that reads from the transaction log, and the trigger queue.
   - Data Processing Statistics – displays the time spent in writing data to the corresponding CDC table.

- Shadow Table Statistics – displays the time spent in reading data from the shadow tables and the number of operations currently in the shadow table.

4. To view the replication status of trigger data, right-click the model and select **Scanner and Apply Task Status** menu item.

The Scanner and Apply Task Status window monitors activity at the replicate database.

On the Inflight Apply Tasks panel, you can view transactions that are written to the CDC tables.

The Scanner Statistics panel displays:

- Status – the status of the scanner that reads from the trigger queue.
- Tasks – tracks the number of apply tasks the scanner has executed and information on the results of those tasks.
- Trigger Queue – provides information on the trigger queue, how many records have been read, deleted, and are still in the queue, how much time has been spent reading and deleting from the trigger queue in total and on average, and the high-water mark (HWM) for the trigger queue.

Stopping and Undeploying Replication Model

To undeploy a model, stop replication using the **Stop Replication** option.

Replication Agent for SAP HANA provides a simple, yet comprehensive mechanism for undeploying a replication model and, optionally, removes the replication artifacts from the primary and the replicate database.

1. Once Replication has stopped, right-click the **<model_name>** and select **Undeploy <model_name>** menu item.

   where *model_name* is the name of the replication model deployed under the **RAXContainer**.

2. On the Undeploy? dialog, select the **Remove Replicate Artifacts** check box to remove all shadow tables, trigger queues, and transaction logs from the primary and CDC tables from the replicate database.

3. Click **OK**.

   When undeployment is successful, you see a confirmation dialog. The Replication Agent for SAP HANA removes all artifacts and the replication model, returning your primary and replicate databases as well as your RAXContainer instance to the same state as before you deployed the model.

Appendix A: Replication Artifacts for Transactional CDC Strategy at the Replicate Database

When the transaction change data capture (CDC) strategy is deployed, a CDC table is created for each table being replicated at the primary database. A transaction log table is also created to record information about all the database transactions that have been replicated. The Replication Agent for SAP HANA populates the CDC tables with the changed data from the primary as well as transaction header information.

Each CDC table has the columns of the primary table from which it is replicated plus four additional columns of auditing information.

Table 3. Change Data Capture Table

Column	Datatype	Description
HRAX_TRANS_NUMBER	`BIGINT`	The sequence number for the transaction.
HRAX_TRANS_TIME	`TIMESTAMP`	The time of the transaction.
HRAX_TRANS_TYPE	`VARCHAR(6)`	The operation type. Valid values are: INSERT | BEFORE | AFTER | DELETE **Note:** BEFORE and AFTER show column values before and after an update operation.
HRAX_TRANSACTION_ID	`VARCHAR(26)`	The transaction identifier.
{USER COLUMN…}	Same as primary	All columns in the primary table are included in the CDC table.

The transaction log table records all the replicated transactions that are executed at the primary database and the operations that have occurred in those transactions.

Table 4. Transaction Log Table

Column	Datatype	Description
HRAX_SEQ_NUMBER	BIGINT	The sequence number for the transaction.
HRAX_TRANS_TIME	TIMESTAMP	The time of the transaction.
HRAX_TRANS_TYPE	VARCHAR(6)	The operation type. Valid values are: INSERT | BEFORE | AFTER | DELETE.
HRAX_TRANSACTION_ID	VARCHAR(26)	The transaction identifier.
HRAX_TABLE_NAME	VARCHAR(255)	The name of the modified table.

These queries return all operations in the CDC_CUSTOMER table ordered by transaction sequence number:

```
select *
from TGT.CDC_CUSTOMER
where HRAX_TRANS_NUMBER > 3
order by HRAX_TRANS_NUMBER;
```

and

```
select CUST.HRAX_TRANS_NUMBER
,CUST.HRAX_TRANS_TIME
,CUST.HRAX_TRANSACTION_ID
,CUST.HRAX_TRANS_TYPE
,CUST.c_w_id
,CUST.c_d_id
from TGT.CDC_CUSTOMER AS CUST, TGT.CDC_TRANSACTION_LOG AS TXNLOG
where (CUST.HRAX_TRANS_NUMBER = TXNLOG.HRAX_SEQ_NUMBER)
and (TXNLOG.HRAX_SEQ_NUMBER > 3 )
ORDER BY CUST.HRAX_TRANS_NUMBER;
```

This query returns a list of all the tables that have received operations from a certain time:

```
select CUST.HRAX_TRANS_NUMBER
,CUST.HRAX_TRANS_TIME
,CUST.HRAX_TRANSACTION_ID
,CUST.HRAX_TRANS_TYPE
,CUST.c_w_id
,CUST.c_d_id
from CESTARGET.CDC_CUSTOMER CUST
where CUST.HRAX_TRANS_TIME > time'2014-02-27 12:00:00'
ORDER BY CUST.HRAX_TRANS_NUMBER;
```

This query returns a list of all the tables that have received **delete** operations from a certain time:

```
select DISTINCT(TXNLOG.HRAX_TABLE_NAME)
from CESTARGET.CDC_TRANSACTION_LOG TXNLOG
```

```
where TXNLOG.HRAX_TRANS_TIME > time'2014-02-27 23:54:40'
and TXNLOG.HRAX_TRANS_TYPE = 'DEL'
```

Appendix A: Replication Artifacts for Transactional CDC Strategy at the Replicate Database

Appendix B: Process Flow for Transactional CDC Replication Strategy

The transactional CDC replication strategy comprises a sequence of operations that are performed during replication.
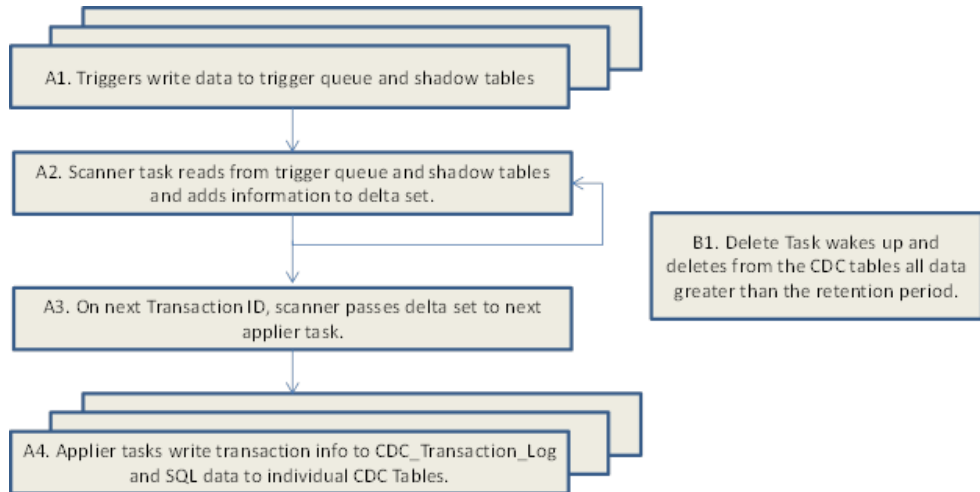
- *Sequence A* – shows the sequence of operations that are performed to move data from the user tables in the primary database to the CDC tables at the replicate database.
- *Sequence B* – shows the task that removes data from the CDC tables.
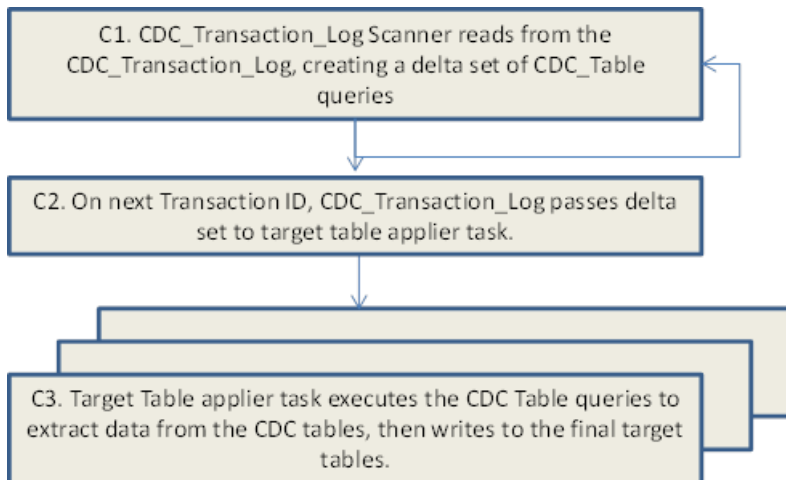
- *Sequence C* – shows a process flow for reading data from the CDC tables.

  Sequence C is not part of Replication Agent for SAP HANA, but is a customized process that you might write to retrieve data from the CDC tables and send the data to the replicate tables.

Appendix C: Sample Load File

A load file sample for creating, replacing, or materializing tables.

```
#################################################################
##Comments begin with the # sign
##Each line will define the options for tables that are to be
included
#################################################################
## The CUSTOMER table will be created on start of replication,
## it will replace an existing table of the same name,
## it will be materialized and its creation will be deferred
## until there is activity in it if it is empty
#################################################################
CUSTOMER,create,replace,materialize, defer
##
#################################################################
## The DISTRICT table will not be created on start of replication,
## it will not replace an existing table of the same name,
## it will not be materialized
## and it will not have deferred creation if it is empty
## on replication startup
#################################################################
DISTRICT
##
HISTORY,create,replace
ITEM, create, replace
NEW_ORDER, create, replace
OORDER, create, replace
STOCK, create, replace
WAREHOUSE, create, replace
```

Appendix C: Sample Load File

Replication Agent for SAP HANA Database

Appendix D: Replication Agent for SAP HANA Command Line Interface

To run the Replication Agent for SAP HANA in the command line interface, save the deployment script of your replication model.

1. Once the replication model is deployed, right-click it and select **Save Deployment Script for <model_name>** menu item.
2. Browse to and select the destination directory.
3. Specify the file name for the script and click **Save**.

You can use a text editor to open the deployment script file.

An example of a deployment script that configures, deploys, populates the tables selected for materialization, and starts the replication model :

```
call ra_add_prop(repname, CES2CES)
call ra_add_prop(primaryUser, CESSOURCE)
call ra_add_prop(replicatePassword, Sybase123)
call ra_add_prop(primaryURL, jdbc:sap://replinuxhana4.den2.sap.corp:
30115/CESSOURCE)
call ra_add_prop(rollbackScanInterval, 5000)
call ra_add_prop(maxTasks, 10)
call ra_add_prop(replicateDBIsolationLevel, 2)
call ra_add_prop(repName, CES2CES)
call ra_add_prop(replicateUser, CESTARGET)
call ra_add_prop(ddlDropTable, false)
call ra_add_prop(handleDDL, false)
call ra_add_prop(primaryPassword, Sybase123)
call ra_add_prop(primaryDriverName, com.sap.db.jdbc.Driver)
call ra_add_prop(primaryDBIsolationLevel, 2)
call ra_add_prop(replicateSchema, CESTARGET)
call ra_add_prop(replicateHost, replinuxhana4.den2.sap.corp)
call ra_add_prop(builderType, TRANS_CDC)
call ra_add_prop(primarySchema, CESSOURCE)
call ra_add_prop(replicatePort, 30115)
call ra_add_prop(commitInterval, 100)
call ra_add_prop(replicateDriverName, com.sap.db.jdbc.Driver)
call ra_add_prop(replicateURL, jdbc:sap://
replinuxhana4.den2.sap.corp:30115/CESTARGET)
call ra_add_prop(primaryHost, replinuxhana4.den2.sap.corp)
call ra_add_prop(cdcRetentionMinutes, 14400)
call ra_add_prop(forceCleanDeploy, false)
call ra_add_prop(primaryPort, 30115)
call ra_add_prop(materializationRetryTime, 60)
call ra_deploy()
call ra_add_table(CES2CES, HISTORY, HISTORY, create=true,
replace=true, materialize=true, deferReplication=true)
call ra_add_table(CES2CES, CUSTOMER, CUSTOMER, create=true,
replace=true, materialize=true, deferReplication=true)
```

```
call ra_add_table(CES2CES, DISTRICT, DISTRICT, create=true,
replace=true, materialize=true, deferReplication=true)
call ra_add_table(CES2CES, OORDER, OORDER, create=true,
replace=true, materialize=true, deferReplication=true)
call ra_add_table(CES2CES, WAREHOUSE, WAREHOUSE, create=true,
replace=true, materialize=true, deferReplication=true)
call ra_add_table(CES2CES, ITEM, ITEM, create=true, replace=true,
materialize=true, deferReplication=true)
call ra_add_table(CES2CES, STOCK, STOCK, create=true, replace=true,
materialize=true, deferReplication=true)
call ra_add_table(CES2CES, NEW_ORDER, NEW_ORDER, create=true,
replace=true, materialize=true, deferReplication=true)
call ra_add_table(CES2CES, ORDER_LINE, ORDER_LINE, create=true,
replace=true, materialize=true, deferReplication=true)
call ra_start(CES2CES)
```

For command descriptions, see the *Appendix E: Command Reference Table* on page 41.

Executing the Deployment Script Using the SAP HANA Agent Control Center GUI

Use the deployment script file in the SAP HANA Agent Control Center GUI to deploy a replication model.

1. Right-click **RAXContainer** and select **Deploy from a Script File** menu item.
2. To select the deployment script file, click **Select a File**.
3. On the Status Messages window, click **View File** to review the deployment script.

   Note: Changes that you make to the contents of the Status Messages window are not saved in the actual deployment script file.

4. (Optional) Select **Stop on any Error** to stop the deployment script when the first error is encountered.
5. Click **Execute**.
   The results appear in the Status Messages window.

Executing the Deployment Script Using the SAP Adaptive Server isql Utility

To execute the commands from the deployment script file, use the **isql** utility included in the Open Client™ installer that is part of the SAP Replication Server media.

Prerequisites
Before executing a deployment script using the **isql** utility, modify the command terminators in the script. By default, **isql** uses go as the command terminator, whereas the Replication

Agent for SAP HANA saves script files using the line character as the command terminator. The **isql** utility does not allow line control characters as command terminators.

Task

The **isql** utility is a command line interactive SQL parser to SAP® Adaptive Server® Enterprise. For more information about the **isql** utility, see the *SAP Adaptive Server Enterprise Utility Guide*.

1. Use a text editor to change the script to use the default **isql** command terminator.

For example, change these lines from:

```
call ra_add_prop(driverName, com.sap.db.jdbc.Driver)
call ra_deploy()
```

to:

```
call ra_add_prop(driverName, com.sap.db.jdbc.Driver)
go
```

```
call ra_deploy()
go
```

2. Save the script.

3. Execute the script from the **isql** command line.

For example:

```
isql -Usa -PSybase123 -SMY_RAX_CONTAINER < c:\Sybase\Scripts
\MyDeployScript.sql
```

In this example:

- *sa* – is the user ID.
- *Sybase123* – is the password for your container.
- *MY_RAX_CONTAINER* – is the server name.

Servers are defined in the `sql.ini` file. The `c:\Sybase\Scripts\MyDeployScript.sql` is the deployment script file.

Appendix E: Command Reference

Replication Agent for SAP HANA command line APIs.

Command	Descriptions
ra_add_prop on page 42	Adds a Replication Agent property for deployment into a replication model.
ra_add_table on page 46	Adds a table for replication with materialization options.
ra_add_tables on page 47	Adds one or more tables for replication.
ra_alter_prop on page 47	Alters a Replication Agent property for a deployed model.
ra_apply_statistics on page 48	Retrieves table apply statistics.
ra_cdc_apply_statistics on page 48	Retrieves Change Data Capture (CDC) statistics.
ra_delete_config on page 49	Deletes a configuration property.
ra_deploy on page 49	Deploys a Replication Agent.
ra_eventual_apply_statistics on page 50	Retrieves eventual apply statistics.
ra_exclude_table on page 50	Excludes a table from being replicated.
ra_get_all_config on page 50	Retrieves server configuration data.
ra_get_all_models on page 51	Retrieves names of all the deployed models.
ra_get_deployed_model on page 51	Retrieves replication model objects.
ra_get_mat_start_time on page 52	Retrieves the materialization start time.
ra_get_performance_data on page 52	Retrieves the performance data object.
ra_get_properties on page 52	Retrieves a specified Replication Agent property, or all properties.
ra_get_timestamp on page 54	Retrieves the server timestamp.
ra_help on page 54	Obtains the help information for a specified Replication Agent for SAP HANA command, or for all commands.
ra_mat_statistics on page 54	Retrieves materialization statistics.

Command	Descriptions
ra_password on page 55	Changes the password of the administrator.
ra_remove_exclude on page 55	Removes a table from the replication exclude list.
ra_remove_table on page 56	Removes tables from the replication include list.
ra_resume on page 56	Resumes a Replication Agent.
ra_role on page 57	Adds, drops, or shows the Replication Agent role (ra_role) privileges.
ra_scanner_statistics on page 57	Retrieves scanner statistics.
ra_set_config on page 58	Sets or adds a configuration property setting.
ra_start on page 58	Starts a Replication Agent.
ra_status on page 59	Retrieves Replication Agent status.
ra_stop on page 60	Stops a Replication Agent.
ra_suspend on page 60	Suspends a Replication Agent.
ra_table_excludes on page 60	Retrieves a list of tables to be excluded from replication.
ra_table_includes on page 61	Retrieves a list of tables to be replicated.
ra_tran_apply_statistics on page 61	Retrieves transactional apply statistics.
ra_trancdc_apply_statistics on page 61	Retrieves transactional CDC apply statistics.
ra_undeploy on page 62	Undeploys a Replication Agent.

ra_add_prop

Adds a Replication Agent property for deployment into a replication model.

Syntax

ra_add_prop propertyName, propertyValue

Parameters

- **propertyName** – specifies the property name you want to deploy.

Property Name	Description
`builderType`	The replication strategy used in a model. Valid values are CDC, TRANS_CDC, EVENTUAL, or TRANSACTIONAL.
`cdcRetentionMinutes`	The amount of time to retain the CDC data. Default: 100 Minimum: 0 Maximum: 2147483647
`commitInterval`	The commit batch size. Default: 100 Minimum: 0 Maximum: 2147483647
`ddlDropTable`	Determines whether data definition language (DDL) drop table is replicated. Default: false Valid values are true or false.
`forceCleanDeploy`	Forces a clean model deployment. Default: false Valid values are true or false.
`handleDDL`	Determines whether DDL statements are replicated. Default: false Valid values are true or false.
`materializationRetryTime`	The amount of time to wait before retrying a failed materialization. Default: 60 seconds Minimum: 0
`maxTasks.`	The maximum number of concurrent materialization tasks Default: 100 Minimum: 0 Maximum: 2147483647

Property Name	Description
`primaryDBIsolationLevel`	The connection transaction isolation level for the primary database.
	Default: TRANSACTION_READ_COMMIT-TED
	Valid values are TRANSAC-TION_READ_COMMITTED, TRANSAC-TION_READ_UNCOMMITTED, TRANS-ACTION_REPEATABLE_READ, and TRANSACTION_SERIALIZABLE.
`primaryDriverName`	A valid primary database JDBC driver name.
`primaryHost`	A valid primary database host name.
`primaryPassword`	A valid password for the primary database user.
`primaryPort`	A valid primary database port number.
`primarySchema`	A valid schema name that exists in the primary database.
`primaryURL.`	A valid primary database JDBC URL.
`primaryUser`	The primary database user name.
	The `primaryUser` property value cannot be blank.
`replicateDBIsolationLevel`	The connection transaction isolation level for the replicate database.
	Default: TRANSACTION_READ_COMMIT-TED
	Valid values are TRANSAC-TION_READ_COMMITTED, TRANSAC-TION_READ_UNCOMMITTED, TRANS-ACTION_REPEATABLE_READ, and TRANSACTION_SERIALIZABLE.
`replicateDriverName`	A valid replicate database JDBC driver name.
`replicateHost`	A valid replicate database host name.
`replicatePassword`	A valid password for the replicate database user.
`replicatePort`	A valid replicate database port number.
`replicateSchema`	A valid schema name that exists in the replicate database.

Property Name	Description
`replicateURL`	A valid replicate database JDBC URL.
`replicateUser`	The replicate database user name. The `replicateUser` property value cannot be blank.
`repName`	The replication model name.
`rollbackScanInterval`	The scan time for CDC and Eventual scanners to check the shadow tables for unprocessed changes. Default: 5000 milliseconds Minimum: 0

- **propertyValue** – specifies the value of the property.

Examples

- **Example 1 –** Adds the `Replication Strategy` property to the `HANA2HANA` replication model:

```
ra_add_prop builderType, TRANS_CDC
go
 PROPERTY NAME PROPERTY VALUE
 ------------- --------------
 buildertype   TRANS_CDC
```

You can separate parameters with a comma, a space, or a comma and a space.

- **Example 2 –** Adds the `Replicate SAP HANA Dataserver Name` property to the `HANA2HANA` replication model:

```
ra_add_prop replicateuser, HANA_rep1
go
 PROPERTY NAME PROPERTY VALUE
 ------------- --------------
 buildertype   TRANS_CDC
 replicateuser HANA_rep1
```

Usage

Use **ra_add_prop** to define the primary and replicate SAP HANA database properties.

ra_add_table

Adds a table for replication with materialization options.

Syntax

```
ra_add_table modelName, primaryTable, replicateTable,
create=[true|false],
replace=[true|false],
materialize=[true|false],
deferReplication=[true|false]
```

Parameters

- **modelName** – specifies the name of the replication model.
- **primaryTable** – specifies the primary SAP HANA database table name.
- **replicateTable** – specifies the replicate SAP HANA database table name.
- **create** – creates the table at the replicate SAP HANA database, if the table in the *primaryTable* parameter does not already exist. Valid values are true or false.
- **replace** – replaces the table at the replicate database, if it exists. Valid values are true or false.
- **materialize** – determines whether to materialize the table. Valid values are true or false.
- **deferReplication** – determines whether to defer replication of a table, if there is no data in that table. Valid values are true or false.

  Use the **deferReplication** parameter to instruct Replication Agent for SAP HANA not to create the table at the replicate database, if there is no data stored for replication.

Examples

- **Example 1** – Adds a primary SAP HANA database table named sap_hana01 in the list of tables for replication with materialization:

```
ra_add_table HANA2HANA, sap_hana01, materialize = true,
go
```

- **Example 2** – Adds the table named sap_ha334 with an option to defer replication:

```
ra_add_table HANA2HANA, sap_ha334, deferReplication = true
go
```

- **Example 3** – Includes all tables that start with soa for replication:

```
ra_add_table HANA2HANA, soa*, *, create=true, replace=true,
materialize=true, deferReplication=false
go
```

Usage

- Set the **materialize** parameter to populate the replicate table from the primary table before beginning trigger-based replication.

  Note: In the *CDC* strategy, the **materialize** parameter populates the replicated CDC tables.

- Use a wildcard to specify table names. For example, you can use an asterisk (*) to specify "TABLE_PREFIX*", which includes all tables that start with TABLE_PREFIX for replication. If you use the wildcard, the **replicateTable** parameter must be set to "*".

ra_add_tables

Adds one or more tables for replication.

Syntax

```
ra_add_tables modelName, tableName,
tableName...
```

Parameters

- **modelName** – specifies the name of the replication model.
- **tableName** – specifies the table name.

Examples

- **Example 1** – Adds a table named sap_hana02 for replication:

```
ra_add_tables HANA2HANA, sap_hana02
go
```

- **Example 2** – Adds the tables name sap_ha03 and sap_ha334 for replication:

```
ra_add_tables HANA2HANA, sap_ha03, sap_ha334
```

ra_alter_prop

Alters a Replication Agent property for a deployed model.

Syntax

```
ra_add_prop modelName, propertyName,
propertyValue...
```

Parameters

- **modelName** – specifies the name of the replication model.

- **propertyName** – specifies the Replication Agent property name.
- **propertyValue** – specifies the value for the Replication Agent property.

Examples

- **Example 1** – Alters the `commitInterval` property in the `HANARAX` model:

```
ra_alter_prop hanarax,commitInterval,100
go
TASKNAME        TYPE        VALUE
------------- ---------- -------------
RA_ALTER_PROP Task Name  RA_ALTER_PROP
RA_ALTER_PROP Task State Completed
```

- **Example 2** – Alters the `ddlDropTable` property in the `HANARAX` model:

```
ra_alter_prop hanarax,ddlDropTable,true
go
TASKNAME        TYPE        VALUE
------------- ---------- -------------
RA_ALTER_PROP Task Name  RA_ALTER_PROP
RA_ALTER_PROP Task State Completed
```

ra_apply_statistics

Retrieves the table apply statistics.

Syntax

ra_apply_statistics `modelName`

Parameters

- **modelName** – specifies the name of the replication model.

ra_cdc_apply_statistics

Retrieves the Change Data Capture (CDC) statistics.

Syntax

ra_cdc_apply_statistics `modelName`

Parameters

- **modelName** – specifies the name of the replication model.

ra_delete_config

Deletes a configuration property.

Syntax

ra_delete_config compname
parmname

Parameters

- **compname** – specifies the name of the replication model.
- **parmname** – specifies the configuration property name that you want to delete.

Examples

- **Example 1** – Deletes the Replication Strategy property from the HANA2HANA
  replication model:

  ```
  ra_delete_config HANA2HANA, builderType
  go
  ```

- **Example 2** – Deletes the roll back scan interval property from the
  HANA2HANA replication model:

  ```
  ra_delete_config HANA2HANA, rollbackScanInterval
  go
  ```

ra_deploy

Deploys a Replication Agent.

Syntax

ra_deploy

Parameters

None.

Usage

Before executing this command, set the replication model properties in the
rs_SetSessionProperties configuration fileor add properties using the **ra_add_prop**
command.

ra_eventual_apply_statistics

Retrieves the eventual apply statistics.

Syntax

ra_eventual_apply_statistics `modelName`

Parameters

- **modelName** – specifies the name of the replication model.

ra_exclude_table

Excludes a table from being replicated.

Syntax

ra_exclude_table `modelName, tableName`

Parameters

- **tableName** – specifies the table name.
- **modelName** – specifies the name of the replication model.

Examples

- **Example 1** – Excludes a table named `sap_hana02` in the HANA2HANA model from being replicated:

  ```
  ra_exclude_table HANA2HANA, sap_hana02
  go
  ```

- **Example 2** – Excludes a table named `sap_ha334` in the HANA2HANA model from being replicated:

  ```
  ra_exclude_table HANA2HANA, sap_ha334
  go
  ```

ra_get_all_config

Retrieves server configuration data.

Syntax

ra_get_all_config

Parameters

None.

ra_get_all_models

Retrieves names of all of the deployed models.

Syntax

ra_get_all_models

Parameters

None.

ra_get_deployed_model

Retrieves replication model objects.

Syntax

ra_get_deployed_model modelName

Parameters

• **modelName** – specifies the name of the replication model.

Examples

• **Example 1** – Retrieves the replication objects in the HANARAX model:

```
ra_get_deployed_model hanarax
go

MODELNAME   STATUS                    MATERIALIZATIONCOMPLETE
MODELUUID                             SCANNERACTIVE PRIMARYDBVERSION
PRIMARYDRIVERVERSION      PRIMARYDBNAME REPLICATEDBVERSION
REPLICATEDRIVERVERSION      REPLICATEDBNAME REPNAME
PRIMARYURL              REPLICATEURL         PRIMARYUSER REPLICATEUSER
PRIMARYSCHEMA REPLICATESCHEMA PRIMARYPASSWORD REPLICATEPASSWORD
PRIMARYDRIVERNAME      REPLICATEDRIVERNAME    MAXTASKS   COMMITINTERVAL
ROLLBACKSCANINTERVAL BUILDERTYPE HANDLEDDL  DDLDROPTABLE FORCECLEANDEPLOY
CDCRETENTIONMINUTES PRIMARYDBISOLATIONLEVEL REPLICATEDBISOLATIONLEVEL
MATERIALIZATIONRETRYTIME NUMBEROFINCLUDEDTABLES NUMBEROFEXCLUDEDTABLES
---------- ----------------------- -----------------------
------------------------------------ ------------ ----------------
---------------------- --------------- -----------------
------------------------- --------------- ----------
-------------------- ---------------------- ----------- -------------
------------- --------------- -------------- -----------------
```

```
-------------------- -------------------- ---------- --------------
------------------ ----------- ---------- ------------
--------------- ------------------ ----------------------
----------------------- -----------------------
-------------------- --------------------
hanarax    Deployed but not running false
35cbb1cf-0e69-4b6a-8717-baafaaa0d43b false          1.00.61.380109
1.00.68.Build 0383365-1110 HDB          1.00.61.380109    1.00.68.Build
0383365-1110 HDB             hanarax    jdbc:sap:rshana2:30215
jdbc:sap:rshana2:30215 BLLPRIMARY  BLLREPLICATE BLLPRIMARY
BLLREPLICATE    Primary3        Replicate3      com.sap.db.jdbc.Driver
com.sap.db.jdbc.Driver 10        100          5000
TRANS_CDC   true       true        false          1000
2                       2                        15
0                       0

(1 row affected)
```

ra_get_mat_start_time

Retrieves the materialization start time.

Syntax

ra_get_mat_start_time

Parameters

None.

ra_get_performance_data

Retrieves the performance data object.

Syntax

ra_get_performance_data

Parameters

None.

ra_get_properties

Retrieves a specified Replication Agent property, or all properties.

Syntax

```
ra_get_properties modelName, [propertyName | none]
```

Parameters

- **modelName** – specifies the name of the replication model.
- **propertyName** – specifies the name of the property that you want to retrieve.

Examples

- **Example 1** – Retrieves the `builderType` property from the HANA2HANA model:

```
ra_get_properties HANA2HANA, builderType
go
```

- **Example 2** – Retrieves all properties for the HANARAX model:

```
ra_get_properties hanarax
go

PROPERTY NAME              PROPERTY VALUE
------------------------   ----------------------
builderType                TRANS_CDC
cdcRetentionMinutes        1000
commitInterval             100
ddlDropTable               true
forceCleanDeploy           false
handleDDL                  true
materializationRetryTime   15
maxTasks                   10
primaryDBIsolationLevel    2
primaryDriverName          com.sap.db.jdbc.Driver
primaryHost                NULL
primaryPassword            ********
primaryPort                NULL
primarySchema              BLLPRIMARY
primaryURL                 jdbc:sap:rshana2:30215
primaryUser                BLLPRIMARY
replicateDBIsolationLevel  2
replicateDriverName        com.sap.db.jdbc.Driver
replicateHost              NULL
replicatePassword          **********
replicatePort              NULL
replicateSchema            BLLREPLICATE
replicateURL               jdbc:sap:rshana2:30215
replicateUser              BLLREPLICATE
repName                    hanarax
rollbackScanInterval       5000

(26 rows affected)
```

ra_get_timestamp

Retrieves the server timestamp.

Syntax

ra_get_timestamp

Parameters

None.

ra_help

Obtains the help information for a specified Replication Agent for SAP HANA command, or for all commands.

Syntax

ra_help [commandName | none]

Parameters

- **commandName** – specifies the name of the command for which you want to obtain the help information.

Examples

- **Example 1** – Obtains for the **ra_get_all_config** command:

```
ra_help ra_get_all_config
go
```

- **Example 2** – Obtains the help information for all commands:

```
ra_help
go
```

ra_mat_statistics

Retrieves materialization statistics.

Syntax

ra_mat_statistics modelName

Replication Agent for SAP HANA Database

Parameters

- **modelName** – specifies the name of the replication model.

ra_password

Changes the password of the administrator.

Syntax

```
ra_password oldPassword
newPassword
```

Parameters

- **oldPassword** – the current password for the administrator user.
- **newPassword** – specifies the new password.

Examples

- **Example 1** – Changes the password of the administrator named rah_admin_user:

```
ra_password rah_admin_user
oldPassword 'sa12341'
newPassword 'sy123'
go
```


ra_remove_exclude

Removes a table from the replication exclude list.

Syntax

```
ra_remove_exclude modelName, tableName
```

Parameters

- **modelName** – specifies the name of the replication model.
- **tableName** – specifies the table name to remove.

Examples

- **Example 1** – Removes a table named sa_demo02 from the exclude list:

```
ra_remove_exclude HANA2HANA, sa_demo02
go
```

- **Example 2 –** Removes a table name `sa_ha121` from the exclude list:

```
ra_remove_exclude HANA2HANA, sa_ha121
go
```

ra_remove_table

Removes a table from the replication include list.

Syntax

```
ra_remove_table modelName, tableName
tableName...
```

Parameters

- **modelName –** specifies the name of the replication model.
- **tableName –** specifies the table name to remove.

  Add a comma (,) between table names to remove more than one table from the include list.

Examples

- **Example 1 –** Removes the tables named `sa_demo03` and `sa_demo4` tables from the include list:

```
ra_remove_table HANA2HANA, sa_demo03, sa_demo4
go
```

- **Example 2 –** Removes a table named `sa_ha334` from the include list:

```
ra_remove_table HANA2HANA, sa_ha334
go
```

ra_resume

Resumes a Replication Agent.

Syntax

```
ra_resume
```

Parameters

None.

ra_role

Adds, drops, or shows Replication Agent privileges.

Syntax

```
ra_role [add | drop],
username ra_role|ra_role rolename |
ra_role rolename add username |
ra_role rolename drop username
```

Parameters

- **add –** adds a privilege to a user name.
- **drop –** revokes a privilege from a user name.
- **rolename –** specifies the case-sensitive role name.

Examples

- **Example 1 –** shows all users with the Replication Agent administrator role:

```
ra_role administrator
go
```

- **Example 2 –** adds the Replication Agent administrator role to the user named "soane":

```
ra_role administrator add soane
go
```

- **Example 3 –** drops the Replication Agent administrator role from the user name "hauser":

```
ra_role administrator drop hauser
go
```

Usage

A **role** consists of a predefined set of privileges and a set of users authorized to invoke the Replication Agent for SAP HANA.

ra_scanner_statistics

Retrieves the scanner statistics.

Syntax

```
ra_scanner_statistics modelName
```

Parameters

- **modelName** – specifies the name of the replication model.

ra_set_config

Sets or adds a configuration property setting.

Syntax

ra_set_config compname, parmname, parmval, desc

Parameters

- **compname** – specifies the name of the replication model.
- **parmname** – specifies the configuration property name that you want to set.
- **parmval** – the value of the configuration property.
- **desc** – adds the description for the configuration property.

Examples

- **Example 1** – Adds the `builderType` property to the `HANA2HANA` replication model:

```
ra_set_config HANA2HANA, builderType, trans_cdc
go
```

ra_start

Starts a Replication Agent.

Syntax

ra_start modelName

Parameters

- **modelName** – specifies the name of the replication model.

Examples

- **Example 1** – Starts the `HANARAX` replication model:

```
ra_start hanarax
go

TASKNAME    TYPE        VALUE
---------- ---------- ----------
```

```
RA_START    Task Name  RA_START
RA_START    Task State Completed

(2 rows affected)
```

ra_status

Retrieves Replication Agent status.

Syntax

ra_status modelName

Parameters

• **modelName** – specifies the name of the replication model.

Examples

• **Example 1** – Retrieves status of the HANARAX model when deployed:

```
ra_status hanarax
go

AGENT STATUS DESCRIPTION              SCANNER RUNNING ERROR COUNT
------------ ----------------------- --------------- -----------
DEPLOYED     Deployed but not running false           0

(1 row affected)
```

• **Example 2** – Retrieves status of the HANARAX model when the model is not deployed:

```
ra_status hanarax
go

TASKNAME                       TYPE       VALUE
--------------------------- ----------
------------------------------------------------
RA_STATUS                      Task Name  RA_STATUS
RA_STATUS                      Task State Error

Get Replication Agent Status Error     Replication Agent model
'hanarax' is not deployed

(3 rows affected)
```

ra_stop

Stops a Replication Agent.

Syntax

ra_stop modelName

Parameters

• **modelName** – specifies the name of the replication model.

Examples

• **Example 1** – Stops the HANARAX model:

```
ra_stop hanarax
go

TASKNAME    TYPE        VALUE
---------- ---------- ----------
RA_STOP     Task Name  RA_STOP
RA_STOP     Task State Completed

(2 rows affected)
```

ra_suspend

Suspends a Replication Agent.

Syntax

ra_suspend

Parameters

None.

ra_table_excludes

Retrieves a list of tables to be excluded from replication.

Syntax

ra_table_excludes modelName

Parameters

- **modelName** – specifies the name of the replication model.

ra_table_includes

Retrieves a list of tables to be replicated.

Syntax

`ra_table_includes` modelName

Parameters

- **modelName** – specifies the name of the replication model.

ra_tran_apply_statistics

Retrieves transactional apply statistics.

Syntax

`ra_tran_apply_statistics` modelName

Parameters

- **modelName** – specifies the name of the replication model.

ra_trancdc_apply_statistics

Retrieves transactional CDC apply statistics.

Syntax

`ra_trancdc_apply_statistics` modelName

Parameters

- **modelName** – specifies the name of the replication model.

ra_undeploy

Undeploys a Replication Agent.

Syntax

```
ra_undeploy modelName, removeReplicationArtifacts
```

Parameters

- **modelName** – specifies the name of the replication model.
- **removeReplicationArtifacts** – determines whether to remove replication model artifacts during undeployment. Valid values are true or false.

Examples

- **Example 1** – Undeploys the HANA2HANA replication model:

```
ra_undeploy HANA2HANA
go
```

- **Example 2**

  Undeploys the HANARAX replication model and removes all replication artifacts deployed:

```
ra_undeploy hanarax,true
go

TASKNAME    TYPE        VALUE
----------- ----------- -----------
RA_UNDEPLOY Task Name   RA_UNDEPLOY
RA_UNDEPLOY Task State  Completed

(2 rows affected)
```

Glossary

Definitions of terms related to Replication Agent for SAP HANA.

applier – An application service that writes data to the change data capture (CDC) tables using one of the CDC modes and the replicate database user-defined tables when a non-CDC mode is enabled.

builder – An application service that builds shadow tables and other control tables.

container – A server that provides a management application programming interface (API), a logging service, and a repository for metadata.

isql – A command line interactive SQL parser to SAP® Adaptive Server® Enterprise (SAP® ASE).

multitenancy – A network model that allows system administrators to deploy a single instance of the software that runs on a server, serving multiple client organizations known as tenants.

Replication Agent for SAP HANA – A lightweight server that replicates data from a primary to a replicate SAP HANA® database.

Replication Management Agent – A distributed management agent that you can use to easily set up and manage replication from any supported databases to an SAP HANA database.

RMI – remote method invocation

A remote procedure call used for communication between DA server and DA agents.

schema – The structure of the database.

TDS – Tabular Data Stream™

An application protocol by which Open Client™ and SAP® Open Server™ applications exchange information.

Glossary